



**PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**



Awdren de Lima Fontão

**DEVGO: UM MODELO PARA GOVERNANÇA DE
DESENVOLVEDORES EM ECOSSISTEMA DE SOFTWARE
MÓVEL A PARTIR DE *DEVELOPER RELATIONS***

Manaus
2019

AWDREN DE LIMA FONTÃO

DEVGO: UM MODELO PARA GOVERNANÇA DE
DESENVOLVEDORES EM ECOSSISTEMA DE SOFTWARE
MÓVEL A PARTIR DE *DEVELOPER RELATIONS*

Tese de doutorado apresentada ao Programa de Pós-Graduação em Informática do Instituto de Computação da Universidade Federal do Amazonas, como requisito parcial para a obtenção do título de Doutor em Informática.

Orientador: Prof. Dr. Arilo Claudio Dias Neto
Coorientador: Prof. Dr. Rodrigo Pereira dos Santos

Manaus
2019

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

F685d Fontão, Awdren de Lima
DEVGO: um modelo para governança de desenvolvedores em
Ecossistema de Software Móvel a partir de Developer Relations /
Awdren de Lima Fontão. 2019
182 f.: il. color; 31 cm.

Orientador: Arilo Claudio Dias Neto
Coorientador: Rodrigo Pereira dos Santos
Tese (Doutorado em Informática) - Universidade Federal do
Amazonas.

1. ecossistema de software. 2. governança. 3. developer
relations. 4. desenvolvedores. 5. aplicação móvel. I. Dias Neto, Arilo
Claudio II. Universidade Federal do Amazonas III. Título



PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
INSTITUTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



UFAM

FOLHA DE APROVAÇÃO

"DEVGO: Um Modelo para Governança de Desenvolvedores em Ecosystema de Software Móvel a Partir de Developer Relations"

AWDREN DE LIMA FONTÃO

Tese de Doutorado defendida e aprovada pela banca examinadora constituída pelos Professores:

Arilo Claudio Dias Neto

Prof. Arilo Claudio Dias Neto - PRESIDENTE

Eduardo Luiz Feitosa

Prof. Eduardo Luiz Feitosa - MEMBRO INTERNO

Bruno Freitas Gadelma

Prof. Bruno Freitas Gadelma - MEMBRO INTERNO

Sabrina dos Santos Marczak

Profa. Sabrina dos Santos Marczak - MEMBRO EXTERNO

Igor Scaliante Wiese

Prof. Igor Scaliante Wiese - MEMBRO EXTERNO

Manaus, 14 de Junho de 2019

*Aos meus pais e a toda minha família, que me ensinaram que a educação
mudaria a minha vida.*

Agradecimentos

À Deus, por ter sido tão generoso comigo me conduzido a conhece-lo por meio de várias pessoas que cruzaram a minha história até aqui. Ele me permitiu nascer e crescer em uma família que me ensinou o valor da educação como força de transformação. Eu não sei o que Ele viu em meu coração, mas confesso: sou grato!

À toda minha família, pela oferta do sorriso sincero e do olhar mais fraterno. Por me permitirem, em muitas vezes abrindo mão de suas próprias escolhas, o avanço, a perseverança e por me acolherem em seu lar.

À minha mãe, Maria Auxiliadora, por nunca ter desistido de mim, por me motivar e por um dia ter dito que eu teria que ser doutor, não pelo título, mas por acreditar no meu potencial. Umas palavras aqui nunca representarão toda força, amor e esforço oferecidos durante a minha vida.

Ao meu pai, João Fontão, por ter me ensinado a amar matemática, a ler, a assistir ao jornal televisivo e por ter dado o meu primeiro computador. E, pela paciência, toda vez que eu o desmontava ou quebrava algo. E pelos valores que me ensinou.

À minha irmã Andreza e à minha prima-irmã Luciane, por compartilharem muito dessa história e, também, por não terem desistido de seus sonhos. A todos meus primos e primas que são mais que isso, são irmãos.

Aos meus avôs Renato Lima e João Fontão (*in memorian*), por terem investido na minha educação, por terem sempre as melhores histórias sobre suas aventuras e por sempre perguntarem como eu estava.

Às minhas avós Eunice Fontão e Maria de Lourdes (*in memorian*), pelo cuidado materno, pelas orações e por criarem sempre um ambiente de amor rodeado pela família. E, por serem, apoio aos meus pais quando as situações apertavam.

A todos meus tios e tias, que sabem da sua ajuda pra minha caminhada. Como a lista é extensa aqui os represento pelas tias Nega, Corrinha, Dora (*in memorian*), Eliete (*in memorian*), Joana, Mônica e Ivone. E pelos tios, Paulo (*in memorian*), Alfremom, Lucas, Lúcio, Alexandre, Zé e Marcelo.

Ao meu orientador e amigo, Arilo Claudio Dias Neto, por ser um exemplo de profissional, por mesmo na distância presencial, ter sido cuidadoso e me ajudar com os conselhos pontuais e certos. Sempre vibrou com as conquistas, colocou os meus pés no chão e me motivou a focar visando o futuro. Um orgulho pra mim é dizer que fui orientado por um profissional tão respeitado, humano e, claro, torcedor do Garantido. Espero corresponder na caminhada profissional. Sempre que você precisar: conte comigo!

Ao meu coorientador e amigo, Rodrigo Pereira dos Santos, por ter atuado como facilitador para eu me inserir na comunidade científica, por exigir o cuidado nos detalhes e por ser um exemplo de jovem pesquisador. Outro orgulho para minha carreira é poder falar que você me orientou. Esteve presente, nas conferências, nos workshops, nas viagens e nas conversas sobre a vida. Não poderia de esquecer a presença na apreensão e comemoração de premiações relevantes para minha carreira. Obrigado por me ajudar a embarcar na amada área de Ecossistemas de Software.

A todos amigos do grupo ExperTS. Quando eu tiver meu grupo de pesquisa, sempre mencionarei o ambiente amistoso, divertido, de trocas de experiências, dos memes, dos debates sobre a pesquisa, dos estudos, dos dias cansativos, das comemorações de aniversários, do “momento do conselho”, das reuniões de grupo... Eu não poderia ter optado por um outro grupo. Futuramente sentirei mais saudades. E a saudade só se apresenta onde existe amor.

Aos amigos do ESC Lab na UNIRIO, por nas oportunidades durante os congressos ou reuniões de grupo, permitirem adentrar a pesquisa de vocês, acolherem minhas ideias e por partilharmos desta história.

À Tayana Conte, por ter sido uma das pessoas que me motivou a amar Engenharia de Software, pelas oportunidades de palestrar para suas turmas, por me dar conselhos e

deixar a “porta” da sala sempre aberta. Obrigado, ainda, por exigir muito de mim na Engenharia de Software Experimental.

A todo corpo administrativo e docente do PPGI do IComp/UFAM, pelo suporte para que eu pudesse realizar minha pesquisa, pelo conhecimento compartilhado, pelas oportunidades de disseminação da pesquisa e por serem referência.

Aos professores membros da minha banca, Sabrina Marczak, Igor Wiese, Bruno Gadelha e Eduardo Feitosa. Nesta fase, eu sou muito agradecido por terem aceito fazer parte e contribuir para a conclusão de uma etapa. Cada um de vocês contribuiu para a minha formação em algum momento.

Aos amigos que sempre foram força para o doutorado, que me motivaram a retornar e sempre foram exemplos de dedicação: Adriana, Anna Beatriz, Bernardo Estácio, Bruno Bonifácio, Davi Viana, Everton Cavalcante, Jackson Feijó, Juliana Fernandes, Ludimila, Priscila Fernandes e Rafael Padilla.

Aos meus amigos do PPGI, das disciplinas, dos cafés, das trocas de experiências e força: Rayol, Hendrio, Kevin, Bruno Ábia, Williamson e Okimoto. A caminhada se tornou mais leve com a companhia de vocês.

Aos meus amigos catequistas, catequizandos (principalmente, do Emaús, que foram os presentes deste período), afilhados e amigos de missão. Cada um de vocês sabe como foi essencial para esta caminhada, a força que deram e as ausências que entenderam. Aqui os represento pela Alciellen, Claudia, Stella, Matheus, Kelly, Lorena, Suzy, Mattheus e Lucas.

Aos amigos que fiz por meio da musculação e do Fitdance. Obrigado pela força e por ajudarem a liberar a tensão da caminhada e, também, pela história de cada um de vocês.

À CAPES pelo apoio financeiro durante o período de formação no doutorado.

A todos vocês que contribuíram de alguma forma para que esta etapa se realizasse. Ao ler você saberá como contribuiu. Obrigado!

Resumo de Tese de Doutorado apresentada à UFAM/AM como parte dos requisitos necessários para a obtenção do grau de Doutor em Informática (D.Sc.)

*DEVGO: UM MODELO PARA GOVERNANÇA DE DESENVOLVEDORES EM
ECOSSISTEMA DE SOFTWARE MÓVEL A PARTIR DE DEVELOPER RELATIONS*

Awdren de Lima Fontão
Junho / 2019

Orientador: Prof. Dr. Arilo Claudio Dias Neto (PPGI/UFAM)

Coorientador: Prof. Dr. Rodrigo Pereira dos Santos (PPGI/UNIRIO)

Organizações que mantêm ecossistemas de software móvel (MSECO), como Apple, Google e Microsoft, têm engajado uma massa crítica de desenvolvedores externos para expandir suas plataformas. Nesse sentido, elas exploram estratégias de governança de desenvolvedores. Estas estratégias dependem da operação associada entre os objetivos da organização e as expectativas dos desenvolvedores. A área de governança de desenvolvedores em MSECO ainda necessita de formalização, de teorias e modelos.

Desta forma, as organizações têm investido em Relações com Desenvolvedores (DevRel, do inglês *Developer Relations*). A indústria exige novas estratégias que promovam a colaboração da organização com os desenvolvedores externos. Como parte dessas estratégias, os profissionais de DevRel precisam ter uma visão realista do comportamento dos desenvolvedores dentro do ecossistema. Assim é possível atender às necessidades dos desenvolvedores e atingir metas da organização.

A definição de estratégias de governança de desenvolvedores não é trivial, consiste em delimitar as ações dos desenvolvedores sem restringir excessivamente o nível desejado de criação de valor. Se a governança de desenvolvedores for insustentável, pode-se incorrer na “morte” do ecossistema. Por exemplo, um dos fatores pelos quais o MSECO Windows Phone foi declarado oficialmente “morto” em 2017 foi porque os desenvolvedores não apoiaram mais a plataforma.

Nesta tese, por meio de um conjunto de estudos secundários, primários e de mineração de repositórios de software envolvendo 68 profissionais de DevRel, é investigada qual a estrutura da governança de desenvolvedores em MSECO e um conjunto de lições aprendidas. Com isto, foi construído e refinado um modelo para governança de desenvolvedores em MSECO chamado DevGo (do inglês, *DEVELOPER GOVERNANCE*). O DevGo é composto por: áreas de foco, fases de avanço do desenvolvedor, estágios, facilitadores e objetos de transferência de valor.

Além disso, um conjunto de lições aprendidas de profissionais de DevRel associadas a estágios do DEVGO, categorias de repositórios e motivações para uso deles com foco em monitoramento. A partir de estudos de caso, os participantes, destacara que a estrutura do DevGo facilita o entendimento da governança de desenvolvedores, dos objetos de transferência de valor, inclusive, para comunicar a importância da área para a organização e a área de foco de monitoramento. Não houve discordância em relação à adequação, ao controle, ao entendimento e à generalidade do DevGo.

Palavras-chave: modelo; ecossistema de software; aplicação móvel; governança; recomendações; *developer relations*; desenvolvedores.

Abstract of Thesis presented to PPGI/UFAM as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

*DEVGO: A DEVELOPER GOVERNANCE MODEL IN MOBILE SOFTWARE ECOSYSTEM
FROM DEVREL*

Awdren de Lima Fontão
June / 2019

Advisor: Prof. Dr. Arilo Claudio Dias Neto (PPGI/UFAM)

Co-advisor: Prof. Dr. Rodrigo Pereira dos Santos (PPGI/UNIRIO)

Organizations that maintain mobile software ecosystems (MSECO), such as Apple, Google and Microsoft, have engaged a critical mass of outside developers to expand their platforms. As such, they explore strategies for developer governance. These strategies depend on the associated operation between the organization's goals and the expectations of the developers. The MSECO developer governance area still needs formalization, theories and models.

In this context, organizations have invested in Developer Relations (DevRel). The industry requires new strategies that promote the organization's collaboration with external developers. As part of these strategies, DevRel professionals need to have a realistic view of developers' behavior within the ecosystem and then meet the developers' needs and achieve organizations' goals.

The definition of developer governance strategies is not trivial; it consists of delimiting the actions of developers without restricting excessively the desired level of value creation. If developer governance is unsustainable, it can lead to the "death" of the ecosystem. For example, one of the factors by which the Windows Phone MSECO was officially declared "dead" in 2017 was because the developers no longer supported the platform.

In this thesis, through a set of secondary and primary studies and mining software repositories involving DevRel professionals, we investigate the structure of MSECO developer governance and a set of lessons learned. Next, we built and refined a model for governance of developers in MSECO called DevGo (DEVELOPER GOVERNANCE). DevGo consists of: areas of focus, developer advancement stages, stages, facilitators, and value transfer objects.

In addition, a set of lessons learned from DevRel professionals associated with DevGo stages, repository categories, and motivations for their use with a focus on monitoring. From the case studies, participants noted that the DevGo framework facilitates understanding of developer governance, value transfer objects, as well as the communication of the importance of the field to the organization and the focus area regarding ecosystem monitoring. There was no disagreement regarding the adequacy, control, understanding and generality of DevGo

Keywords: model; software ecosystem; mobile application; governance; lessons learned; developer relations; developer.

ÍNDICE

LISTA DE ABREVIATURAS E SIGLAS	XIII
LISTA DE FIGURAS	XIV
LISTA DE TABELAS	XV
CAPÍTULO 1 - INTRODUÇÃO	1
1.1. Contextualização e Motivação	1
1.2. Descrição do Problema	3
1.3. Hipótese	3
1.4. Questões de Pesquisa	4
1.5. Objetivos	5
1.5.1. Objetivo Geral	5
1.5.2. Objetivos Específicos	5
1.6. Metodologia de Pesquisa	6
1.7. Contribuições da Tese	9
1.7.1. Principais Contribuições	9
1.7.2. Outros Resultados	10
1.8. Estrutura da Tese	12
CAPÍTULO 2 - REFERENCIAL TEÓRICO	13
2.1 Ecosistema de Software (ECOS) e de Software Móvel (MSECO)	13
2.2 Governança de Ecosistema de Software	16
2.3 Governança de Desenvolvedores em ECOS	17
2.3.1 Necessidade e revisões sistemáticas anteriores	17
2.3.2 Protocolo do Mapeamento Sistemático	19
2.3.3 (QP1) Em quais tipos de ECOS os estudos estão focados?	22
2.3.4 (QP2) Qual é a definição de “governança de desenvolvedores” ?	23
2.3.5 (QP3) Quais estratégias são utilizadas para governar desenvolvedores?	26
2.3.6 Ameaças à Validade	39
2.3.7 Conclusões do mapeamento	40
2.4 Trabalhos Relacionados	40
2.5 Considerações finais	44
CAPÍTULO 3 – ESTUDOS PARA CONCEPÇÃO E REFINAMENTO DO DEVGO: MODELO DE GOVERNANÇA DE DESENVOLVEDORES	45
3.1 Visão geral dos estudos	45
3.2 Pesquisa de opinião com especialistas	45
3.2.1 Objetivo do Estudo e Questão de Pesquisa	45
3.2.2 Instrumentação	46
3.2.3 Execução do Estudo	46
3.2.4 Análise dos Resultados	47
3.2.5 Ameaças à Validade	49
3.2.6 Conclusões do Estudo	50
3.3 Entrevistas com especialistas	50
3.3.1 Objetivo do Estudo	50
3.3.2 Instrumentação	50
3.3.3 Execução do Estudo	51
3.3.4 Análise dos Resultados	51
3.3.5 Ameaças à Validade	56
3.3.6 Conclusões do estudo	57
3.4 Identificação de lições aprendidas em DevRel	58
3.4.1 Objetivo do Estudo	58
3.4.2 Estratégia de busca e extração de dados	58
3.4.3 Análise dos resultados	60
3.4.4 Ameaças à Validade	60
3.4.5 Conclusões do Estudo	61
3.5 Estudo Exploratório sobre Repositórios	61
3.5.1 Objetivo do Estudo	61
	X

3.5.2	Instrumentação	61
3.5.3	Execução	62
3.5.4	Análise dos resultados	62
3.5.5	Ameaças à Validade	62
3.5.6	Conclusões do Estudo	63
3.6	Pesquisa de opinião sobre Percepção de Valor	64
3.6.1	Objetivo do Estudo e Questão de Pesquisa	65
3.6.2	Instrumentação	65
3.6.3	Execução do Estudo	66
3.6.4	Análise sobre a Percepção de Valor em DevRel	67
3.6.5	Análise dos Resultados dos Repositórios e Motivações	77
3.6.6	Ameaças à Validade	80
3.6.7	Conclusão da Pesquisa de Opinião	81
3.7	Considerações do Capítulo	82
CAPÍTULO 4 – CAPTURANDO AS EXPECTATIVAS DO DESENVOLVEDOR		83
4.1	Definição do Estudo	83
4.2	Minerando questões técnicas	83
4.2.1	Planejamento e projeto do estudo	83
4.2.2	Questões de pesquisa	84
4.3	Seleção dos dados	85
4.4	Execução do Estudo	86
4.4.1	Detectando tópicos relevantes	86
4.4.2	Filtrando a base de dados	87
4.5	Análise dos resultados e discussões	87
4.5.1	(QP1) Quais são os tópicos quentes extraídos de perguntas técnicas feitas por desenvolvedores de cada MSECO?	87
4.5.2	(QP2) O que pode ser entendido dos três principais MSECOs baseado na relação entre questões e emoções primárias?	89
4.5.3	(QP3) Quais são os tópicos relevantes extraídos a partir de questões técnicas sem resposta em cada MSECO?	96
4.5.4	(QP4) Quais são as perguntas em que mais os desenvolvedores se engajam?	97
4.5.5	(QP5) Existe alguma relação entre perguntas e eventos oficiais?	99
4.5.6	(QP6) Qual é o ranking do número de medalhas recebidas pelos desenvolvedores em cada plataforma?	102
4.6	Ameaças à Validade	104
4.7	Considerações do Capítulo	105
CAPÍTULO 5 – DEVGO: MODELO DE GOVERNANÇA DE DESENVOLVEDORES		106
5.1	Estrutura do DEVGO	106
5.1.1	Áreas de foco	108
5.1.2	Fases, estágios e facilitadores	114
5.2	Conclusões e Implicações do DEVGO	124
CAPÍTULO 6 – ESTUDO DE CASO: RETENÇÃO DE DESENVOLVEDORES		126
6.1	A retenção de desenvolvedores em MSECO por profissionais de DevRel	126
6.1.1	Objetivo	126
6.1.2	Questão de Pesquisa do Estudo de Caso	127
6.1.3	Questões de Pesquisa e Métricas	127
6.1.4	Planejamento do Estudo	128
6.1.5	Execução do Estudo	129
6.1.6	Ameaças à Validade	129
6.1.7	Análise Quantitativa	131
6.1.8	Análise Qualitativa	134
6.1.9	Conclusões do Estudo	145
CAPÍTULO 7 – CONCLUSÕES E PERSPECTIVAS FUTURAS		148
7.1	Conclusões	148
7.2	Limitações	150
7.3	Perspectivas Futuras	150
REFERÊNCIAS BIBLIOGRÁFICAS		152
APÊNDICE A – Artigos Selecionados no Mapeamento Sistemático		158

APÊNDICE B – Pesquisa de Opinião sobre Relevância de Estratégias	164
APÊNDICE C – Dados sobre Entrevistas	165
APÊNDICE D – Pesquisa de Opinião sobre Percepção de Valor	166
APÊNDICE E – Estudo de Caso e Entrevistas	168

LISTA DE ABREVIATURAS E SIGLAS

API - Interface de Programação de Aplicações (em inglês, *Application Programming Interface*)

APPS – Aplicações móveis

ASO – Otimização de Loja de Aplicações (em inglês, *App Store Optimization*)

DevGo – Nome do modelo de governança de desenvolvedores (em inglês, *DEveloper GOvernance*)

DevRel - Relações com Desenvolvedores (em inglês, *Developer Relations*)

DX – Experiência de Desenvolvedor (em inglês, *Developer Experience*)

ECOS – Ecossistema de Software

MSECO – Ecossistema de Software Móvel

SDK – Kit para Desenvolvimento de Software (em inglês, *Software Development Kit*)

SKO – *Software Knowledge Operation*

SKU – *Software Knowledge Utilization*

UX – Experiência de Usuário (em inglês, *User eXperience*)

LISTA DE FIGURAS

Figura 1. Metodologia de pesquisa.	7
Figura 2. Quantidade de artigos publicados por ano.	22
Figura 3. Versão inicial do modelo DevGo.....	49
Figura 4. Segunda versão do modelo DevGo.....	57
Figura 5. Quantidade de artigos selecionados do Medium por ano.....	59
Figura 6. Quantidade de lições aprendidas extraídas por ano.	59
Figura 7. Distribuição de lições aprendidas por estágio do modelo.....	60
Figura 8. A terceira versão do modelo DevGo.....	63
Figura 9. Matriz com motivações e repositórios da pesquisa de opinião.....	66
Figura 10. Fontes e itens operacionalizadores em DevRel expansão do framework de Amit e Zott (2001).....	76
Figura 11. Repositórios e motivações.....	79
Figura 12. Quarta versão do modelo DevGo – objetos de transferência de valor.	81
Figura 13. Emoções primárias identificadas em cada MSECO (Android, iOS e Windows Phone).	89
Figura 14. Eventos por MSECO e frequência de tags.	100
Figura 15. Diagrama de Veen - Desenvolvedores que mais recebem medalhas.....	103
Figura 16. Estrutura do modelo DevGo.	106
Figura 17. DevGo - modelo para governança de desenvolvedores.....	107
Figura 18. Objetos de transferência de valor – Plataforma e Produtos.	108
Figura 19. Objetos de transferência de valor – DevRel (Evangelismo e Advocacia).....	110
Figura 20. Objetos de transferência de valor – Fluxo de Avanço do Desenvolvedor.	110
Figura 21. Objetos de transferência de valor – Monitoramento.	113
Figura 22. Adequação do DevGo.....	131
Figura 23. Controle do DevGo.	132
Figura 24. Entendimento do DevGo.....	132
Figura 25. Generalidade do DevGo.	133

LISTA DE TABELAS

Tabela 1. Objetivo do estudo de mapeamento sistemático - GQM.	19
Tabela 2. Itens para extração.	22
Tabela 3. Artigos por tipo de ecossistema.	23
Tabela 4. Comparação entre trabalhos relacionados.	41
Tabela 5. Participantes da pesquisa de opinião.	47
Tabela 6. As 13 estratégias consideradas relevantes pelos profissionais de DevRel.	48
Tabela 7. Lista de motivações.	64
Tabela 8. Itens operacionalizadores de fontes de valor.	68
Tabela 9. Tópicos mais frequentes por MSECO.	88
Tabela 10. Android – tristeza – As cinco questões mais visualizadas.	90
Tabela 11. Windows Phone – tristeza – cinco questões mais visualizadas.	90
Tabela 12. iOS – tristeza – cinco questões mais visualizadas.	90
Tabela 13. Android - raiva - cinco questões mais visualizadas.	91
Tabela 14. Windows Phone - raiva - cinco questões mais visualizadas.	92
Tabela 15. iOS - raiva - cinco questões mais visualizadas.	93
Tabela 16. Android - alegria - cinco questões mais visualizadas.	93
Tabela 17. Windows Phone - alegria - Cinco questões mais visualizadas.	94
Tabela 18. iOS - alegria - cinco questões mais visualizadas.	95
Tabela 19. Tópicos quentes para questões não respondidas.	96
Tabela 20. Tópicos quentes - as questões mais respondidas.	98
Tabela 21. Tópicos quentes - questões mais visualizadas.	99
Tabela 22. Frequência de postagens entre fevereiro/2015 e janeiro/2016.	100
Tabela 23. Ranking - número de medalhas conquistadas.	103
Tabela 24. As cinco medalhas mais recebidas em cada MSECO.	103
Tabela 25. Categorias de Repositórios.	111
Tabela 26. Repositório e Motivações.	112
Tabela 27. Caracterização dos Participantes.	129
Tabela 28. Seleção de elementos da área de foco Plataforma e Produtos.	137
Tabela 29. Seleção de elementos da área de foco DevRel (Evangelismo e Advocacia) – Parte 1.	138
Tabela 30. Seleção de elementos da área de foco DevRel (Evangelismo e Advocacia) – Parte 2.	139
Tabela 31. Seleção de elementos da área de foco Fluxo de Avanço do Desenvolvedor – Parte 1.	140
Tabela 32. Seleção de elementos da área de foco Fluxo de Avanço do Desenvolvedor – Parte 2.	141
Tabela 33. Lições aprendidas indicadas para retenção.	141
Tabela 34. Insights para monitoramento indicadas para retenção.	141
Tabela 35. Seleção de elementos da área de foco Monitoramento.	142
Tabela 36. Resumo de resultados para cada critério.	146

Tabela 37. Resumo de resultados para cada critério (cont.).	147
Tabela 38. Perfil dos participantes das entrevistas.....	165

CAPÍTULO 1 - INTRODUÇÃO

Neste capítulo é apresentado o contexto do trabalho e o que motivou esta pesquisa. São também apresentados os seus objetivos, a hipótese, a metodologia de pesquisa e a organização deste texto.

1.1. Contextualização e Motivação

Os relacionamentos entre desenvolvedores, empresas, usuários e organizações que mantêm uma plataforma tecnológica, que passaram a existir com a evolução do desenvolvimento de software direcionaram o cenário de produtos de software para formarem os chamados Ecossistemas de Software (ECOS). Desta forma, fornecedores e consumidores passaram a criar tecnologia de forma colaborativa para gerar valor para contribuidores, usuários e empresas envolvidas em um ECOS [Bosch 2009]. As organizações que expandem suas plataformas por meio de ECOS precisam envolver desenvolvedores externos como forma de atender a demanda de usuários.

No domínio de plataformas móveis, as organizações de software, como Apple e Google, têm investido em uma infraestrutura para recrutar e engajar desenvolvedores externos como forma de contribuir para a expansão da plataforma. Até 2020, é esperada uma renda em torno de 188,9 bilhões de dólares gerada pelo consumo de aplicações móveis¹. Este cenário tem sido explorado como um tipo de ECOS específico para o domínio de aplicações móveis (*apps*) chamado Ecossistema de Software Móvel (MSECO, do inglês *Mobile Software Ecosystem*) [Fontão *et al.* 2015] [Steglich *et al.* 2019].

Dentre os elementos de um MSECO, a *app* consiste em um software que é embarcado em um dispositivo móvel (p.ex.: smartphone, tablet e smartwatch) e adiciona funcionalidades extras ao dispositivo. Uma *app* é projetada, construída e submetida à loja de apps (p.ex.: Google Play) do MSECO pelo desenvolvedor (i.e., um único desenvolvedor, uma equipe, uma startup ou uma empresa). O desenvolvedor se envolve na criação de contribuições, que podem ser, por exemplo, apps, artigos técnicos ou componentes reutilizáveis. Parte dessas contribuições são armazenadas em repositórios proprietários, como a Loja de Apps (p.ex.: Google Play e App Store) e de código (p.ex.: GitHub).

¹ Statista, "App Developers - Statistics & Facts," <https://www.statista.com/topics/1694/app-developers>

Todo o gerenciamento deste fluxo de consumo e produção de contribuições dentro do MSECO é de responsabilidade da organização central. Esse contexto de relacionamento entre os elementos retrata, do ponto de vista de negócios, o MSECO como um tipo de ECOS com característica híbrida, ou seja, utiliza-se de estratégias proprietárias (p.ex.: a loja de apps como repositório das contribuições, os critérios para monetização e qualidade das apps) e *open-source* (p.ex.: eventos para contato próximo com os desenvolvedores e mecanismos de controle da comunidade).

Analisando o cenário dos principais MSECOs², estima-se que há uma quantidade em torno de 12 milhões de desenvolvedores de aplicações móveis no mundo. O desenvolvedor possui responsabilidades e um desempenho esperado pela organização central, de forma que se mantenha a sustentabilidade do ecossistema, ou seja, a capacidade de aumentar ou manter a comunidade de desenvolvedores durante o tempo e sobreviver a mudanças [Manikas e Hansen 2013a].

O bem-estar econômico e social dos desenvolvedores, assim como os objetivos da organização central, devem ser garantidos [Vorraber *et al.* 2018]. Para isso, algumas das organizações possuem um grupo de relacionamentos com o desenvolvedor (DevRel, do inglês *Developer Relations*). Esses relacionamentos que ocorrem entre os desenvolvedores do ecossistema e a organização central, que envolvem complexas atividades, requerem um esforço de coordenação pela organização central que mantém o MSECO por meio de estratégias de governança.

A governança consiste em um modelo que reúne um conjunto de premissas, conceitos, valores e práticas relativos à organização e ao relacionamento entre as partes envolvidas. As organizações utilizam modelos de governança para a atingir suas metas, utilizar melhor os recursos disponíveis e aumentar seus lucros e reduzir eventuais riscos [Jansen e Cusumano 2012]. Quando uma organização central está definindo estratégias de governança de desenvolvedores, o comprometimento com gerenciamento sustentável é um facilitador chave para o engajamento de desenvolvedores [Morisio *et al.* 2002][Sanchez *et al.* 2017].

Pode-se perceber que a governança de desenvolvedores em um MSECO é um processo de evolução contínuo que envolve o entendimento de como uma massa crítica de desenvolvedores (p.ex.: individual, comunidade, startups e empresas) pode ser construída e guiada em um ecossistema.

² <http://www.businessofapps.com/news/12-million-mobile-developers-worldwide-nearly-half-develop-android-first/>

1.2. Descrição do Problema

A governança de desenvolvedores ainda é um desafio para pesquisadores e profissionais que atuam na área de ecossistemas. No que está relacionado a comunidade científica, esta ainda carece de pesquisa, formalização, de teorias e modelos, ferramentas de apoio e estudos experimentais [Manikas 2016][Alves *et al.* 2017]. A seleção de estratégias de governança de desenvolvedores não é uma tarefa trivial, pois o desafio está em delimitar as ações dos desenvolvedores sem restringir excessivamente o nível desejado de inovação e criação de valor no MSECO.

Para a indústria o processo de expansão de uma plataforma por meio de abordagem de ecossistemas exige, por meio da governança de desenvolvedores, criação de novas estratégias que promovam a colaboração da organização central com os desenvolvedores externos [Kude *et al.* 2018]. Como parte dessas estratégias, os profissionais de DevRel precisam ter uma visão realista do comportamento das comunidades de desenvolvedores dentro do ecossistema. Desta forma será possível à organização central atender às necessidades dos desenvolvedores e atingir metas do ecossistema [Huber *et al.* 2017].

Se a organização central não possui estratégias sustentáveis para governar desenvolvedores, a organização corre o risco de falhar no atendimento das demandas dos usuários (p.ex.: número, qualidade e variedade de contribuições). Uma governança de desenvolvedores insustentável durante o tempo pode resultar na “morte” do ecossistema [Wareham *et al.* 2014][Kude 2018]; por exemplo, o MSECO Windows Phone foi declarado oficialmente “morto” em 2017 porque os desenvolvedores não apoiaram a plataforma³. As organizações que mantêm MSECOS podem passar por dificuldades ao tentar conectar os objetivos da organização com as expectativas do desenvolvedor, além de monitorar os efeitos das ações de governança de desenvolvedores [Mäenpää *et al.* 2017].

1.3. Hipótese

A carência por modelos, o apoio para a definição e seleção de estratégias sustentáveis, a necessidade de visão realista da relação cooperativa entre a organização central e a massa crítica de desenvolvedores externos levaram à seguinte hipótese a ser investigada:

³ <https://www.theverge.com/2017/10/9/16446280/microsoft-finally-admits-windows-phone-is-dead>

Um modelo envolvendo um conjunto de elementos estruturais e lições aprendidas impacta positivamente na percepção de profissionais de DevRel para a governança de desenvolvedores em MSECO.

A percepção dos profissionais de DevRel, em relação ao modelo, será medida em relação aos seguintes critérios, baseadas em um conjunto de critérios definidos por Strauss e Corbin (1998) para garantir objetividade durante a avaliação de teorias: adequação, entendimento, generalidade e controle. A descrição de cada um dos critérios é apresentada no Capítulo 6, na Seção 6.1.3. A partir da hipótese surgem as questões de pesquisas apresentadas na próxima seção.

1.4. Questões de Pesquisa

A partir do que foi apresentado na seção anterior, esta pesquisa tem como objetivo investigar e responder à seguinte questão:

O uso de um modelo por profissionais de DevRel para governar desenvolvedores no contexto de Ecossistema de Software Móvel (MSECO) é adequado, controlável, entendível e generalizável?

Como forma de responder à questão de pesquisa principal, apresentada acima, foram definidas as seguintes questões:

- [QP1] O que a literatura científica diz sobre a governança de desenvolvedores em ECOS?
- [QP2] Qual a relevância das estratégias de governança de desenvolvedores para profissionais de DevRel?
- [QP3] Quais são os elementos que fazem parte da estrutura de um modelo para a governança de desenvolvedores em ECOS?
- [QP4] Quais são as lições aprendidas compartilhadas por profissionais de DevRel para a governança de desenvolvedores?
- [QP5] Quais os repositórios externos e internos à organização que podem ser utilizados para o monitoramento em governança de desenvolvedores em MSECO?
- [QP6] Qual a percepção dos profissionais sobre valor em DevRel e quais as motivações que levam o profissional de DevRel a usar os repositórios durante atividades de governança de desenvolvedores em MSECO?
- [QP7] O que pode ser entendido a partir dos principais MSECOS com base em questões técnicas em um repositório de Perguntas e Respostas?

As questões QP1 a QP4 se concentram em obter uma visão de governança a partir da perspectiva das organizações que mantêm ecossistemas favorecendo o entendimento dos objetivos organizacionais. As próximas questões, QP5 a QP7, focadas em monitoramento da massa crítica de desenvolvedores externos, capturam o valor de repositórios para governança de desenvolvedores em MSECO e como os repositórios podem ser utilizados para apoiar a governança de desenvolvedores. A questão principal, também, é respondida por meio de uma validação do modelo e do conjunto de lições aprendidas por profissionais de DevRel que atuam em MSECO.

Considerando as revisões de literatura técnica existentes sobre ECOS [Barbosa *et al.* 2013][Manikas e Hansen 2013b][Manikas 2016] [Franco-Bedoya *et al.* 2017], governança de ECOS [Alves *et al.* 2017], MSECO [Fontão *et al.* 2015][Steiglich *et al.* 2019], e repositórios de Q&A [Farias *et al.* 2016], há uma indicação de que os repositórios podem ser utilizados para extrair informações sobre a perspectiva sócio-técnica de um ECOS. E, por isto, o repositório analisado dentro da QP7 foi da categoria de Perguntas & Respostas, mais especificamente, o Stack Overflow⁴.

1.5. Objetivos

1.5.1. Objetivo Geral

Estabelecer um modelo envolvendo elementos estruturais e um conjunto de lições aprendidas para a governança de desenvolvedores em MSECO que seja adequado, entendível, controlável e generalizável para apoiar estratégias de profissionais de DevRel.

1.5.2. Objetivos Específicos

Para atingir o objetivo geral desta pesquisa, pretende-se alcançar os seguintes resultados intermediários:

- **Definição de um corpo de conhecimento sobre Governança de Desenvolvedores em ECOS:** contendo as estratégias para governar desenvolvedores identificadas na literatura técnica e sua relevância para os profissionais;
- **Identificação e análise da estrutura para governança de desenvolvedores:** a partir da perspectiva de DevRel e, assim, contribuir para a construir a estrutura do modelo;

⁴ <http://www.stackoverflow.com>

- **Definição de um corpo de conhecimento com lições aprendidas em DevRel:** com o objetivo de identificar um conjunto de lições aprendidas por profissionais de DevRel que serão associadas a partes do modelo;
- **Identificação de um mecanismo de monitoramento:** com o objetivo de identificar que repositórios podem ser utilizados, por qual motivo são utilizados por profissionais de DevRel e como podem fornecer informações para favorecer o feedback entre organização central e desenvolvedores.

1.6. Metodologia de Pesquisa

A metodologia de pesquisa utilizada neste trabalho (**Figura 1**) foi conduzida pelas questões de pesquisas descritas na Seção 1.4. Considera-se, nesta tese, a partir de diretrizes da Engenharia de Software Experimental um conjunto de métodos visando criar bases mais sólidas para a definição e validação do modelo.

O modelo proposto nesta tese é chamado DevGo (do inglês, *Developer Governance*). Na **Figura 1**, são apresentados os métodos utilizados para responder a cada uma das questões de pesquisa apresentadas na Seção 1.4. Abaixo são descritos os métodos utilizados para cada questão, assim como, os resultados obtidos.

Método para QP1: mapeamento sistemático da literatura existente sobre Governança de Desenvolvedores em ECOS, seguindo os princípios de um mapeamento sistemático indicados por [Kitchenham e Charters 2007] e [Petersen *et al.* 2015]. O protocolo definido, os procedimentos de execução e a análise dos resultados do mapeamento sistemático podem ser vistos no Capítulo 2.

Resultado para QP1: uma definição e um conjunto de estratégias para governança de desenvolvedores em ECOS.

Método para QP2: pesquisa de opinião, seguindo os procedimentos indicados por Molléri *et al.* (2016) e Linaker *et al.* (2015), com especialistas (profissionais de DevRel) sobre a relevância das estratégias para governança de desenvolvedores extraídas da literatura técnica.

Resultado para QP2: o conjunto de estratégias consideradas relevantes pelos profissionais de DevRel foi utilizado para a construção da primeira versão do modelo DevGo como forma de identificar a estrutura que apoia a governança de desenvolvedores em ecossistemas.

Método para QP3: um conjunto de entrevistas com profissionais de DevRel como forma de análise e refinamento do modelo DevGo.

Resultado: a segunda versão do modelo DevGo.

Método para QP4: a análise de artigos de profissionais de DevRel no *Medium*⁵, portal que contém artigos técnicos. Para isto, utilizou-se de procedimentos de mapeamento sistemático de literatura cinza propostos por Garousi *et al.* (2016).

Resultado para QP4: um conjunto de lições aprendidas classificadas em cada estágio do modelo DevGo.

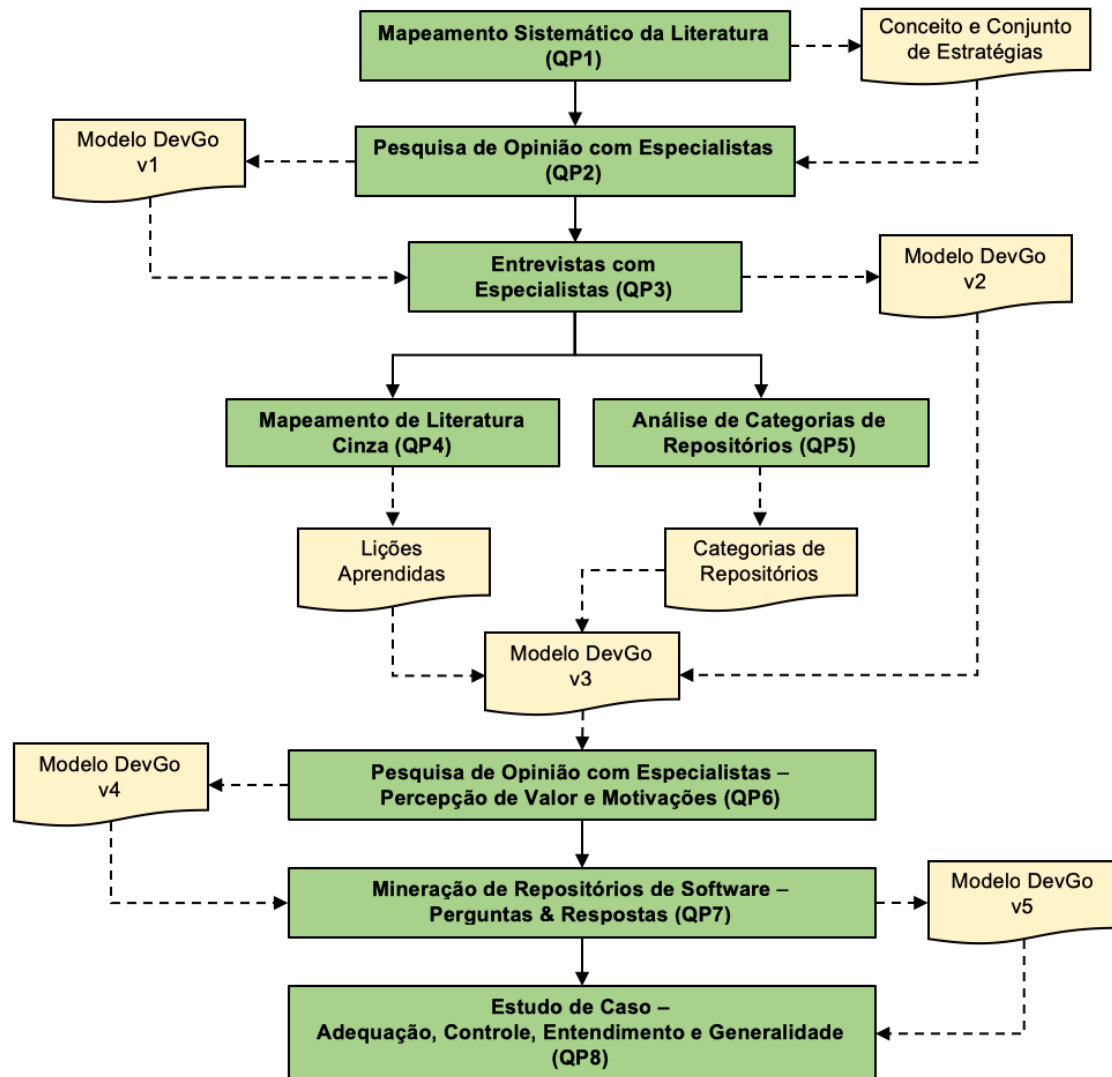


Figura 1. Metodologia de pesquisa.

Método para QP5: um estudo de caso exploratório para analisar a estrutura de repositórios internos e externos de MSECOs seguido de uma revisão por pares com especialistas com objetivo de caracterizar os repositórios em relação às suas características comuns.

Resultado para QP5: um conjunto de categorias agrupando um conjunto de repositórios internos e externos a organização central. Os resultados obtidos em QP5 ajudaram no refinamento do DevGo gerando assim a terceira versão do modelo que

⁵ <http://medium.com>

passou a conter as lições aprendidas por estágios e as categorias de repositórios para monitoramento.

Método para QP6: uma pesquisa de opinião com profissionais visando obter uma lista de motivações mais frequentes para cada categoria de repositórios identificadas previamente. Esta pesquisa também continha uma questão aberta para coletar a percepção sobre o que é valor em DevRel. Para analisar os comentários dos participantes foi utilizado como base o framework de Amit e Zott (2001) que trata sobre aspectos de valor em negócios. Este framework foi utilizado por Hyrynsalmi *et al.* (2014) para analisar as fontes de valor em um MSECO a partir da perspectiva de desenvolvedores de aplicações móveis.

A perspectiva de valor está relacionada a investigação da dimensão de negócios, como indicado no framework ReuseECOS [Santos e Werner 2012]. Esta dimensão foca no fluxo de conhecimento e a criação de valor. Este estudo aprimorou os objetos de transferência de valor do DevGo contribuindo para o refinamento e desta forma a quarta versão do modelo.

Resultado para QP6: um conjunto de motivações mais frequentes associadas a cada categoria de repositórios. Um conjunto de comentários dos profissionais e de itens relacionados a percepção sobre valor em DevRel categorizados em torno de quatro fontes de criação de valor: (1) Eficiência; (2) Retenção; (3) Inovação; (4) Complementaridade. Este estudo permitiu aprimorar os objetos de transferência de valor do DevGo e o conjunto de lições aprendidas.

Método para QP7: Mineração de Repositórios de Software (MRS). A MRS foca em descobrir informação útil sobre os elementos circundantes ao software por meio da extração e análise de dados de repositórios. Nesta tese, foram utilizados mecanismos de MRS para obter informações sobre o desenvolvedor a partir das outras duas dimensões de análise de um ecossistema, como segue. Esta etapa da pesquisa buscou explorar as dimensões técnica e social de um ECOS, segundo o framework ReuseECOS [Santos e Werner 2012], como apresentado anteriormente.

- *Dimensão técnica:* análise de tópicos frequentes em: questões que foram respondidas e questões sem resposta. Esta dimensão foca na plataforma do ecossistema e sua infraestrutura, assim é possível definir o ciclo de vida e as características da plataforma;
- *Dimensão social:* reputação dos desenvolvedores e perguntas em que os desenvolvedores mais se engajam. Esta dimensão envolve os relacionamentos entre os atores do ecossistema, o entendimento das motivações e como eles cooperam ou competem.

Resultado para QP7: um conjunto de insights obtido a partir da análise dos resultados que aprimoram a área de monitoramento do DevGo. Isto contribuiu para aprimorar o modelo para uma quinta versão.

Método para QP8: um estudo de caso do tipo confirmatório com entrevistas envolvendo coleta de dados. A amostra de participantes deste estudo está dentro da população de profissionais de DevRel que atuam em governança de desenvolvedores em MSECO. Para este estudo o foco esteve no estágio de Retenção do DevGo.

Resultado para QP8: a validação do quanto o modelo DevGo representa a visão e perspectiva dos profissionais de DevRel para a governança de desenvolvedores em MSECO por meio da definição dos elementos gerais do modelo incluindo as lições aprendidas.

1.7. Contribuições da Tese

1.7.1. Principais Contribuições

Há um conjunto de contribuições durante a definição e validação do modelo DevGo:

- **Um conceito de governança de desenvolvedores em ECOS:** descrito no Capítulo 2, há uma síntese de conceito geral para a governança de desenvolvedores em ECOS, obtida a partir da análise do conceito para categorias (proprietário, código aberto e híbrido) de ECOS;
- **Um conjunto de estratégias para governança de desenvolvedores em ECOS:** as estratégias foram obtidas a partir da análise do mapeamento sistemático da literatura e, também, são descritas no Capítulo 2 por categorias de ECOS;
- **O modelo para governança de desenvolvedores em MSECO – DevGo:** descrito no Capítulo 5, para a governança de desenvolvedores em MSECO, contendo: áreas de foco, fases e estágios do fluxo de avanço de desenvolvedores, facilitadores de atividades, objetos de transferência de valor, repositórios e insights para monitoramento;
- **Um conjunto de lições aprendidas de DevRel para governança de desenvolvedores:** cada lição aprendida foi associada a um dos estágios do DevGo, desta forma elas são apresentadas no Capítulo 5. Essas lições aprendidas foram indicadas por profissionais de DevRel que atuam com governança de desenvolvedores em ECOS.

O modelo DevGo, incluindo o conjunto de lições aprendidas, tem como elementos beneficiados dentro de um MSECO: a organização central, que é interessada na expansão do ecossistema, atração e engajamento de desenvolvedores, e os profissionais de DevRel, que atuam mais diretamente nas atividades de governança de desenvolvedores. O DevGo pode ajudar no entendimento sobre a estrutura que a apoia a governança de desenvolvedores e prover um conjunto de lições aprendidas, para:

1. Guiar pequenas e médias organizações na construção de ecossistemas que envolvam desenvolvedores;
2. Apoiar grandes organizações na evolução de estratégias de governança de desenvolvedores;
3. Apoiar pesquisadores e profissionais da área na estruturação da atividade de governança de desenvolvedores e na condução de pesquisas futuras.

1.7.2. Outros Resultados

Nesta seção são descritas as contribuições que envolvem artigos científicos em conferências, periódicos e workshops. Além disso, apresentam-se os minicursos, capítulos de livro, atuações como revisor de periódicos e conferências. Ainda são descritas contribuições que envolvem a docência: orientações e disciplinas. As premiações obtidas durante o doutorado, também, fazem parte desta seção.

- **Artigos em conferência**

1. GoDev-DX: Governança de Desenvolvedores em Ecossistemas de Software Móvel a partir da Experiência do Desenvolvedor [Fontão e Dias-Neto 2016];
2. Facing up the primary emotions in Mobile Software Ecosystems from Developer Experience [Fontão *et al.* 2017b];
3. Towards a Guideline-Based Approach to Govern Developers in Mobile Software Ecosystems [Fontão *et al.* 2017a];
4. Hearing the Voice of Developers in Mobile Software Ecosystems [Fontão *et al.* 2017c];
5. Challenges in Model Based Testing for Mobile Applications [Ascate *et al.* 2017];
6. Tester Experience: Concept, Issues and Definition [Ekwoje *et al.* 2017];
7. Investigating factors that influence developers experience in mobile software ecosystems [Fontão *et al.* 2017d];
8. Mobile Application Development Training in Mobile Software Ecosystem: Investigating the Developer eXperience [Fontão *et al.* 2018a];

9. Which factors affect the evangelist's support during training sessions in Mobile Software Ecosystems? [Fontão *et al.* 2018b];
10. Governança de desenvolvedores em Ecossistemas de Software Móvel – Workshop de Teses e Dissertações do CBSOFT 2018.

- **Artigos em periódicos**

1. Exploiting Repositories in Mobile Software Ecosystems from a Governance Perspective [Fontão *et al.* 2018c];
2. Supporting governance of mobile application developers from mining and analyzing technical questions in stack overflow [Fontão *et al.* 2018d];
3. Developer Governance in Software Ecosystem: A Systematic Mapping Study (*ACM Computing Surveys* – *Em revisão*).

- **Capítulos de livro**

1. Governança de desenvolvedores em Ecossistemas de Software [Fontão *et al.*, 2018e];
2. Towards a guideline-based approach to govern developers in mobile software ecosystems [Fontão *et al.*, 2018f];
3. Ecossistemas de Software. Livro Computadores e Sociedade da Série Elsevier - SBC.

- **Minicursos**

1. Certificação de Apps em Ecossistemas de Software Móvel (CBSOFT 2017);
2. Governança de Desenvolvedores em Ecossistemas de Software (SBSI 2018).

- **Premiações**

1. *Best Paper Award (PhD, Poster Category)*: "GoDev-DX: Developers' Governance in Mobile Software Ecosystems from Developer Experience (DX)", The III Latin-American School on Software Engineering (ELA-ES) - Sociedade Brasileira de Computação;
2. Entre os 5 melhores artigos da "Research Track", Simpósio Brasileiro em Engenharia de Software – 2017;
3. Reconhecimento pela produtividade no biênio 2017-2018, Instituto de Computação - Universidade Federal do Amazonas.

1.8. Estrutura da Tese

Este trabalho está organizado em seis capítulos, sendo este o primeiro capítulo de introdução, que apresentou a motivação, problema, objetivos e metodologia e o contexto da pesquisa. A organização do texto deste trabalho segue a estrutura abaixo:

- **Capítulo 2: “Referencial teórico”:** descreve os principais conceitos relacionados à pesquisa: ECOS, MSECO e governança de ecossistemas. Além disso, é apresentado o corpo de conhecimento sobre governança de desenvolvedores, que foi obtido por meio de mapeamento sistemático da literatura. Este corpo de conhecimento consiste em definições de governança de desenvolvedores para os seguintes tipos de ECOS: *open-source*, proprietário e híbrido. E, também, de um conjunto de estratégias para governança de desenvolvedores.
- **Capítulo 3: “Estudos para concepção e refinamento do DevGo: modelo de governança de desenvolvedores”:** apresenta os resultados dos estudos que ajudaram na concepção e refinamento do DevGo: (1) Pesquisa de opinião sobre relevância de estratégias presentes no corpo de conhecimento; (2) Entrevistas com profissionais para a análise da estrutura do DevGo; (3) Identificação de lições aprendidas para a governança de desenvolvedores a partir de DevRel; (4) Estudo Exploratório definição de categorias de de repositórios; (5) Pesquisa sobre percepção de valor;
- **Capítulo 4: “Capturando as expectativas dos desenvolvedores”:** descreve resultados de análise de informações do Stack Overflow, da categoria de repositório de Perguntas e Respostas, sobre contribuições de desenvolvedores dos MSECOs (Android, iOS e Windows Phone);
- **Capítulo 5: “DevGo: modelo para governança de desenvolvedores em MSECO”:** apresenta a estrutura e a descrição dos elementos que compõem o modelo proposto neste trabalho. São apresentadas também as lições aprendidas na perspectiva de DevRel e um conjunto de insights para monitoramento da governança de desenvolvedores em MSECO;
- **Capítulo 6: “Estudo de Caso – retenção de desenvolvedores”:** neste capítulo é descrito o planejamento, execução e análise de resultados para a validação do modelo com profissionais. O caso estudado envolve a retenção de desenvolvedores em MSECO;
- **Capítulo 7: “Considerações finais”:** neste capítulo são apresentadas as contribuições já geradas, as conclusões frente aos trabalhos existentes, as limitações e os trabalhos futuros.

CAPÍTULO 2 - REFERENCIAL TEÓRICO

Neste capítulo são apresentados os conceitos relacionados ao desenvolvimento da pesquisa, a fundamentação teórica que embasa este trabalho e as pesquisas correlatas encontradas na literatura.

2.1 Ecosistema de Software (ECOS) e de Software Móvel (MSECO)

Segundo Bosch (2009), ECOS consiste de uma plataforma de software, um conjunto de desenvolvedores internos e externos e uma comunidade de especialistas em serviço à comunidade de usuários que constroem solução de elementos relevantes para satisfazer a necessidade deles. Os tipos de elementos que compõem um ECOS, seus papéis e atividades são descritos por [Jansen *et al.* 2013] e, no contexto deste trabalho, destacamos:

- *Plataforma:* termo que se refere a padrão de arquitetura, protocolo de comunicação ou qualquer conhecimento fundamental e compartilhado. Uma plataforma é a base na qual vários elementos técnicos de um ecossistema são construídos. Provê o suporte para a customização em larga escala;
- *Organização Central:* provê uma plataforma tecnológica; define e compartilha padrões e práticas; e aumenta o valor da plataforma e seus produtos por meio do aumento dos contribuidores;
- *Contribuidor:* faz uso da plataforma disponibilizada pela organização geral para gerar valor a partir de padrões bem definidos. Um desenvolvedor, por exemplo, é um tipo de contribuidor em um ECOS que adota a plataforma e publica extensões (p.ex.:, apps), essas contribuições mudam as características da plataforma e favorecem a criação de comunidades.

Manikas (2016) ainda ajuda a classificar os ECOS a partir de uma perspectiva de meios de criação de valor, como segue:

- *Proprietário:* onde a criação de valor no ecossistema é baseada em contribuições proprietárias. Normalmente protegido por processos de gerenciamento de propriedade intelectual e o valor refere-se à compensação monetária. Industrial, Plataforma como Serviço e ecossistemas de comércio eletrônico são exemplos desta categoria;

- *Open-Source*: as contribuições estão abertas ao resto dos atores ou ao público. Os valores se referem a compensações não monetárias (p.ex.: conhecimento e experiência ou precisa de satisfação). A Fundação Eclipse, o Gnome, a Fundação Apache e os ecossistemas do governo público são exemplos;
- *Híbrido*: suporta contribuições proprietárias e de código aberto. Por exemplo, os ecossistemas de software móvel se utilizam de estratégia proprietária (a loja de apps, um repositório controlado pela organização) e de estratégias open-source para o engajamento da comunidade (acesso a ferramentas, submissão e publicação de contribuições). Os seguintes ecossistemas foram adicionados a esta categoria: ecossistemas de Android, iOS, Windows Phone e software móvel.

Um ECOS especificamente voltado para apps e híbrido é estudado na engenharia de aplicações móveis como Ecossistema de Software Móvel (MSECO, do inglês *Mobile Software Ecosystem*), onde a loja de apps além de ser um ambiente utilizado para armazenar e acessar apps também serve para coletar as opiniões dos usuários sobre a qualidade do produto. Em um MSECO, empresas precisam agregar valor a seus produtos, neste caso, os dispositivos móveis, por meio das apps disponibilizadas em suas lojas ou da extensão da plataforma por APIs que serão utilizadas em apps. E há uma grande necessidade pelo engajamento de uma massa crítica de desenvolvedores externos a estrutura da organização. Pois estas empresas não conseguem atender toda a demanda dos usuários somente com sua estrutura interna [Eklund e Bosch 2014][Song *et al.* 2018].

No contexto apresentado anteriormente, as comunidades de desenvolvedores atuam diretamente na solução para a crescente demanda dos usuários de um MSECO [Rastogi e Nagappan 2016][Constantinou e Mens 2017]. Por outro lado, percebe-se uma limitação destes desenvolvedores, visto que eles não são terceirizados e não possuem vínculo empregatício. No máximo, estes desenvolvedores possuem um vínculo emotivo à plataforma/marca, sabendo que o seu envolvimento dentro daquele MSECO pode ajudar em seu desenvolvimento profissional. Além dos elementos já descritos acima como parte de um ECOS e que também compõem um MSECO, abaixo descrevemos alguns dos elementos que compõem um MSECO [Fontão *et al.* 2015] temos:

- *Usuário*: representa o núcleo de um MSECO. É responsável por adquirir conteúdo de uma loja de apps e criar novas formas de interação. Ao adquirir e utilizar essas apps, o usuário também pode prover feedback para o

desenvolvedor da app. O número de usuários ativos em um MSECO reflete a satisfação dos usuários;

- *Desenvolvedor*: responsável por novas ideias, desenvolvimento de apps respondendo aos requisitos dos usuários, produzindo material técnico que suporte o desenvolvimento de apps ou liderando comunidades de desenvolvedores;
- *Comunidade*: uma estrutura de colaboração e coordenação de atividades composta por contribuidores internos e externos. Existem comunidades específicas, como: as comunidades de usuários e as comunidades de desenvolvedores;
- *Aplicações Móveis (apps)*: um artefato produzido por desenvolvedor(es) que é adquirido e avaliado por usuários. É uma aplicação individual que representa uma unidade básica que pode ser agrupada e categorizadas. Uma app deve obedecer a padrões mínimos de qualidade impostos pelas organizações;
- *Loja de Apps*: um canal de distribuição utilizado para publicar, armazenar, buscar e recuperar aplicações móveis. É um mercado altamente competitivo onde desenvolvedores precisam prover conteúdo para um grande número de usuários;
- *Evangelista ou profissional de DevRel (Relações com Desenvolvedores, do inglês Developer Relations)*: participa em treinamentos, palestras e competições de desenvolvimento com o objetivo de ajudar na expansão do MSECO e na formação de novos desenvolvedores. Este ator do ecossistema faz parte da comunidade de especialistas e é um funcionário interno da organização do MSECO. É um especialista em um campo específico e tem conhecimento das atividades dentro do MSECO.

Os ECOS podem ser analisados a partir das seguintes dimensões: técnica, social e de negócios. Essas dimensões fazem parte do framework ReuseECOS proposto por Santos e Werner (2012). A *dimensão técnica* foca na plataforma do ecossistema e sua infraestrutura, assim é possível definir o ciclo de vida e as características da plataforma. A *dimensão social* envolve os relacionamentos entre os atores do ecossistema, o entendimento das motivações e como eles cooperam ou competem. Por fim, a *dimensão de negócios* foca no fluxo do conhecimento e a criação de valor.

Todo esse contexto favoreceu com que o processo de inovação em várias áreas de negócio deslocasse o processo tradicional de inovação, onde o desenvolvimento de um produto passou de totalmente fechado dentro de uma

organização a um ambiente onde a participação de contribuidores externos exerce um importante papel [Chesbrough 2003][Fontão *et al.* 2018c][Vorraber *et al.* 2018].

Como forma de apoiar as dimensões visando o desenvolvimento, estabelecimento e valor da plataforma, a organização central emprega mecanismos de governança para estabelecer níveis de controle, direitos sobre decisões e o escopo de propriedade e propriedade compartilhada.

2.2 Governança de Ecossistema de Software

A **governança** é definida como a forma que uma organização é gerenciada, incluindo suas responsabilidades e processos de apoio à decisão [Dubinsky e Kruchten 2009]. A governança envolve a atribuição de responsabilidades e direitos de decisões, assim como, os indicadores e políticas que permitem a avaliação contínua. Carina *et al.* (2017) definem **governança de ECOS** como ferramentas gerenciais dos atores do ecossistema que têm o objetivo de influenciar a sustentabilidade do ecossistema. A sustentabilidade é a capacidade de aumentar ou manter a comunidade de desenvolvedores ao longo do tempo e sobreviver a mudanças [Manikas 2016].

O gerenciamento e monitoramento bem-sucedidos ainda são grandes desafios para profissionais de ecossistemas [Dhungana *et al.* 2010][Vorraber *et al.* 2018]. Isto acontece porque a comunidade de ecossistemas ainda carece de teorias específicas de gerenciamento, ferramentas de suporte e experiência consolidada na área [Manikas 2016]. Neste cenário, há três principais categorias de mecanismos de governança [Alves *et al.* 2017]: 1) Criação de valor – gerar e distribuir valor; 2) Coordenação – manter a consistência e integração de atividades, relacionamentos e estruturas do ecossistema; e, 3) Controle e abertura organizacional – capturar a atenção entre modelos abertos e fechados.

Jansen *et al.* (2012) ainda indicam que é necessário buscar a clareza da governança dentro do ecossistema. Para isso os autores apontam partes da governança que ajudam a estabelecer esta clareza, que são:

1. *Clareza do ecossistema*: há questões essenciais para o sucesso de um ecossistema, se um ecossistema não for claro para os elementos que o compõe não é possível ter uma estratégia de governança clara;
2. *Clareza da estratégia de governança*: as organizações podem estabelecer regras, procedimentos, protocolos e processos formalizados, o que pode ajudar no controle do ecossistema. Isto ajuda a estabelecer uma estratégia de governança clara;

3. *Responsabilidade*: os elementos que compõem o ecossistema são responsáveis por ele, se não houver execução correta das responsabilidades de cada elemento uma estratégia de governança não poderá ser garantida;
4. *Medição*: com o objetivo de determinar o benefício para o ecossistema a partir da utilização de uma estratégia de governança, utilizando-se de indicadores para analisar o estado atual do ecossistema e prospectar o estado futuro;
5. *Compartilhamento de conhecimento*: é um aspecto importante do núcleo de negócios de um ecossistema como parte de uma estratégia de governança.

Baars e Jansen (2012) dizem que a governança pode ajudar uma empresa a atingir seus objetivos, fazer melhor uso dos recursos disponíveis e direcionar a um aumento na renda e na redução de riscos. A governança de ecossistema requer um equilíbrio de controle pela organização central e de autonomia entre os desenvolvedores que são externos à organização [Tiwana *et al.* 2010][Bettenburg *et al.* 2013][Steinmacher *et al.* 2015][Song *et al.* 2018]. Entretanto, uma vez que este é um campo relativamente novo, muitas organizações podem não saber como gerenciar efetivamente seus ecossistemas, ou como iniciar um ecossistema. Não existe uma formalização adequada para a governança do ecossistema e há muitos desafios a serem superados pelas organizações [Jansen *et al.* 2009], por exemplo, a atração e o envolvimento dos desenvolvedores. Este envolvimento pode ser analisado a partir da perspectiva de governança de desenvolvedores.

2.3 Governança de Desenvolvedores em ECOS

Nesta seção, apresenta-se o corpo de conhecimento sobre governança de desenvolvedores em ECOS. Para isto foi planejado e executado uma revisão da literatura existente com procedimentos de mapeamento sistemático.

2.3.1 Necessidade e revisões sistemáticas anteriores

As revisões de literatura existentes sobre ECOS e governança de ECOS não revelam estudos que investigam a governança dos desenvolvedores mesmo que haja uma indicação da necessidade de entendê-la [Manikas 2016][Alves *et al.* 2017]. Nesta seção, os estudos de revisão de literatura são discutidos do ponto de vista da governança de ECOS. Os estudos foram organizados cronologicamente para ajudar a compreender o avanço das necessidades de pesquisa.

Barbosa *et al.* (2013) analisaram 44 estudos. Este estudo mapeou o que se conhece sobre a perspectiva dos ECOS. Os autores concluem que a pesquisa de ECOS está concentrada em 8 áreas principais, nas quais os mais relevantes são

software de código aberto, modelagem de ecossistemas e questões de negócios. Os autores não exploraram a área de negócios e não investigaram o conceito de governança dentro dos ECOS.

Fontão *et al.* (2015) identificaram 28 estudos relacionados ao MSECO. Neste estudo, os autores destacaram os principais benefícios do MSECOS: atrair e apoiar os desenvolvedores, ajudando-os a aprender e criar conteúdo e a loja de apps. Os autores discutem que a adoção de uma plataforma por desenvolvedores externos é essencial para o seu crescimento. No entanto, não há discussão sobre mecanismos de governança envolvendo desenvolvedores.

Manikas (2016) analisou 231 trabalhos e fornece uma visão geral da pesquisa em ECOS. Neste estudo, o autor indica a governança como conceito emergente na literatura sobre ECOS. O estabelecimento de teorias, métodos e ferramentas específicas (por exemplo, governança) para problemas do ecossistema são essenciais para a evolução do campo. O autor também motiva para estudos que se concentrem em estudar os aspectos de um ECOS em profundidade e focando em um único aspecto (por exemplo, governança).

Alves *et al.* (2017) analisaram 63 estudos sobre a governança de ECOS, enquadrando definições, classificando os mecanismos de governança e propondo uma agenda de pesquisa. Neste contexto, os autores sugerem o estabelecimento de um glossário comum e quadro conceitual que colete as definições de governança em torno do conhecimento de governança dos ECOS. Existe uma necessidade de conhecimento mais consumível e prático para os profissionais. Os autores apontaram a necessidade de entender a governança dos desenvolvedores, porque as organizações devem entender as motivações e expectativas dos desenvolvedores para adotar mecanismos de governança adequados.

Franco-Bedoya *et al.* (2017) analisaram 82 artigos para avaliar o estado da arte em ECOS *open-source*. Neste estudo, os autores identificaram a seguinte oportunidade de pesquisa: abordar conceitos importantes como controle organizacional, autorregulação do ecossistema, organização da rede, papel da tecnologia e compartilhamento de valores.

Steiglich *et al.* (2019) identificaram 63 estudos relacionados a MSECO, por meio de um mapeamento sistemático, visando atualizar o corpo de conhecimento sobre a área. Os autores concluem que, apesar da relevância de MSECO, o crescimento de investigação ainda é baixo. As comunidades de desenvolvedores e os evangelistas ainda são áreas pouco estudadas.

Considerando os estudos mencionados acima, não há indícios de estudos sistemáticos de revisão/mapeamento da literatura que investiguem a governança dos

desenvolvedores em ECOS. No entanto, há uma indicação da necessidade de investigar a governança de contribuidores específicos em ECOS (por exemplo, desenvolvedores). Neste estudo é investigada a governança dos desenvolvedores em ECOS, então foi realizado um estudo de mapeamento sistemático e, na próxima seção, discute-se o planejamento, execução e análise de resultados.

2.3.2 Protocolo do Mapeamento Sistemático

2.3.2.1 Objetivo e Questões de Pesquisa

Este estudo de mapeamento sistemático foi realizado considerando as recomendações descritas em [Kitchenham e Charters 2007] e [Petersen *et al.* 2015]. O objetivo principal deste estudo foi estruturado usando a abordagem (GQM, *Goal-Question-Metric*) [Basili 1994] apresentada na **Tabela 1**.

Tabela 1. Objetivo do estudo de mapeamento sistemático - GQM.

Analisar	A governança de desenvolvedores em ECOS
Com o propósito de	Caracterizar
Em respeito a	Definições e Estratégias
Do ponto de vista de	Pesquisadores
No contexto de	Estudos acadêmicos em Engenharia de Software

Então, o desafio de entender e selecionar as estratégias do ecossistema que contribuem para a governança dos desenvolvedores direcionou a realização de um mapeamento sistemático da literatura técnica para responder à pergunta de pesquisa: "*O que a literatura científica diz sobre a governança dos desenvolvedores em ECOS?*". Optou-se por envolver ECOS de uma forma geral para capturar estratégias aplicáveis em MSECO pela característica híbrida. Para abordar o objetivo apresentado e responder à principal questão de pesquisa, três questões de pesquisa (QPs) foram definidas:

QP1. Em quais tipos de ECOS os estudos estão focados?

Razão: um tipo de ECOS pode se referir a uma categoria específica de ecossistema de uma perspectiva comercial (ou seja, propriedade, *open-source* e híbrido). Ao responder a esta questão de pesquisa é possível obter informações sobre quais tipos de ECOS têm sido mais investigados do ponto de vista da governança de desenvolvedores. Isto também pode indicar lacunas de pesquisa e levar a futuras orientações de pesquisa.

QP2. Qual é a definição de “governança de desenvolvedores”?

Razão: identificar o que é o termo “governança de desenvolvedores” utilizado em estudos primários que apresentam uma estratégia para governar desenvolvedores em ECOS. As respostas para esta QP ajudam a identificar alguns pontos de vista comuns/conflitantes sobre a governança dos desenvolvedores em ECOS.

QP3. Quais estratégias são usadas na governança dos desenvolvedores?

Razão: obter uma visão geral das estratégias que foram propostas, definidas e empregadas para a governança dos desenvolvedores. Esta resposta ajuda os pesquisadores a identificar vantagens e lacunas em estratégias para governar os desenvolvedores em ECOS e podem apoiar o profissional de DevRel em quais estratégias eles podem usar nas atividades de governança de desenvolvedores.

2.3.2.2 Estratégia de Busca

Como forma de melhorar a *string* de busca, foram utilizadas as seguintes abordagens, considerando as recomendações descritas por [Petersen *et al.* 2015]. O protocolo do mapeamento sistemático foi revisado por três pesquisadores com alta experiência em engenharia de software experimental e no planejamento/execução de mapeamentos ou revisões sistemáticas da literatura. Após isso, foram realizados 10 ciclos de execução da *string* de busca para refinamentos na Scopus. Para aprimorar a *string*, foram utilizados 8 artigos de controle e foram identificados sinônimos para as palavras-chave no *Theasurus*⁶ anteriormente identificadas.

2.3.2.3 String de Busca, bibliotecas digitais e busca manual

A *string* de busca utilizada para a identificação de estudos primários foi definida por meio do uso do critério PICO (População, Intervenção, Comparação e Resultados) [Kitchenham e Charters 2007]. A população neste mapeamento é Ecossistema de Software. A intervenção é 'desenvolvedor'. A comparação foi suprimida, pois esse estudo consiste num mapeamento do corpo de conhecimento existente sobre governança de desenvolvedores. O resultado (no inglês, *outcome*) é governança. A *string* de busca é definida como segue:

```
("software ecosystem" OR "software ecosystems" OR "software  
supply network" OR "software vendor" OR "software vendors" OR  
"software supply industry")  
AND  
(developer OR developers OR programmer OR programmers OR  
engineer OR engineers)  
AND  
(governance OR coordination OR conduction OR administration  
OR control OR direction OR manage OR management OR guidance)
```

⁶ <https://www.thesaurus.com>

A pesquisa foi realizada nos bancos de dados do IEEE Xplore⁷, Scopus⁸, Science Direct⁹, ACM DL¹⁰, Engineering Village¹¹, Web of Science¹² e SpringerLink¹³. Os bancos de dados foram selecionados com base na experiência relatada por [Petersen *et al.* 2015]. Também foram pesquisados artigos dos seguintes locais de pesquisa sobre ECOS (não totalmente cobertos por bibliotecas digitais internacionais): Workshop sobre Desenvolvimento de Software Distribuído (WDDS) e Workshop sobre Desenvolvimento de Software Distribuído, Ecossistemas de Software e Sistemas de Sistemas (WDES), *International Workshop on Software Ecosystems* (IWSECO) e o *Workshop Ecosystem Architectures* (WEA).

Cinco pesquisadores, com experiência em mapeamento sistemático e ECOS, avaliaram cada estudo que foi recuperado da busca automatizada ou da busca manual para decidir se deveria ou não ser incluído considerando seu título, resumo e palavras-chave. As discrepâncias na seleção foram resolvidas por consenso entre os pesquisadores depois de ler todo o artigo. Qualquer artigo que não atendesse a todos os critérios de inclusão deveria ser excluído. Foram utilizados os seguintes critérios:

[Inc01] Artigos que passaram por revisão por pares;

[Inc02] Artigos escritos em Inglês;

[Inc03] Artigos que focam nos aspectos relacionados a ECOS;

[Inc04] Artigos que são relacionados a governança;

[Inc05] Artigos acessíveis na Web ou por meio de contato com os autores;

[Inc06] Artigos não-duplicados.

Foram encontrados um total de 920 artigos, aplicando a *string* de busca nas bibliotecas digitais. Destes artigos, 671 não eram duplicados. Depois de aplicar o Filtro 1 (seleção com base no título, palavras-chave e resumo), foram selecionados 245 estudos. O Filtro 2 (análise completa do estudo) nos ajudou a selecionar 91 documentos relevantes. Os filtros foram aplicados por meio de um processo de revisão por pares.

O total de 91 artigos publicados entre 2002 e 2018 (**Figura 2**) foram selecionados para extração de informações.

⁷ <https://ieeexplore.ieee.org/Xplore/home.jsp>

⁸ <https://www.scopus.com/search/form.uri?display=basic>

⁹ <https://www.sciencedirect.com/>

¹⁰ <https://dl.acm.org/>

¹¹ <https://www.engineeringvillage.com/search/quick.url>

¹² <https://login.webofknowledge.com>

¹³ <https://link.springer.com/>

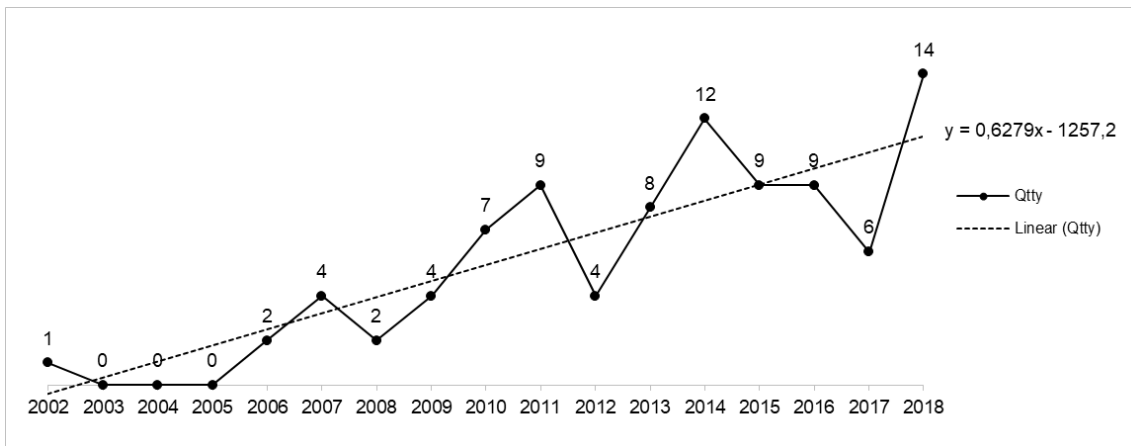


Figura 2. Quantidade de artigos publicados por ano.

A **Figura 2** fornece informações claras sobre a tendência do número de estudos publicados relacionados com a governança dos desenvolvedores em ECOS. De 2009 a 2018, o número de estudos publicados tem aumentado em geral. Desde 2014, houve pelo menos 50 estudos. A lista com os artigos selecionados está no **Apêndice A**.

2.3.2.4 Estratégia de Extração de Dados

Para cada artigo selecionado, nós extraímos informações para caracterizar a governança de desenvolvedores em ECOS, como descrito abaixo na **Tabela 2**.

Tabela 2. Itens para extração.

Nome do Item	Descrição	QP
Título	O título do trabalho	-
Autor(es)	O(s) autor(es) do trabalho	-
Ano	O ano de publicação do trabalho	-
Resumo	O resumo do trabalho	-
Palavras-chave	Conjunto de palavras-chave usadas no artigo para descrever o estudo	-
Veículo de publicação	Conferência/Revista onde o artigo foi publicado	-
Tipo de ECOS	Baseado nos conceitos de Manikas (2016) para classificar ecossistemas de software a partir da estrutura de negócios	QP1
Definição de Governança	Os aspectos que podem compor a definição de governança de desenvolvedores em ECOS	QP2
Funcionamento da Estratégia	Quais estratégias e como elas têm sido empregadas para a governança de desenvolvedores	QP3

2.3.3 (QP1) Em quais tipos de ECOS os estudos estão focados?

Para classificar o tipo de ecossistema de cada estudo foram utilizados os conceitos de Manikas do ponto de vista da estrutura de negócios [Manikas 2016], que ajuda a definir os meios de criação de valor no ecossistema. Desta forma, como já explicado na Seção 2.1 os ECOS podem ser classificados como: Proprietário, *Open*-

source e Híbrido. O ponto de vista da estrutura do negócio foi selecionado porque a criação de valor está relacionada à obtenção dos resultados desejados [Christensen *et al.* 2014], isso é essencial em qualquer negócio pois ajuda a alcançar ganhos estratégicos.

No contexto deste estudo, os resultados desejados são produzidos pela sinergia entre os objetivos principais e as necessidades dos desenvolvedores. Os artigos classificados por tipo de ECOS são indicados na **Tabela 3**.

Tabela 3. Artigos por tipo de ecossistema.

Tipo	Total (%)	Identificadores dos artigos
<i>Open-Source</i>	38 (42%)	S1, S3, S6, S9, S12, S14, S17, S21, S25, S27, S28, S29, S32, S34, S36, S39, S41, S42, S45, S48, S49, S51, S53, S54, S57, S58, S60, S62, S70, S71, S73, S75, S76, S77, S80, S82, S85, S89
Proprietário	28 (31%)	S2, S7, S8, S11, S13, S15, S16, S18, S23, S24, S26, S30, S31, S35, S40, S43, S44, S47, S50, S52, S59, S63, S64, S66, S72, S74, S79, S81, S83, S87
Híbrido	25 (27%)	S4, S5, S10, S19, S20, S22, S33, S37, S38, S46, S55, S56, S61, S65, S67, S68, S69, S78, S84, S86, S88, S90, S91

Percebe-se que o *Open-Source* é o tipo de ecossistema mais explorado nos estudos identificados (43%), seguido por Proprietário (34%) e Híbrido (23%).

2.3.4 (QP2) Qual é a definição de “governança de desenvolvedores” ?

Com o objetivo de construir uma definição de governança do desenvolvedor, foram analisadas as definições para cada tipo de ecossistema: *open-source* (Seção 3.2.4.1), proprietário (Seção 3.2.4.2) e híbrido (Seção 3.2.4.3). Ao final da análise para cada tipo, construiu-se uma definição comum de governança de desenvolvedores em ECOS.

2.3.4.1 “Governança de desenvolvedores” em ecossistemas open-source

Além de orientar a evolução do ecossistema [S39], proporcionando a sensação de segurança [S29] e apoiando a saúde do ecossistema [S57] que ajuda a garantir um maior bem-estar social para os desenvolvedores [S41]. A governança dos desenvolvedores é usada para permitir que os desenvolvedores se auto-coordenem por meio de uma tecnologia [S54]. Isto aumenta a possibilidade do ecossistema ser habitável para desenvolvedores [S49] recrutando, educando e socializando desenvolvedores para manter a vitalidade do ecossistema [S28]. Outro aspecto é o envolvimento dos desenvolvedores em projetos de organização [S27] como forma de estabelecer uma equipe produtiva e colaborativa [S62]. A governança do desenvolvedor também se concentra em unificar a comunidade de desenvolvedores

[S48] por meio do engajamento em processos que requerem mudanças na estrutura de uma organização [S36] e em como determinar quais tarefas de desenvolvimento precisam ser feitas e quando elas precisam ser completadas [S21].

A governança dos desenvolvedores, proporciona transparência sobre o impacto comercial de questões internas de qualidade [S49], acesso aberto e público aos artefatos [S6], recursos sociais que permitem relacionamentos entre os desenvolvedores [S58] fornecendo, desta forma, um ambiente para as comunidades. Este ambiente é compatível com a resolução de problemas dos membros individuais [S42] mantendo um ciclo de feedback contínuo das contribuições dos desenvolvedores [S39] e ajuda em uma economia significativa de custos em despesas de tecnologia [S1], incluindo a redução de riscos [S9].

A partir da discussão acima, gerou-se o seguinte **conceito para a governança dos desenvolvedores no contexto dos ecossistemas open-source**:

A governança do desenvolvedor consiste em instrumentos para ajudar os desenvolvedores a coordenar a tecnologia e resolver problemas na comunidade (melhor e mais rápido). Do ponto de vista da organização central, é o estabelecimento de estratégias para garantir o bem-estar social dos desenvolvedores com base no aumento da habitabilidade do ECOS, recrutando-os, educando-os e socializando-os. A governança do desenvolvedor também suporta a saúde do ecossistema, mantendo a vitalidade e reduzindo custos e riscos.

2.3.4.2 “Governança de desenvolvedores” em ecossistemas proprietários

A governança do desenvolvedor além de permitir a visualização/definição de responsabilidades [S8] e a documentação explícita das dívidas, direitos e obrigações [S44], serve para analisar a evolução [S63] e antecipar as próximas mudanças no ecossistema [S13] influenciadas pelos desenvolvedores. A governança do desenvolvedor permite que as organizações centrais obtenham informações sobre o possível papel do conhecimento no avanço de seus produtos [S11] e respondam rapidamente às mudanças de desempenho e uso em seus serviços fornecidos aos desenvolvedores [S18]. Governar desenvolvedores também está relacionado com facilitar a interação entre desenvolvedores e outros elementos do ecossistema [S3], alavancando a dinâmica de rede [S31], suportando a comunicação (propagação eficiente de informações) [S52] e monitorando o relacionamento com a organização central [S23] incluindo o objetivo da organização e dos desenvolvedores por meio do suporte ao alcance desses objetivos e do monitoramento de desempenho [S30].

A governança do desenvolvedor ajuda a identificar novas oportunidades de negócios [S8], reduzindo o custo [S40] e teorizando sobre ligações fracas em seu

modelo de negócios [S13], melhorando os processos de software como manutenção, gerenciamento e suporte ao cliente [S26] envolvendo o desenvolvedor em uma plataforma de software técnico e serviços associados em um modelo organizacional [S52]. Os mecanismos de governança são usados para controlar os desenvolvedores através de meios tecnológicos e não tecnológicos, para afetar o ambiente econômico dos desenvolvedores [S35] e engajar desenvolvedores *multi-homing* como forma de acessar mercados potenciais maiores [S64], identificando os fatores que os desenvolvedores levam em conta ao decidir sobre uma plataforma de desenvolvimento móvel [S47].

A partir da discussão acima foi possível sintetizar **o seguinte conceito para a governança de desenvolvedores no contexto de ecossistemas proprietários:**

A governança do desenvolvedor é um conjunto de estratégias que ajudam os desenvolvedores a criar e entregar produtos e serviços ao alinhar os objetivos da organização central (explicitando direitos e obrigações) com os objetivos do desenvolvedor e influenciar as próximas mudanças no ecossistema. É o estabelecimento de mecanismos para monitorar os fatores que afetam a adoção da plataforma e o ambiente econômico do desenvolvedor, e para controlar os desenvolvedores por meios tecnológicos e não-tecnológicos.

2.3.4.3 “Governança de desenvolvedores” em ecossistemas híbridos

A governança dos desenvolvedores deve que promover a saúde do ecossistema [S22], analisando as características internas da saúde e estabilidade do ecossistema [S37], promovendo a inovação dos desenvolvedores e atraindo novos desenvolvedores para reagir a mercados e clientes em rápida mudança [S56]. A governança dos desenvolvedores mantém uma plataforma que atrai desenvolvedores externos e estabelece relações colaborativas entre desenvolvedores e organização central [S55], apoiando o processo complexo de forjar e nutrir uma comunidade próspera considerando vários aspectos [S33] como: esforço de integração, custo de funcionalidade, impacto arquitetural e acoplamento [S10].

A governança dos desenvolvedores suporta a criação e o uso de produtos [S65], expandindo a demanda por usuários de produtos de software. Ocorre para ajudar os desenvolvedores a ter sucesso dentro do ecossistema e encontrar um nicho apropriado [S4]. A governança dos desenvolvedores afeta as características comerciais e econômicas da indústria de software [S4] e afeta a qualidade da plataforma do ecossistema das contribuições dos desenvolvedores, evitando os efeitos negativos do desenvolvimento distribuído [S19]. Isso afeta as características

econômicas e de negócios da indústria de software [S4] e, apoia a organização central para estudar e alinhar os interesses das partes interessadas, apoiando os acordos de ganha-ganha [S20].

A partir da discussão acima, chegou-se um **conceito para a governança de desenvolvedores no contexto de ecossistemas híbridos**:

A governança do desenvolvedor é definida por um conjunto de mecanismos para promover a estabilidade ao forjar e nutrir uma comunidade próspera que mantém relações colaborativas entre desenvolvedores e organização em um contexto “ganha-ganha”, equilíbrio entre a necessidade de autonomia e a integridade da plataforma dos desenvolvedores. Isso ocorre criando produtos e novas capacidades para reagir aos mercados em rápida mudança e às necessidades dos clientes. Afeta as características comerciais e econômicas da indústria e a qualidade da plataforma do ecossistema.

2.3.4.4 Definição de governança de desenvolvedores

A partir da combinação dos conceitos para a governança dos desenvolvedores em ECOS, apresentados nas seções anteriores, foi possível gerar o seguinte **conceito geral para a governança dos desenvolvedores em ECOS**:

A governança de desenvolvedores consiste em instrumentos para manter relacionamentos sinérgicos entre desenvolvedores internos e externos, usuários do ECOS e organização central determinados a garantir o bem-estar financeiro, social e técnico para os desenvolvedores. Ela mantém a vitalidade do ecossistema e reduz custos e riscos.

2.3.5 (QP3) Quais estratégias são utilizadas para governar desenvolvedores?

As estratégias extraídas para cada tipo de ecossistema: código aberto, proprietário e híbrido. O conjunto de estratégias também foi classificado com base nos grupos de governança do ECOS propostos por Alves *et al.* (2017): *Coordenação* - estratégias para proteger a estabilidade e a aliança de atores, ações, interações e configuração do ecossistema; *Criação de Valor* - envolve estratégias para sustentar os negócios para todo o ecossistema; e *Abertura e controle da organização* - instrumentos para capturar a harmonia notória entre modelos organizacionais abertos versus fechados.

Extraímos 42 estratégias para governar desenvolvedores em ecossistemas de software classificados da seguinte forma:

- 43% (18) Open-source: Coordenação: 8 (44,4%); Criação de Valor: 6 (33,3%); e Abertura e controle da Organização: 4 (22,2%);
- 38% (16) Proprietário: Abertura e controle da organização: 8 (50%); Coordenação: 4 (25%); e Criação de Valor: 4 (25,5%);
- 19% (8) Híbrido: Criação de Valor: 5 (62,5%); Coordenação: 2 (25%); e abertura e controle da Organização: 1 (12,5%).

As estratégias são descritas a seguir em forma de um fluxo de execução.

2.3.5.1 Ecossistemas Open-Source – Coordenação

1. *Manter a capacidade absorptiva* [S48]: tipo de estratégia utilizada quando as organizações estão dispostas a garantir que a plataforma e os produtos do ecossistema continuem respondendo às necessidades da comunidade de desenvolvedores e usuários. Isso se destina a manter a atratividade dos desenvolvedores, incentivando-os. Nesse cenário, é importante que as organizações combinem as contribuições feitas por equipes internas de desenvolvedores com contribuições de desenvolvedores externos.
2. *Analisar os desenvolvedores utilizando modelos*:
 - *Modelo de negócios* [S3]: forma de ajudar a garantir a dinâmica do ganho e o uso de ferramentas apropriadas. Por meio de escolher a licença certa de código aberto, como uma plataforma de software, modelo de impacto no faturamento, necessidade de mercado identificada e a ênfase em serviços com foco no servidor;
 - *Modelo de desenvolvimento* [S3]: baseia-se em um processo de desenvolvimento bem definido que depende unicamente da coordenação de muitos desenvolvedores. Esse modelo compreende: defender uma comunidade abrangente de desenvolvedores e empresas parceiras; comunicar-se com os desenvolvedores com frequência e vigor; lançar novos softwares com frequência; aplicar forte coordenação; ser responsivo e garantir visibilidade no processo;
 - *Modelo de coordenação* [S9]: estratégia para estabelecer e comunicar quais componentes do processo de coordenação a organização central deseja conhecer. Pode ser analisado por meio de um meta-modelo que visa estabelecer um vocabulário comum que compreende os seguintes aspectos: atributos, restrições, objetos e as relações entre eles.

- *Modelo de gerenciamento de contribuição* [S36]: um tipo de estratégia de governança de desenvolvedores a partir da perspectiva de contribuições. Este modelo é composto pelas seguintes fases:
 - i. *Concepção*: a organização central precisa dar suporte aos desenvolvedores, por meio de recursos técnicos e interação com a comunidade, para que as contribuições se tornem concretas;
 - ii. *Submissão*: após a concepção da contribuição, o desenvolvedor pode enviá-la a um repositório de distribuição (por exemplo, lojas de aplicativos móveis). Como muitas dessas contribuições vêm de desenvolvedores externos, controlar a qualidade da contribuição é uma preocupação;
 - iii. *Revisão*: ao passar por esta fase, a contribuição submetida tem três status possíveis: (1) Contribuição aceita; (2) Contribuição precisa ser modificada e submetida novamente; (3) Rejeitada;
 - iv. *Integração*: embora a contribuição seja aceita, pode entrar em conflito com algum outro componente da plataforma do ECOS. Se a contribuição não se integrar, o desenvolvedor será aconselhado a executar a remoção de conflitos e enviar uma versão atualizada da contribuição. Isso também pode ocorrer quando há uma mudança na plataforma do ECOS.
- *Modelo de comunidade* [S54]: analisa a comunidade de desenvolvedores como uma cebola. No centro estão os desenvolvedores mais influentes e ativos e, a cada outra camada, consiste em desenvolvedores menos influentes. Os desenvolvedores no centro da “cebola” são aqueles que coordenam o projeto de alguma forma e contribuem mais do que outros. Os desenvolvedores mais distantes do centro são aqueles que aproveitam ao máximo os recursos gerados pelo projeto. Esses desenvolvedores são essenciais porque ajudam na melhoria e disseminação de recursos em torno de uma contribuição para a plataforma do ecossistema;
- *Modelo de fluxo de avanço do desenvolvedor* [S28]: envolve quatro possíveis caminhos:
 - i. *Acelerado*: está relacionado a um desenvolvedor que participa do ecossistema, inicialmente, por meio social, (por exemplo, fóruns, sites de perguntas e respostas, lista de discussão) e, em seguida, engaja-se dentro do ecossistema;

- ii. *Social para técnico*: envolve desenvolvedores que participam do ecossistema em um meio social e, em seguida, progridem para atividades técnicas. Um critério de diferenciação deste caminho é que aqui o desenvolvedor só pode estar envolvido em um meio;
 - iii. *Técnico*: desenvolvedor que foi integrado por meio técnico (por exemplo, repositórios de código) e está envolvido apenas nesse caminho;
 - iv. *Técnico para social*: o desenvolvedor está começando no meio técnico e foi identificado que somente após o primeiro lançamento de uma contribuição é que os desenvolvedores mudam para o meio social. Isso pode estar relacionado à disseminação da contribuição como uma forma de encontrar o reconhecimento da chave e da comunidade.
3. *Monitorar a comunidade de desenvolvedores* [S33]: Essa estratégia é baseada na análise das seguintes métricas: Entradas (Estrutura do Grupo): Tamanho; Heterogeneidade dos Membros; Dispersão Geográfica; Contribuição Externa; Processo (Coordenação do Grupo) e Padrões de Interação; Papéis informais; Resultado da estrutura de rede (desempenho do grupo); Tempo de resolução de defeitos;
4. *Analisar o modelo de parceria estabelecido entre os desenvolvedores* [S33][S43]: os desenvolvedores devem pagar taxas de parceria e garantir que suas contribuições considerem os aspectos relacionados à certificação de qualidade. Em resposta, a organização central fornece aos desenvolvedores benefícios que incluem: ferramentas de desenvolvimento, conferências para desenvolvedores e suporte técnico. Esta estratégia baseia-se na utilização de indicadores de rede como:
- *Centralização*: é um indicador usado quando uma comunidade dentro do ECOS está centrada em torno de um ou mais desenvolvedores. Isso pode ajudar a analisar quanto conhecimento gerado no ECOS é distribuído para outros desenvolvedores e para a organização central;
 - *Coeficiente de agrupamento*: é um indicador que reflete os grupos de conexões entre desenvolvedores no ecossistema. Este indicador pode ajudar a entender as comunidades de desenvolvedores dentro do ecossistema e como os recursos do ecossistema são usados;

- *Grau de Centralidade*: é um indicador para analisar e avaliar a posição dos desenvolvedores dentro do ECOS. Essa posição é importante porque os desenvolvedores especialistas que estão em posições que ajudam a estabelecer relacionamentos com outros desenvolvedores aprimorando, assim, o conhecimento e criação/evolução de contribuições.

2.3.5.2 Ecossistemas Open-Source – Abertura e Controle da Organização

1. *Analisar e projetar o fluxo de requisitos* [S12]: baseia-se na teoria de negociação e de rede. Este fluxo destina-se a ajudar na tomada de decisões baseadas em requisitos que surgem da comunidade de desenvolvedores e usuários do ecossistema. Isso pode ajudar na solução de problemas de comunicação entre as comunidades de desenvolvedores e dos desenvolvedores com organização central. Desta forma, a organização central pode atribuir poder à comunidade de desenvolvedores;
2. *Estabelecer conformidade com o gerenciamento de riscos corporativos* [S30]: é importante descrever processos que assegurem a aderência de ecossistema a obrigações regulatórias, legais, contratuais e outras. Esses processos são compostos por: Gerenciamento de riscos; Gerenciamento de políticas; Gestão de auditoria; Relatórios/Dashboards/Analytics; Gestão de conformidade; Controla testes e gerenciamento; Controles financeiros; Pesquisas; Gestão de fluxo de trabalho; Governança corporativa; e auditorias e conformidade de TI;
3. *Estabelecer e comunicar estilos arquiteturais* [S38]: esses estilos podem ajudar a capacitar a evolução de todos os membros do ecossistema;
4. *Planejar e executar conferências de desenvolvedores* [S51]: isto suporta decisões sobre os produtos das plataformas do ECOS. Os eventos do desenvolvedor (por exemplo, conferências, *hackathons*, *meetups*) têm a finalidade de reunir os líderes da comunidade de desenvolvedores para permitir o compartilhamento de insights sobre os avanços necessários em produtos e plataformas;
5. *Delimitar zonas controláveis* [S31]: é composto por três zonas de autoridade da organização central em relação aos desenvolvedores: (1) Zona de Controle, que auxilia no alto nível de observação dos desenvolvedores, além do controle direto das relações e atividades; (2) Zona de Influência, nesta zona, a organização central só pode influenciar as atividades dos desenvolvedores através de informações e incentivos direcionados; e (3) A Zona de Ruído que

envolve todos os desenvolvedores que são indiferentes às estratégias de organização central;

6. *Delimitar zonas de auto-organização* [S31]: visa apoiar a criatividade do desenvolvedor, reduzindo o controle da comunidade. Desta forma, o desenvolvedor externo se auto-organiza ao longo do tempo adaptando sua estratégia com o objetivo de obter reconhecimento e criar valor. Esta estratégia pode ser realizada pelo uso e incentivo de mecanismos de controle dentro de um projeto de contribuições apropriadas para o ecossistema;
7. *Identificar e mitigar barreiras* [S1]: ajuda a identificar os obstáculos para a participação e o avanço dos desenvolvedores em um ECOS. É importante construir e revisar um plano para mitigar as barreiras através de “remédios” com o objetivo de prevenir e reduzir a influência deles no ecossistema.

2.3.5.3 Ecossistemas Open-Source – Criação de Valor

1. *Analisar e agir a partir do conhecimento gerado por desenvolvedores* [S18]: isto cobre os tipos de conhecimento sobre a operação de software (SOK, do inglês *Software Operation Knowledge*) consiste em quatro tipos de conhecimento: (1) Feedback do usuário final; (2) Qualidade; (3) Plataforma e Uso de Produtos; (4) Desempenho. Esta é uma maneira de obter insights sobre as opiniões dos usuários durante o uso do software. Além de ser um método sistemático para analisar e atuar sobre o conhecimento obtido dentro do ECOS;
2. *Integrar desenvolvedores com consumidores e fornecedores* [S50]: estratégia executada por meio das seguintes etapas:
 - (1) A organização central negocia contratos para aceitar contribuições para a plataforma do ecossistema com cada desenvolvedor separadamente;
 - (2) O desenvolvedor negocia diretamente com outro desenvolvedor, gerando colaborações. Desta forma, os componentes de código aberto podem facilitar a colaboração [S23];
 - (3) O desenvolvedor assume toda a responsabilidade pela submissão e integração da contribuição com outros desenvolvedores. A aceitação da contribuição para a plataforma é feita através de critérios de certificação de qualidade.
3. *Analisar a dinâmica do ecossistema* [S61]: visa controlar os desenvolvedores como produtores de contribuições para o ECOS. Em um nível inferior, estão os desenvolvedores que agem como produtores primários, seus recursos podem ser usados por todos os tipos de desenvolvedores. Em um nível mais alto, estão os desenvolvedores agem como predadores, porque quanto maior o

nível, mais eles se concentram em usar os recursos do ECOS para suas contribuições sem gerar recursos para a comunidade. Os desenvolvedores que estão no nível médio criam contribuições para manter a arquitetura do ECOS gerando dependência entre os componentes usados nas contribuições. É importante deixar claro que os desenvolvedores têm recursos técnicos, econômicos e sociais limitados que devem ser apoiados para que a organização possa atender às demandas dos usuários.

2.3.5.4 Ecossistemas Proprietários – Coordenação de Desenvolvedores

1. *Explorar a faixa de responsabilidades dos desenvolvedores [S29]:* engloba o estabelecimento de suporte (organização e infraestrutura) ao desenvolvedor, medição da redução de riscos e garantia de abertura e transparência:
 - Estabelecer apoio: responsabilidade e papel devem ser claros;
 - Medir o risco: as contribuições dos desenvolvedores devem ser revisadas em alguns pontos de verificação como forma de garantir a integração com a arquitetura da plataforma do ECOS;
 - Garantir transparência e abertura: quando houver uma proposta de contribuição do desenvolvedor (objetivo, escopo, roteiro etc.) estabelecida. Depois disso, é necessário que a contribuição seja aprovada em critérios de qualidade por organização central. Quando a contribuição está disponível para a comunidade, os usuários e desenvolvedores podem fornecer análises que devem ser públicas e visíveis para todos;
 - Suporte de infraestrutura de TI: é necessário preparar infraestruturas para o gerenciamento de projetos, como repositórios de código fonte, fóruns, exemplos de código, sites de gerenciamento de problemas.
2. *Minimizar a perda social [S7]:*
 - Envolve as seguintes partes interessadas: desenvolvedor de software (fornecedor), implantadores de software (organizações de software), organização central;
 - Processo de disseminação do conhecimento da vulnerabilidade: analisa como a organização central deve divulgar as vulnerabilidades existentes no ecossistema sem causar perda social.
3. *Identificar o modelo de negócios e a rede de gerenciamento de parceiros [S52]:* esta estratégia é baseada na identificação dos objetivos dos desenvolvedores, “facilitadores” e suas respectivas “ferramentas” que permitem que os objetivos sejam alcançados, os “efeitos” descrevendo as percepções de ambos desenvolvedores e organização central sobre as contribuições, e

“influenciadores” que são motivadores para o engajamento ou o desligamento do desenvolvedor.

- O “compartilhamento de recursos” é o facilitador mais importante em um ecossistema. Em seguida, o “fluxo do ciclo de vida” que, através das ferramentas de desenvolvimento, colaboração, publicação e certificação de qualidade, permite aos desenvolvedores colaborar e gerar contribuições dentro do esperado por organização central. A organização central precisa ser transparente sobre os objetivos e processos organizacionais relacionados às contribuições para o ecossistema. Esses processos devem ser compatíveis com os processos de organização central internos;
 - Outro ponto é que o desenvolvedor precisa ter acesso ao roteiro de produto da organização para poder planejar um portfólio que continue a servir organização central e usuários. O compartilhamento dessas informações é fundamental para interagir com o desenvolvedor. Compreender a governança a partir do contexto de produto de software e das cadeias de suprimentos [S8], identificando todos os participantes, as conexões entre eles e o fluxo de informações entre a rede de participantes.
4. *Monitorar os indicadores do desenvolvedor como Produtividade, tamanho, dispersão e experiência [S19].*

2.3.5.5 Ecossistemas Proprietários – Abertura e Controle

1. *Identificar estruturas organizacionais, empresariais e de software [S2];*
2. *Avaliar o status do ecossistema e o sucesso das decisões [S16]:*
 - SECO-SAM: consiste em um modelo e um conjunto de ferramentas de pesquisa. O SECO-SAM define as principais características de um SECO que pode ser mensurável.
3. *Aumentar a capacidade de resposta às mudanças no desempenho e na utilização de serviços [S11]:*
 - Utilização de conhecimento sobre serviço (SKU, do inglês *Software Knowledge Utilization*);
 - Quantificar o uso e o feedback do serviço.
4. *Estabelecer uma plataforma para publicação e propagação de conhecimento do ecossistema dentro do ecossistema [S26]:*
 - Conhecimento de Operação de Software (SKO, do inglês *Software Knowledge Operation*);
 - Relatórios de erros, indicadores de desempenho;

- Desenvolver diretrizes que ajudem a aumentar a visibilidade do ecossistema e capacitar os desenvolvedores;
 - Definir/redefinir certificação de qualidade para manter a robustez do ecossistema.
5. *Fornecer um ambiente tecnológico para que organizações e desenvolvedores armazenem e compartilhem conteúdo de treinamento para o público-alvo de desenvolvimento de software* [S14]: ajuda na manutenção e expansão do conhecimento dentro do ecossistema. Tanto desenvolvedores quanto organização podem criar e revisar conteúdo sobre produtos e plataforma.
- Acompanhe a eficácia de cada sessão de treinamento definindo uma Estrutura de Treinamento Interativa (ITF).
6. *Permitir a autonomia e a heteronomia dos desenvolvedores* [S25]:
- Analise a estratégia de governança com base nos seguintes tipos:
 - i. *Organização centralizada*: a organização central permite que um desenvolvedor planeje e organize ações em uma comunidade que gera contribuições para uma plataforma do ECOS;
 - ii. *Colaboração colegial*: é a estratégia mais livre para colaboração com desenvolvedores. Não há responsabilidades definidas para o desenvolvedor participar do ecossistema. No entanto, existem diretrizes para participação e contribuição;
 - iii. *Guerreiro solitário*: Esse tipo de estratégia prioriza desenvolvedores que agem mais livremente dentro do ecossistema. É o primeiro tipo de desenvolvedor que tem alta autonomia no ecossistema. Para isso, é importante que o ecossistema forneça recursos e reconhecimento.
 - Aumentar a flexibilidade dos fornecedores de software [S11]:
 - i. Utilização de conhecimento de serviço (SKU).
7. *Garantir abertura e transparência* [S29].

2.3.5.6 Ecossistemas Proprietários – Criação de Valor

1. *Aumentar a organização de desenvolvedores com base no feedback* [S39]: envolve uma estratégia em que todas as informações são importantes para a manutenção de desenvolvedores, recursos e ferramentas do ecossistema. Essa estratégia é um mecanismo importante que permite que desenvolvedores e organização central se adaptem e permaneçam competitivos no mercado;

2. *Estabelecer um modelo de gestão de contribuição* [S49]: propõe algumas perspectivas de análise baseadas em negócios, arquitetura, processo e organização da seguinte forma:
 - Negócio: os desenvolvedores são possíveis fontes de renda para o ECOS. Mas eles têm objetivos, projetos e expectativas do ponto de vista de negócios que devem ser levados em conta pela organização central. Os modelos de negócios devem se basear principalmente na redução de custos para dimensionar atividades que promovem o relacionamento com os desenvolvedores;
 - Arquitetura: desacoplamento das contribuições dos desenvolvedores que farão parte da plataforma. Isso permitirá que, se alguma contribuição gerar problemas, não tenha impacto na estrutura do ecossistema;
 - Processo de contribuição: a organização central deve direcionar os desenvolvedores para sincronizar projetos, o ciclo de vida da contribuição e os processos de engenharia de software, como uma forma de alinhar as diversas comunidades dentro do ECOS.
3. *Forneça um repositório central para garantir a consistência das contribuições* [S53]: ajuda a armazenar contribuições consistentes (por exemplo, pacotes de software, código-fonte, modelos) que são criados e disponibilizados por vários agentes do ECOS, como desenvolvedores, empresas e funcionários da organização. Os dados presentes nos repositórios podem ser extraídos e analisados para extrair informações que auxiliem na governança dos elementos do ecossistema;
4. *Implementar um mecanismo de cooperação baseado em inovação aberta* [S34]: nesta estratégia, a chave deve permitir o compartilhamento de ideias e conhecimentos relacionados à plataforma do ECOS entre desenvolvedores, recursos técnicos, dispositivos e outros produtos. Isto ajuda a melhorar o intercâmbio de informações entre desenvolvedores externos/internos, usuários e organização central. Maior transparência do conhecimento contribuirá com uma maior possibilidade de surgirem oportunidades de negócios entre a comunidade de desenvolvedores e a organização central. É composto por:
 - Aceitação de contribuição: consiga aceitação da solução devido a agir em base de diretrizes da plataforma;
 - Simulação de Ecossistemas: usar métodos de simulação para obter insights sobre modelos que podem influenciar a produtividade e a robustez do ecossistema a partir da perspectiva de engajamento e atração de desenvolvedores externos;

- Integração de modelos: insira os modelos identificados por simulação como parte da estratégia de governança do desenvolvedor. Pode ser realizado da seguinte forma: (a) disseminação do modelo com os atores envolvidos no ecossistema; (b) automatizado, onde um sistema de simulação é importante para prever situações dentro do ecossistema a partir da aplicação de estratégias de governança; e, (c) interativamente, onde os atores do ECOS interagem com o sistema de simulação;
- Processo de certificação de qualidade: garantir que um desenvolvedor terceirizado gera contribuições de acordo com um conjunto de critérios estabelecidos.

5. *Apoiar a resolução dos desafios de empreendedorismo [S35]:*

- Implemente uma infraestrutura de serviços com foco nos negócios: quando um desenvolvedor inicia o negócio, sua missão é fazer algo que amam, manter o negócio sustentável e aumentar as vendas. Para isso, eles precisam ter suporte na resolução de desafios empresariais, implementação de serviços de infraestrutura mais adequados para os negócios e adaptação à mudança de tecnologia;
- Controlar o progresso da plataforma e a mudança de tecnologias: os desenvolvedores são responsáveis por seu relacionamento com as partes interessadas, ou seja, clientes, provedores de infraestrutura e a plataforma. Enquanto isso, os desenvolvedores também tentam equilibrar o lado pessoal e profissional do negócio. Os desenvolvedores dependem muito de provedores de serviços de mercado, como agregadores de software, e-commerce e provedores de pagamento.

2.3.5.7 Ecossistemas Híbridos – Coordenação de desenvolvedores

1. *Atuar para planejar, executar e analisar uma coordenação descentralizada de desenvolvedores [S21]:* a comunidade de desenvolvedores externos é grande e dispersa. A coordenação desses desenvolvedores requer processos de governança que devem ser ajustáveis conceitualmente e em configurações;
2. *Estabelecer uma estrutura organizacional que permita a colaboração da comunidade de desenvolvedores considerando [S21]:*
 - Nível micro: neste nível, a organização central deve fornecer estruturas de ecossistema para favorecer a comunicação entre os desenvolvedores, especialmente sobre como os recursos para a expansão da plataforma são constituídos e podem ser usados;

- Nível meso: a organização central deve permitir que os desenvolvedores possam trabalhar individualmente em uma infraestrutura que permita a colaboração. É importante que a infraestrutura fornecida direcione os desenvolvedores para atingir as metas organizacionais. Isso pode ser garantido por políticas que envolvem o projeto, desenvolvimento, certificação de qualidade e marketing de contribuições geradas pelos desenvolvedores. Essas políticas devem atenuar o risco de conflitos na comunidade de desenvolvedores;
- Nível macro: a organização central deve garantir que quaisquer conflitos que possam existir entre os desenvolvedores não afetem o ECOS. Isso pode ser feito estabelecendo padrões de conduta no ecossistema. Os conflitos envolvem, por exemplo, expectativas e valores dos desenvolvedores, roadmaps e compartilhamento de conhecimento, bem como barreiras técnicas que podem interferir no avanço do desenvolvedor.

2.3.5.8 Ecossistemas Híbridos – Abertura e Controle

1. *Use um processo baseado em negociação ganha-ganha envolvendo estrutura social, estrutura de informação e dinâmica [S20]:*
 - Estrutura Social: a organização central deve permitir a colaboração entre departamentos organizacionais como engenharia, gerenciamento e marketing para monitorar o desempenho dos produtos para desenvolvedores externos. Essa estrutura social interna aliada à estrutura dos desenvolvedores proporcionará à organização central um ambiente colaborativo e um feedback contínuo sobre o ECOS. Além disso, tanto os departamentos internos quanto os desenvolvedores terão uma visão do mercado (incluindo concorrentes), oportunidades de negócios, usuários e contribuições;
 - Estrutura de Informação: deve envolver interesses identificados tanto pelos desenvolvedores quanto pela organização central. A organização central procura atingir os objetivos da organização em sinergia com os objetivos do desenvolvedor. Dessa maneira, os usuários de produtos do ecossistema lucrarão com o valor gerado pelas contribuições que usam essas informações. Os acordos envolvem funcionalidades, qualidade de software e requisitos que, se bem negociados, permitirão a colaboração dentro do ecossistema;
 - A dinâmica: a organização central precisa identificar o que leva os desenvolvedores a entrar, sair e se movimentar dentro do ecossistema.

Isso ajudará a melhorar os relacionamentos com os perfis do desenvolvedor e dimensionar as atividades de integração do desenvolvedor. Existem dois tipos possíveis de integração:

- i. Ativo: que envolve algum funcionário interno da organização que irá trabalhar na integração dos desenvolvedores;
- ii. Passivo: a identificação de interesses, valores, expectativas dos desenvolvedores dentro do ecossistema que tornarão um desenvolvedor atraente para a organização.

2.3.5.9 Ecossistemas Híbridos – Criação de Valor

1. *Ajudar os desenvolvedores a reutilizar recursos técnicos de outras pessoas [S10] de maneira pragmática e oportunista:*
 - Oportunamente: é uma maneira de acelerar o desenvolvimento de produtos ou serviços. Serve como extensão da plataforma com componentes resultantes da contribuição de um desenvolvedor que não desenvolveu a contribuição para disponibilizá-la para reutilização;
 - Pragmaticamente: está relacionado ao aprimoramento da plataforma e dos recursos técnicos através da disseminação de componentes de um desenvolvedor que podem não ter sido desenvolvidos com a reutilização em mente.
2. *Identificar as cadeias de valor:* as atividades que um desenvolvedor realiza em um cenário específico para planejar, construir, liberar contribuições valiosas [S37]. É uma atividade importante para uma empresa fornecedora em um ECOS com o objetivo de apoiar desenvolvedores de terceiros a contribuir usando seus produtos (por exemplo, ferramentas, frameworks, SDKs). A organização central deve garantir que as contribuições funcionem bem quando combinadas para gerar novas cadeias de valor.
3. *Investir em produtos do ecossistema para maximizar seu lucro e os desenvolvedores investem para aumentar sua reputação e benefícios [S4]:* é importante aumentar a qualidade do produto e expandir a demanda de contribuições para a plataforma. Nesse cenário, a organização central deve investir em sua produção e na plataforma para aumentar seus ganhos e ajudar os desenvolvedores a investir esforços para aumentar sua reputação;
4. *Desenvolver um sistema baseado na meritocracia de desenvolvedores [S45]:* neste sistema, os desenvolvedores são escolhidos com base em suas contribuições, reputação e realização de metas. Em seguida, eles avançam no

ecossistema e podem ser usados como importantes atores para apoiar as comunidades de desenvolvedores:

- Suporte ao desenvolvedor no aumento de sua reputação;
- Investir em contribuições para maximizar os lucros da organização.

5. Analisar *repositórios de software* [S24]:

- Componente ou dados de software reutilizáveis;
- Repositório de dados do ecossistema;
- Site de rede social.

6. Usar dados de *repositórios de software existentes e uma ferramenta de rede social para apoiar o entendimento das dimensões social, de negócio e técnica das interações entre desenvolvedores* [S46]: ele pode ajudar na modelagem e no monitoramento de ECOS.

2.3.6 Ameaças à Validade

Nesta seção, identificamos possíveis ameaças à validade de nossos resultados de revisão, usando a taxonomia de Wohlin *et al.* (2013).

Validade de *constructo*: as principais ameaças nesta categoria estão relacionadas às questões de pesquisa, bibliotecas de dados usadas e *string* de pesquisa. As questões de pesquisa podem não abordar todos os aspectos da governança dos desenvolvedores em ECOS. Para minimizar essa ameaça, foram elaboradas perguntas de pesquisa cobrindo as principais respostas para explorar o que a literatura científica diz sobre a governança de desenvolvedores em ECOS. Foi selecionado um conjunto específico de bibliotecas digitais porque elas são fontes bem conhecidas, conforme indicado por [Petersen *et al.* 2015] e são usadas em revisões sistemáticas relacionadas ou estudos de mapeamento.

Com relação à *string* de busca usada, ela não possui muitos elementos, mas para minimizar essa ameaça, tomou-se o cuidado de criar uma *string* capaz de encontrar resultados coerentes. Para o termo de pesquisa ecossistema de software, muitos termos foram usados e a maioria deles foi extraída dos trabalhos relacionados. Para minimizar essa ameaça se utilizou um grupo de controle dos principais trabalhos conhecidos no campo de governança dos desenvolvedores. Para melhorar a *string* de busca esta foi executada 10 execuções no Scopus e extraímos sinônimos de palavras-chave previamente identificadas.

Validade interna: pode-se considerar as decisões subjetivas que podem ter ocorrido durante a seleção de estudos primários e extração de dados. Alguns estudos relevantes podem não ser selecionados como estudos primários. Para minimizar essa ameaça, seguiu-se o plano, guiado pelos critérios de inclusão e exclusão. O protocolo

de mapeamento sistemático foi revisado por pesquisadores com larga experiência em Engenharia de Software Experimental e planejamento/execução de revisões sistemáticas.

Dentre as ameaças à **validade de conclusão**, as ameaças identificadas são o esquema de classificação e a maneira como foram agrupados os documentos e foram estabelecidas as relações entre eles. No entanto, outras revisões podem ter outros esquemas de classificação e formas de agrupar e analisar os documentos. Outra ameaça está relacionada à granularidade das informações apresentadas nos estudos primários revisados. Se alguma informação não foi descrita nestes estudos, pode afetar as conclusões.

A **validade da confiabilidade** está relacionada a problemas que afetam a capacidade de desenhar que as operações de um estudo podem ser repetidas com os mesmos resultados. Acreditamos que nosso estudo pode ser facilmente replicado seguindo as etapas descritas e usando a *string* de busca.

2.3.7 Conclusões do mapeamento

Percebe-se a partir da análise das questões de pesquisa investigadas e das oportunidades de pesquisa que é necessário investir na formalização e avaliação prática de estratégias para governar os desenvolvedores como forma de apoiar o recrutamento, engajamento e sobrevivência dos desenvolvedores dentro do ECOS. A diretriz obtida pelos pesquisadores é que essas estratégias devem se concentrar na coordenação, criação de valor e abertura e controle da organização. Há uma necessidade de estudos sobre como os relacionamentos surgem e ocorrem entre os desenvolvedores de um ECOS. Outro ponto é a definição, análise e avaliação dos mecanismos (por exemplo, práticas, recomendações, abordagens, ferramentas) para a governança de desenvolvedores que favoreçam a criação de valor pelos atores envolvidos no ecossistema.

As estratégias apesar de certa evidência experimental precisam ter a relevância para a governança de desenvolvedores em ECOS analisadas por profissionais que atuam no cenário real. E, por isso, o Capítulo 3 apresenta uma pesquisa de opinião que investiga esse aspecto de relevância das estratégias.

2.4 Trabalhos Relacionados

Após a análise de revisões e mapeamentos sistemáticos sobre ECOS [Barbosa *et al.* 2013][Manikas 2016] [Franco-Bedoya *et al.* 2017], MSECO [Fontão *et al.* 2015][Steglich *et al.* 2019] e governança de ECOS [Alves *et al.* 2017], não foram encontradas abordagens de governança de desenvolvedores em MSECO. A

comparação resumida dos trabalhos relacionadas é apresentada na **Tabela 4** e a descrição de cada trabalho é discutida em seguida.

Tabela 4. Comparação entre trabalhos relacionados.

Trabalho	Contribuição	Aspecto de Governança	Desenvolvedores? / MSECO?
Jensen e Scachi (2010)	Práticas e processos para governança de projetos <i>open-source</i>	Controle de desenvolvedores e coordenação de atividades	Não / Não
Jergensen <i>et al.</i> (2011)	Modelo de migração de desenvolvedores <i>open-source</i>	Fluxo de movimentação dos desenvolvedores	Sim / Não
Qiu <i>et al.</i> (2011)	Modelo de governança entre desenvolvedores	Recursos e ambiente dos desenvolvedores	Sim / Não
Kilamo <i>et al.</i> (2012)	Diretrizes e melhores práticas para criar um ECOS <i>open-source</i> em torno de projeto proprietário.	Expansão de plataforma	Sim / Não
Albert <i>et al.</i> (2013)	Abordagem para localização no mercado e mapeamento de relações	Recursos técnicos e organização consumidora	Não / Não
Wareham <i>et al.</i> (2014)	Entendimento da governança de um ECOS	Estabilidade e evolução de um ECOS	Não / Não
Sadi <i>et al.</i> (2015)	Abordagem para identificação de desenvolvedores	Projeto de Colaboração	Sim / Sim
Foerderer <i>et al.</i> (2018)	Análise de limites do conhecimento envolvidos em governança	Conhecimento em torno de recursos do ECOS	Não / Não
Kude <i>et al.</i> (2018)	Estratégias para recrutar, selecionar e treinar gerentes de parcerias	Programas de Parceria	Não / Não
Vorraber <i>et al.</i> (2018)	Framework para alinhamento de objetivos entre desenvolvedores e gerentes de negócios	Mecanismos de criação de valor em torno de um projeto	Sim / Não

Jensen e Scacchi (2010) examinam como as práticas e processos permitem governar os projetos *open-source* quando reunidos e configurados como redes de interação socio-técnica. Os elementos avaliados podem ser analisados em nível micro ou macro. O nível micro consiste em recursos que os desenvolvedores utilizam para controlar o trabalho do projeto e as contribuições, como exemplo temos: colaboração, políticas e diretrizes, estratégia arquitetural, liderança e controle, transparência de decisões e resolução de conflitos. Já os de nível macro consistem na relação entre o projeto com outros projetos do ecossistema, como: a coordenação, liderança, controle e resolução de conflitos. Os autores realizaram dois estudos de caso que indicam que

há uma necessidade de um modelo de governança de desenvolvedores que envolva a comunidade nos níveis micro e macro.

Jergensen *et al.* (2011) estudam a migração de desenvolvedores dentro de um ecossistema open-source por meio de um modelo de “cebola” que descreve o fluxo de avanço dos desenvolvedores dentro do ecossistema. Os autores identificaram que o uso de repositórios colaborativos de projetos de software como o Github e BitBucket permitiram um avanço na socialização entre desenvolvedores. No entanto, isto fez com que as contribuições para um projeto passassem por processos de verificação e validação de qualidade antes de compor o projeto. Os autores indicam que há uma necessidade por identificar e analisar a variedade de efeitos técnicos e sociais que surgem dentro da comunidade de desenvolvedores.

Qiu *et al.* (2011) realizaram um estudo para a identificação do modelo de governança dentre desenvolvedores da plataforma Mac a partir de uma perspectiva lógica: lógica do ecossistema e lógica da plataforma. Para isto, concentra-se na identificação de recursos e no ambiente dos desenvolvedores. Ainda investigam o lado dos desenvolvedores e o papel da organização, neste caso a Apple, que geram impactos no dinamismo e na dimensão econômica dos desenvolvedores. Como pontos principais, identificados para compor a lógica do ecossistema e da plataforma, temos como exemplo: (Ecossistema) qualidade das apps, aumentar vendas, construir um negócio sustentável, crescimento orgânico e a reputação/reconhecimento do desenvolvedor; (Plataforma) crescimento guiado pela organização, construir competitividade entre as apps, qualidade avaliada por meio de comentários de usuários. Os autores indicam a necessidade por examinar ambientes, políticas de governança e recursos de outros ECOS e o impacto disso nos desenvolvedores.

Kilamo *et al.* (2012) investigam como construir um ecossistema *open-source* ao redor de um software proprietário. Os autores discutem que as empresas têm buscado complementar socialmente a plataforma com comunidades de desenvolvedores. Os autores ainda comentam que este é um campo de pesquisa relativamente novo, que há poucas diretrizes em como criar e manter um ECOS sustentável para uma plataforma proprietária. Por meio de um estudo de caso, analisam diretrizes e melhores práticas para o cenário apresentado. Os autores focaram mais no nascimento do ecossistema e como garantir o crescimento. Por isso, apontam uma necessidade de estudo dos outros estágios envolvidos, como garantir o crescimento deles e a identificação de práticas de apoio a decisão e métodos de comunicação.

Albert *et al.* (2013) propõem uma abordagem para governança de ECOS que permita à organização se localizar no mercado e mapear suas relações com fornecedores, distribuidores, produtos e tecnologia por meio de uma ferramenta

chamada Brechó-SECOGov. O foco está nos recursos técnicos e na percepção da organização consumidora pelos participantes, mas não na análise de um ecossistema híbrido como um MSECO. Além disso, o trabalho não foca em desenvolvedores.

Wareham *et al.* (2014) estudam a governança de uma plataforma como forma de orquestrar um portfólio de contribuições produzidas por atores independentes e heterogêneos que forma um ecossistema. Dentro disso os autores investigam mecanismos de governança apropriados para promover um balanceamento entre a estabilidade e evolução do ecossistema. Os autores contribuem com o entendimento da dinâmica de um ECOS, com o efetivo projeto da governança de uma plataforma e explicitam tensões resultantes das relações entre os envolvidos no ecossistema. Os autores sugerem o estudo das diferentes fases de evolução da governança.

Sadi *et al.* (2015) propuseram uma abordagem genérica baseada nos ecossistemas Android e iOS para identificar tipos de desenvolvedores e derivar soluções alternativas para projetar uma colaboração apropriada. Este estudo se concentra nos objetivos e critérios de decisão dos desenvolvedores, mas não fornece orientação específica sobre como essas atividades podem ser realizadas. Para demonstrar a viabilidade das recomendações propostas, experimentação em casos reais é necessária.

Foerderer *et al.* (2018) analisam os limites do conhecimento envolvidos na governança de um ECOS proprietário. Eles analisam vários recursos incluindo portais de apoio ao desenvolvedor, documentação e workshops. A análise indica que esses recursos ajudam na definição do escopo para o qual o conhecimento será direcionado, permitindo a escalabilidade do conhecimento no ecossistema. Não há uma análise específica da governança de desenvolvedores do ponto de vista dos recursos utilizados. Além disso, o foco é em ECOS de uma forma geral, e não em MSECO.

Kude *et al.* (2018) exploram as competências para perfis de gerentes de parcerias para a governança de ECOS. Eles ainda discutem como esses perfis ajudam a iniciar, crescer e manter ECOS. Os autores discutem estratégias para recrutar, selecionar e treinar gerentes de parceria visando a governança de ECOS. As competências identificadas envolvem: atenção às situações, *networking*, relacionamentos e programas de parceria. Uma das opiniões obtidas dos profissionais entrevistados, no estudo, é de que a governança em ECOS precisa ser estruturada como forma de reduzir custos. E, que isto, facilitará a conexão da organização com parceiros e setores internos. Apesar dos autores discutirem papéis que ajudem a apoiar a governança, como em alguns dos trabalhos relacionados, o foco não é governança de desenvolvedores e MSECO.

Vorraber *et al.* (2018) exploram um framework para alinhar objetivos entre desenvolvedores e gerentes de negócios em torno de um projeto de software *open-source* e livre. O framework, que tem como foco gerentes de projeto, ajuda a ganhar insights relacionados a mecanismos de valor dentro do ecossistema. A análise, por meio do framework, é feita a partir de um projeto dentro do ecossistema. Os autores indicam que há a necessidade de criar uma visão holística de necessidades, valores e conexões dentro do ECOS na perspectiva dos desenvolvedores. Uma vez que este é um ator essencial para o sucesso e crescimento de contribuições no ECOS.

2.5 Considerações finais

Com os conceitos e análise dos trabalhos relacionados, incluindo revisões da literatura técnica, percebeu-se a necessidade por analisar o corpo de conhecimento. Após a realização de um mapeamento sistemático da literatura sobre governança de desenvolvedores em ECOS, extraiu-se um conjunto de estratégias e um conjunto de definições.

Tanto as estratégias quanto as definições para governança de desenvolvedores, como foram propostas a partir da comunidade científica, precisam da análise por profissionais da área de DevRel. Isto permite o entendimento da relevância do corpo de conhecimento identificado. Essa caracterização ajudará a entender que estratégias têm sido utilizadas pelos profissionais. Para isso no próximo capítulo são apresentados um conjunto de estudos que ajudam a construir um modelo de governança de desenvolvedores tendo como base o corpo de conhecimento.

CAPÍTULO 3 – ESTUDOS PARA CONCEPÇÃO E REFINAMENTO DO DEVGO: MODELO DE GOVERNANÇA DE DESENVOLVEDORES

*Neste capítulo apresenta-se o planejamento, execução e análise dos resultados de um conjunto de estudos para a concepção e refinamento do modelo de governança proposto nesta tese. O modelo é chamado de DevGo (do inglês, **Developer Governance**).*

3.1 Visão geral dos estudos

A partir do mapeamento sistemático apresentado no capítulo anterior, foi possível extrair 42 estratégias e uma definição para a governança de desenvolvedores em ECOS. Para construir e refinar o modelo DevGo (do inglês, **Developer Governance**), foram realizadas as seguintes etapas de pesquisa, como já descrito na Seção 1.6 sobre a metodologia de pesquisa:

1. Pesquisa de opinião com profissionais para a análise da relevância das 42 estratégias identificadas no mapeamento sistemático (Seção 3.2);
2. Entrevistas com profissionais de DevRel para refinamento da estrutura da primeira versão do DevGo (Seção 3.3);
3. Identificação e análise de um conjunto de lições aprendidas de DevRel (Seção 3.4);
4. Estudo exploratório e revisão por pares com especialistas para a análise de repositórios (Seção 3.5);
5. Pesquisa de opinião sobre percepção de valor em DevRel para a governança de desenvolvedores (Seção 3.6).

3.2 Pesquisa de opinião com especialistas

3.2.1 Objetivo do Estudo e Questão de Pesquisa

Esta pesquisa de opinião foi planejada e executada com o objetivo de **analisar** estratégias para a governança de desenvolvedores **com o propósito de** caracterizar **com respeito** à relevância **do ponto de vista** dos profissionais de DevRel **no contexto de** atividades de governança de desenvolvedores em MSECO.

A questão de pesquisa investigada neste estudo é: “Qual a relevância de cada uma das estratégias para a governança de desenvolvedores em MSECO?”

Métrica: Percentual de estratégias relevantes para a governança de desenvolvedores em MSECO (lista de estratégias consolidada a partir do conjunto inicial ou adicionadas pelos participantes).

3.2.2 Instrumentação

Foi proposto um questionário aos participantes com o objetivo de caracterizar a sua formação no que se refere à sua experiência e aos ecossistemas nos quais atua/atuou, além do Termo de Consentimento Livre e Esclarecido – TCLE. Além disso, foi preparado um segundo questionário para avaliar a relevância das estratégias para a governança de desenvolvedores no contexto de MSECO por meio da questão de pesquisa citada na subseção anterior.

Foi utilizada a escala ordinal de Likert, oferecendo as opções: 0. Sem relevância; 1. Baixa relevância; 3. Média relevância; e 5. Alta relevância. Além disso, havia uma questão aberta “Qual a sua opinião sobre as estratégias apresentadas? Há alguma outra estratégia?” O questionário referente à avaliação está disponível, além do **Apêndice B**, em: <https://goo.gl/forms/gKagSqWex6nHIQBw1>.

3.2.3 Execução do Estudo

Para classificar o conjunto de 42 estratégias de governança de desenvolvedores, foram contatados 60 gerentes de DevRel identificados no LinkedIn¹⁴ com o título de cargo envolvendo o termo “*developer relations*” – 18 deles responderam à pesquisa de opinião, dando 85% de nível de confiança sobre a quantidade de participantes de acordo com a fórmula de [Hamburg 1980] e uma taxa de resposta de 30%. A estratégia aplicada para a seleção dos participantes foi a “amostragem probabilística” com o objetivo de eliminar a subjetividade na escolha e obter uma amostra não enviesada e representativa da população alvo.

Todos os participantes, caracterizados na **Tabela 5**, trabalham ou trabalharam com pelo menos um dos seguintes ecossistemas: Android, Blackberry, iOS, Windows, Nokia S40, Symbian. Os participantes também trabalham em subsidiárias dessas organizações no Brasil, China, EUA, Israel, Canadá e México. Os participantes da pesquisa de opinião possuem uma média de 5 ($\pm 3,06$) anos de experiência profissional em DevRel.

O cálculo da relevância de cada estratégia utilizou como base os anos de experiência do participante e o ecossistema em que trabalha ou trabalhou. É necessário diferenciar as respostas dos participantes associando um peso a cada um

¹⁴ <http://www.linkedin.com>

deles, considerando por exemplo, anos de experiência e nível de experiência [Dias Neto e Travassos 2008]. O nível de relevância foi calculado da seguinte forma:

1. Cada participante tem um peso (W_i) associado:

$$W_i = \frac{\text{Número de Ecossistemas} + \frac{\text{Anos de Experiência}}{\text{Mediana}(\text{Anos de experiência de todos participantes})}}{5}$$
2. Então, a resposta de cada participante é multiplicada pelo seu peso (W_i);
3. Depois disso, os resultados para cada estratégia foram somados;
4. Finalmente, foi calculado o nível de relevância em um valor entre 0 e 100% por meio da normalização do valor obtido no passo anterior. Para cada estratégia (j), dividiu-se o valor alcançado no passo anterior pelo valor máximo possível: $Relevância(j) = \frac{Total(j)}{\sum_i^N W(i)*5}$.

Tabela 5. Participantes da pesquisa de opinião.

ID	Ecossistemas	Anos de Experiência	País	Peso
P1	Android	4	Brasil	2
P2	Android, iOS	2	Canadá	2,5
P3	Windows	4	Brasil	2
P4	Windows	2	México	1,5
P5	Android, Windows	6	EUA	3,5
P6	iOS	2	Brasil	1,5
P7	Android	2	Brasil	1,5
P8	Android, iOS, Windows	3	Brasil	3,75
P9	Android, Windows, Symbian, S40	5	Brasil	5,25
P10	Android, iOS, Windows	7	México	4,75
P11	Windows	5	Canadá	2,25
P12	Windows, BlackBerry	3	Brasil	2,75
P13	Android	4	Brasil	2
P14	Android	9	China	3,25
P15	Android, iOS, Windows	10	EUA	5,5
P16	Android, Windows, Symbian	13	México	4,25
P17	iOS	4	EUA	2
P18	Android, iOS, Web	11	Israel	5,75

3.2.4 Análise dos Resultados

Foram selecionadas 13 estratégias (**Tabela 6**) que alcançaram um nível de relevância igual ou maior do que a mediana calculada (74%) a partir das relevâncias do conjunto total. As estratégias mais relevantes abrangem as três categorias de mecanismos de governança: Coordenação de Desenvolvedores - 9 (38%), Criação de Valor - 8 (33%) e Abertura e Controle da Organização - 7 (29%). O conjunto de estratégias fundamentadas no corpo de conhecimento e avaliadas por profissionais serviu de base para a concepção da versão inicial do modelo DevGo.

Baseando-se nas respostas dos participantes o ranking de relevância também ajudou a estabelecer uma visão mais compreensiva da governança de

desenvolvedores. Há uma necessidade de identificar o dinamismo social da governança de desenvolvedores. Isto pode ser analisado a partir de uma visão estruturada (guiada mais aproximadamente pela organização) ou uma visão orgânica (tomada de decisão centralizada, sem muita supervisão direta). Outro aspecto indicado por alguns participantes é a necessidade de adaptação das estratégias para regiões específicas (p.ex.: América Latina, Europa). Isto pode ser feito com um trabalho aproximado dos times de DevRel locais com o time de DevRel global.

A versão inicial do modelo que compõe a abordagem DevGo, foi concebida a partir da análise das estratégias mais relevantes por três pesquisadores com conhecimento em Engenharia de Software. Para isso foram utilizados procedimentos análise temática [Cruzes e Dyba 2011] e de criação de mapas mentais para extrair os componentes principais do modelo. Os passos seguintes de análise temática foram executados: 1) Ler e reler dados de entrevistas; 2) Gerar códigos iniciais; 3) Combinar códigos; 4) Analisar como os dados são suportados por temas; 5) Definir cada tema; e 6) Decidir quais temas fazem contribuições significativas para a compreensão do que está acontecendo no conjunto de dados.

Como pode ser visto na **Figura 3**, a primeira versão modelo DevGo é composta pela definição de governança de desenvolvedores, um conjunto de habilitadores, fases de avanço do desenvolvedor (*Assistência, Entrada, Contribuição e Reconhecimento*), repositórios e um conjunto de objetivos organizacionais.

Tabela 6. As 13 estratégias consideradas relevantes pelos profissionais de DevRel.

#	Estratégia	Relevância
E1	Analisar as barreiras para a participação do desenvolvedor e potenciais remédios para elas.	90%
E2	Estabelecer modelos de parceria analisando indicadores de monitoramento da comunidade.	84%
E3	Analisar o fluxo de progresso do desenvolvedor dentro do ecossistema.	80%
E4	Identificar as necessidades do mercado para construir um processo de desenvolvimento bem definido.	80%
E5	Atender às necessidades dos desenvolvedores para manter a capacidade de absorção e evitar desencorajá-los.	80%
E6	Usar os dados e a rede social do ecossistema para entender melhor os elementos técnicos da plataforma.	79%
E7	Aumentar a maturidade do desenvolvedor com base no feedback.	79%
E8	Analisar o ciclo de vida de desenvolvedores de negócios e gerenciamento de parceiros.	77%
E9	Desenvolver um sistema no qual os desenvolvedores talentosos são escolhidos e seguir em frente com base em suas realizações.	76%
E10	Oferecer conferências de desenvolvedores para permitir a discussão de instruções sobre a plataforma subjacente.	74%
E11	Estabelecer de uma plataforma para publicação e propagação de relatórios de erros, medidas de desempenho, etc.	74%
E12	Facilitar a integração entre desenvolvedores e usuários, agregando valor à plataforma.	74%
E13	Identificar a comunidade com base na estrutura, processos, padrões de interação e desempenho.	74%

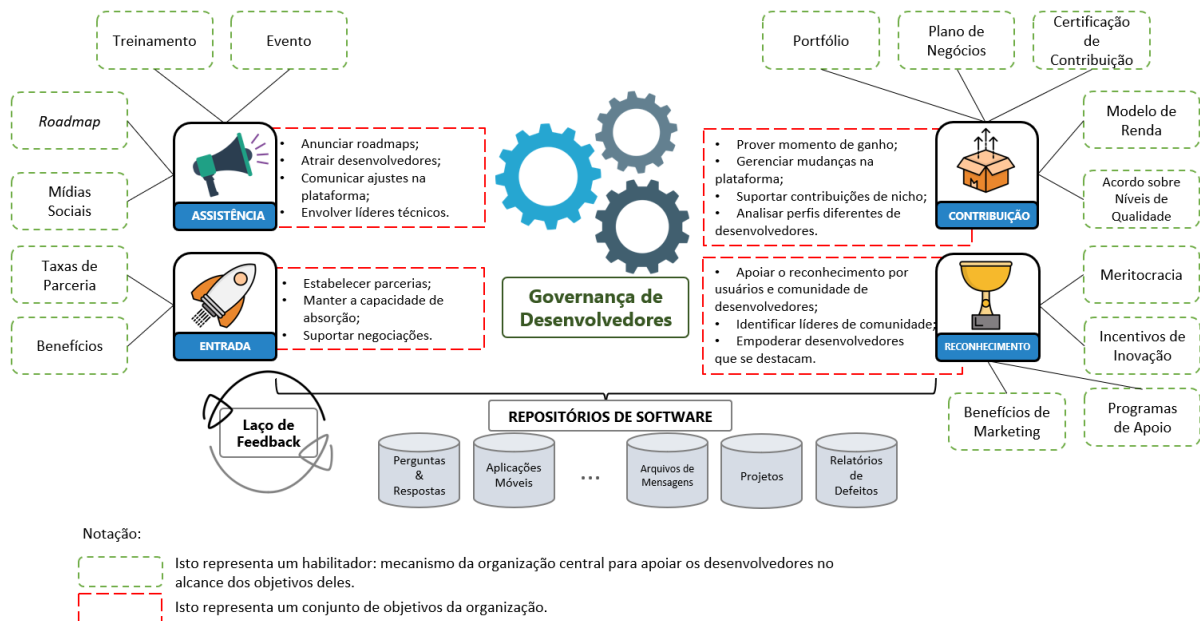


Figura 3. Versão inicial do modelo DevGo.

Para cada fase há um conjunto de habilitadores que são mecanismos da organização para apoiar os desenvolvedores. Os repositórios, nesta versão, servem como uma forma de obter feedback dos desenvolvedores para cada uma das fases de avanço do desenvolvedor.

3.2.5 Ameaças à Validade

As ameaças à validade foram organizadas em 4 categorias: validade de *constructo*, interna, externa, e de conclusão.

Validade de conclusão: realizada por meio de simples demonstração do nível de relevância (ou não) das estratégias extraídas do mapeamento sistemático.

Validade interna: propõe-se selecionar profissionais de DevRel que atuam/atuaram em MSECO. Deste modo, assumiu-se que eles são representativos para a população de profissionais de DevRel em MSECO e que possam dar a perspectiva da organização central sobre a governança de desenvolvedores.

Validade de *constructo*: o estudo está caracterizado pela análise da relevância das estratégias para a governança de desenvolvedores em MSECO. As estratégias foram extraídas de estudos primários, por meio de um mapeamento sistemático da literatura, envolvendo experimentos com desenvolvedores em ecossistemas de software.

Validade externa: os participantes do estudo em geral podem ser considerados representativos para a população de profissionais de DevRel. Para a avaliação do nível de envolvimento na governança de desenvolvedores no contexto de

MSECO, os dados do questionário sobre a experiência dos participantes foram analisados.

Os materiais utilizados no estudo podem ser considerados representativos e atuais para o problema em análise. Tais materiais são formados pelas estratégias que estão relacionadas à governança de desenvolvedores em MSECO.

3.2.6 Conclusões do Estudo

A análise da relevância de cada uma das 42 estratégias, extraídas por meio do mapeamento sistemático, pelos participantes da pesquisa de opinião permitiu obter um conjunto base de 13 estratégias relevantes para a governança de desenvolvedores em ecossistemas. Com isso foi possível gerar uma versão inicial do modelo que descreve a estrutura necessária para apoiar a governança de desenvolvedores. Um ponto importante é que havia a necessidade de refinar o modelo por meio da avaliação de profissionais de DevRel que trabalham nos cenários reais de governança de desenvolvedores em MSECO. Por isto, foi realizado um conjunto de entrevistas como descrito na próxima seção.

3.3 Entrevistas com especialistas

Esta seção apresenta um conjunto de entrevistas planejado e executado com o objetivo de refinar o modelo de governança de desenvolvedores em MSECO.

3.3.1 Objetivo do Estudo

As entrevistas foram planejadas e executadas com o objetivo de **analisar** os elementos que compõem a estrutura do modelo de governança de desenvolvedores **com o propósito de** refinar com respeito à aderência ao cenário real **do ponto de vista** dos profissionais de DevRel **no contexto de** MSECO.

Para isso, foi investigada por meio das entrevistas a seguinte questão de pesquisa: *“Quais são os elementos (conceitos, fases de avanço do desenvolvedor, habilitadores, objetivos) que fazem parte da estrutura de um modelo para a governança de desenvolvedores em MSECO?”*

3.3.2 Instrumentação

Foi proposto um questionário aos participantes, com o objetivo de caracterizar a sua formação no que se refere à sua experiência e aos ecossistemas nos quais atua/atuou, além do Termo de Consentimento Livre e Esclarecido – TCLE. Além disso, foi definido um conjunto de perguntas para guiar a entrevista semiestruturada. As perguntas cobrem a estrutura do modelo e como ela é aderente ao cenário real de

governança de desenvolvedores em ecossistemas. As perguntas de entrevista (PE) são:

- PE1) Você concorda com o conceito e com a estrutura do modelo de governança de desenvolvedores apresentado?
- PE2) Qual a sua opinião sobre ajustes no conceito e a estrutura do modelo de governança de desenvolvedores apresentado?
- PE3) Você concorda com as fases apresentadas que são relacionadas ao fluxo de avanço do desenvolvedor dentro de um ecossistema?
- PE4) Quais são os ajustes necessários nas fases?
- PE5) Você concorda com os habilitadores de governança de desenvolvedores que estão associados as fases de avanço do desenvolvedor?
- PE6) Quais são os ajustes necessários nos habilitadores?
- PE7) Você concorda com os objetivos organizacionais apresentados?
- PE8) Há algum objetivo que não foi apresentado e algum ajuste nos existentes?

3.3.3 Execução do Estudo

Para avaliar o DevGo foram realizadas entrevistas semiestruturadas com 15 profissionais experientes em DevRel (por exemplo, diretores e gerentes). Esses participantes foram selecionados seguindo o mesmo procedimento usado na para a pesquisa de opinião. No **Apêndice C** são apresentados os perfis de cada um dos participantes, eles tiveram uma média de 7 ($\pm 2,03$) anos de experiência profissional em DevRel. Foram entrevistados participantes dos seguintes países: Alemanha, Brasil, Canadá, EUA, Inglaterra e México. Eles avaliaram o modelo DevGo e a definição de governança de desenvolvedores, que foi resumida a partir de evidências do mapeamento sistemático. Das entrevistas, foram obtidas e aceitas um total de 28 sugestões.

3.3.4 Análise dos Resultados

Em relação às quatro sugestões voltadas ao conceito de governança de desenvolvedores, que foram sintetizadas a partir dos estudos encontrados no mapeamento sistemático, há a confirmação da abrangência do conceito, como diz o seguinte participante:

“O conceito apresentado parece muito bom e parece abranger a maioria dos itens.” (P1)

No entanto, o conceito precisa deixar clara a importância de se considerar diferentes perfis de desenvolvedores durante a governança de desenvolvedores. A segmentação do desenvolvedor deve ser levada em conta e que as comunidades de desenvolvedores específicas devem ser fomentadas por meio de parcerias para aprimorar a criação de nicho:

“Deve considerar a segmentação do desenvolvedor.” (P4)

Um outro participante (P8), apesar de entender o conceito, alerta que o conceito não é bem entendido entre as organizações que se beneficiam do cenário de ecossistemas. Ou ainda que a governança de desenvolvedores não é de conhecimento da comunidade de desenvolvedores dentro do ecossistema, sendo este também um comentário do participante P7:

“Certamente o conceito de governança do desenvolvedor proposto é claro e descreve como deve ser a interação com as partes interessadas e como cada uma delas será beneficiada. Acredito que não seja tão claro na maioria das empresas ou não seja conhecido pela rede de desenvolvedores.” (P8)

“Minha única preocupação é sobre a comunicação deste modelo com a rede do desenvolvedor e como seria o gerenciamento de conflitos entre, principalmente, os desenvolvedores. Apesar desse ponto, acho que é uma boa proposta.” (P7)

O participante P12 indica que a necessidade de incluir a equipe de Marketing de Desenvolvedores em conjunto com os profissionais de DevRel para que seja possível a governança de desenvolvedores:

“O Developer Marketing trabalha em conjunto com o Developer Relations, ajudando a melhorar o conhecimento do conteúdo, fornecendo pesquisas de mercado, dando suporte a eventos do desenvolvedor e criando marcas consistentes.”

Desta forma a definição de governança de desenvolvedores em ecossistemas consiste em: *“É um conjunto de mecanismos (ou seja, marketing de desenvolvedor e relações com desenvolvedores - evangelismo e advocacia) para apoiar relacionamentos ganha-ganha entre uma comunidade próspera de desenvolvedores e uma organização com o objetivo de garantir/monitorar o bem-estar econômico e social dos desenvolvedores. É preciso considerar a segmentação do desenvolvedor para manter a vitalidade do ecossistema e reduzir custos e riscos”.*

A análise de sugestões, **relacionadas à Estrutura do Modelo**, as duas primeiras sugestões estão relacionadas a estrutura do modelo que em sua versão inicial não demonstrava com clareza os benefícios para a organização e para os

desenvolvedores, assim como, a sequência e envolvimento entre os elementos da estrutura do modelo e o ciclo de feedback. E, estes podem ser alguns dos motivos que também leve a seguinte percepção do participante P7: *“No geral, meu palpite é que o modelo apresentado faz sentido como ponto de partida”*. Abaixo os comentários dos participantes sobre esses aspectos:

“Eu gostaria de ver com mais clareza quais são os benefícios dos objetivos organizacionais (para a organização) e os benefícios (para os desenvolvedores) - talvez mudando a maneira como é mostrado na imagem.” (P1)

“Eu entendo que há uma sequência que poderia ser evidenciada, mostrando causa e efeito 1) Fases (o núcleo do modelo); 2) Facilitadores; e 3) Metas. Seria bom também evidenciar os custos e riscos que são citados, e como esse modelo está mitigando isso.” (P2)

“Praticamente completo, apenas o ciclo não está claro.” (P3)

“Parece que você tem praticamente tudo coberto com esses facilitadores, no entanto, este modelo tem que ser muito empírico para que isso seja ajustado no tempo de acordo com as informações coletadas do ciclo de feedback.” (P6)

Sobre as etapas a partir do comentário do participante P15, foi possível perceber que era necessário expandir as quatro etapas presentes na versão inicial do modelo:

“A estrutura que usamos no nosso setor de DX envolve a conscientização, avaliação, compra/construção, publicação, encaminhamento/defesa (ou algo parecido).” (P15)

O participante P5 sugeriu que fossem incorporados ao modelo valores ágeis como colaboração, propriedade, transparência e confiança para governar os desenvolvedores dentro de um ecossistema.

Ainda sobre as etapas, o participante P6 não conseguiu perceber a ligação entre elas, o que pode indicar que não foi possível dar a ideia de etapas de progresso do desenvolvedor dentro do ecossistema:

Eu tenho um feedback sobre a ausência de um vínculo claro entre "contribuir", "participar" e "reconhecer" em uma plataforma de gerenciamento de projetos sociais e de compartilhamento de fontes no estilo do Github. Acredito que fazer das plataformas sociais uma parte de "contribuir" e "reconhecer" é um forte ganho na proposta de valor. Eu gostaria de citar, como um exemplo de um ambiente de desenvolvedor que

faz isso bem, a migração de partes do mecanismo Unity para projetos oficiais do Github para alavancar o acesso às contribuições da organização e da comunidade.”

As relações com desenvolvedores envolvem um grupo de engenheiros de software que são extrovertidos e excelentes em falar em público, como indicado pelo participante P13: *“O DevRel inclui evangelismo tecnológico, gestão de líderes da comunidade, treinamento e documentação. A parte de advocacia trabalha com desenvolvedores existentes, já o evangelismo foca na divulgação da “palavra” da organização. Como característica comum, tanto advocacia quanto o evangelismo devem construir confiança ajudando outros departamentos da organização.”*

A partir do comentário do participante P13 consideramos dentro da área de foco que trata sobre DevRel, tanto o evangelismo quanto a advocacia. Desta forma, o DevRel serve como uma interface entre os desenvolvedores, as equipes técnicas e de produto do ecossistema. Neste contexto, o evangelismo se concentra na promoção e conscientização. Por outro lado, a advocacia prioriza a coleta de feedback de produtos dos desenvolvedores.

Em relação aos objetivos para cada fase mencionados no modelo (**Figura 3**), percebeu-se a partir da análise da sugestão do participante P1 que é necessário deixar claro no modelo um melhor alinhamento entre os objetivos da organização e como isso pode estar presente: *“Necessidade de afirmar mais claramente o que essas metas representam para a organização em termos de ganho (usuários, receitas, visibilidade, etc.)”*.

Outro ponto é inserir objetivos que estejam relacionados a venda de produtos e serviços da organização, como indicado pelo participante P2: *“Na fase de Contribuição, eu incluiria algumas metas relacionadas a vendas, afinal, esse é o propósito da organização.”* Em relação a sinergia, ou seja, relacionamento ganha-ganha entre a organização e os desenvolvedores não fica claro como é possível suportá-la:

“Eu acho que eles são bons o suficiente também. É muito importante ter em mente que essas relações devem ser um modelo ganha-ganha e, para o desenvolvedor, qual é o real impacto de seu trabalho e uma expectativa de ser recompensada no mesmo nível.” (P7)

Outro ponto é que a fase de sensibilização é importante para divulgar, gerar material para o departamento de marketing da organização e fortalecer as relações públicas da organização, refinamento percebido a partir do comentário do participante

P12: *“Sensibilização Tecnológica: divulgar, gerar leads para marketing, relações públicas puramente tecnológicas”.*

O participante P14, o mais experiente dentre os participantes, fez uma consideração sobre quando o objetivo da organização central é evangelizar ou advogar dentro da governança de desenvolvedores em ecossistemas: *“Se uma empresa deseja se concentrar na sensibilização ou potencialmente manter os custos baixos participando de menos atividades, então eles podem decidir se concentrar no evangelismo. Os evangelistas podem se concentrar muito mais em atividades que resultam em aquisições, como documentos, postagens em blogs, patrocínio de eventos e palestras. Veja que as atividades mencionadas podem ajudar na sensibilização e na contribuição. Se uma empresa priorizar a coleta de feedback de produtos de desenvolvedores, a advocacia pode ser a melhor abordagem. Os advogados podem realizar todas as atividades acima como parte de seu programa de relações com o desenvolvedor. Particularmente, estes profissionais estão mais ligados com atividades que envolvam o produto ou a retenção de desenvolvedores.”*

Essas considerações conduziram o modelo a possuir Áreas de Foco, como forma de agrupar melhor os objetivos da organização e como o desenvolvedor pode se beneficiar de estar engajado em um ecossistema. Assim foi possível ajustar as relações entre as áreas de foco, incluindo artefatos de entrada e gerados dentro de cada área de foco. A área de foco ainda fortalece o entendimento de onde a organização deseja investir para expandir o ecossistema.

No que está relacionado às **sugestões sobre o fluxo de avanço do desenvolvedor**, há a necessidade de evidenciar uma direção no fluxo da imagem, assim como, o tempo de duração de cada uma das fases:

“Pode ser mostrada mais claramente a direção do fluxo na imagem.” (P1)

“Mostrar a temporalidade dessas fases.” (P9)

Em relação a fase de referência, o participante P10, comentou sobre o escopo: *“Relacionamentos são dois compromissos e representam um investimento de tempo por parte do influenciador. Parcerias estratégicas com líderes da comunidade. Reconhecimento: os desenvolvedores constroem reconhecimento e atingem seus próprios objetivos, levando a interações positivas e relacionamentos mais próximos”.* Ainda sobre escopo o participante P13 comentou sobre a sensibilização: *“Sensibilização – consciência da plataforma e o que faz.”*

Houve ainda sugestões voltadas a uma melhor definição dos elementos que compõem o fluxo de avanço do desenvolvedor, que envolvem a importância de cada um dos elementos e novas fases do fluxo, como segue:

“Integração tem um conceito pouco difuso, não muito claro qual é o papel dele. É uma organização separada?” (P6)

“Fases: Descoberta, Integração, Suporte e Advocacia. Podem ser agrupados em Lançamento, Crescimento e Maturidade” (P11)

As **sugestões relativas aos facilitadores** envolvem a adição de novos facilitadores ou a presença em outra fase de um que já foi identificado previamente. Como sugerido:

“Incluiria suporte técnico na fase de integração.” (P2)

“Impacto de treinamento as 4 fases.” (P3)

“Concordo com os facilitadores, com a ressalva da supracitada falta de ‘ferramentas de colaboração de projetos sociais’ na parte ‘contribuindo’ e ‘reconhecendo’.” (P5)

Há ainda a necessidade de descrever os benefícios dos facilitadores para os desenvolvedores, de não misturar atividades de governança de desenvolvedores com canais de comunicação com o desenvolvedor:

“Parece muito bom e eu não vejo nenhum outro facilitador do que os retratados. (faltando apenas benefícios mais claros para o desenvolvedor - por exemplo: o que um modelo de receita, etc. representaria para eles).” (P1)

“Os facilitadores não devem misturar atividades e canais.” (P4)

3.3.5 Ameaças à Validade

As ameaças à validade foram organizadas em 4 categorias: validade de *constructo*, interna, externa, e de conclusão.

Validade de conclusão: realizada por meio da análise da estrutura do modelo direcionada pelas questões da entrevista.

Validade interna: propôs-se entrevistar profissionais de DevRel que atuam/atuaram em MSECOS envolvidos em governança de desenvolvedores. Deste modo, assumiu-se que eles são representativos para a população de evangelistas em MSECO.

Validade de *constructo*: O estudo está caracterizado pela análise dos elementos envolvidos na estrutura de um modelo concebido a partir de estratégias relevantes para a governança de desenvolvedores em MSECO.

Validade externa: os participantes do estudo em geral podem ser considerados representativos para a população de profissionais de DevRel. Para a avaliação do nível de envolvimento em governança de desenvolvedores no contexto de MSECO, os dados do questionário sobre a experiência dos participantes foram analisados.

3.3.6 Conclusões do estudo

O DevGo foi refinado e, desta forma, passou a cobrir os aspectos indicados por profissionais de DevRel representando a perspectiva da organização central. Para isso foram inseridas áreas de foco e a estrutura do modelo foi revisitada, como apresentado na **Figura 4**. As fases foram redistribuídas em níveis de avanço e cada um deles em estágios. Os habilitadores passaram a ser chamados de facilitadores e foram associados aos estágios do DevGo. Além disso, entre cada área de foco foram representadas as setas que informam trocas de objetos de valor.

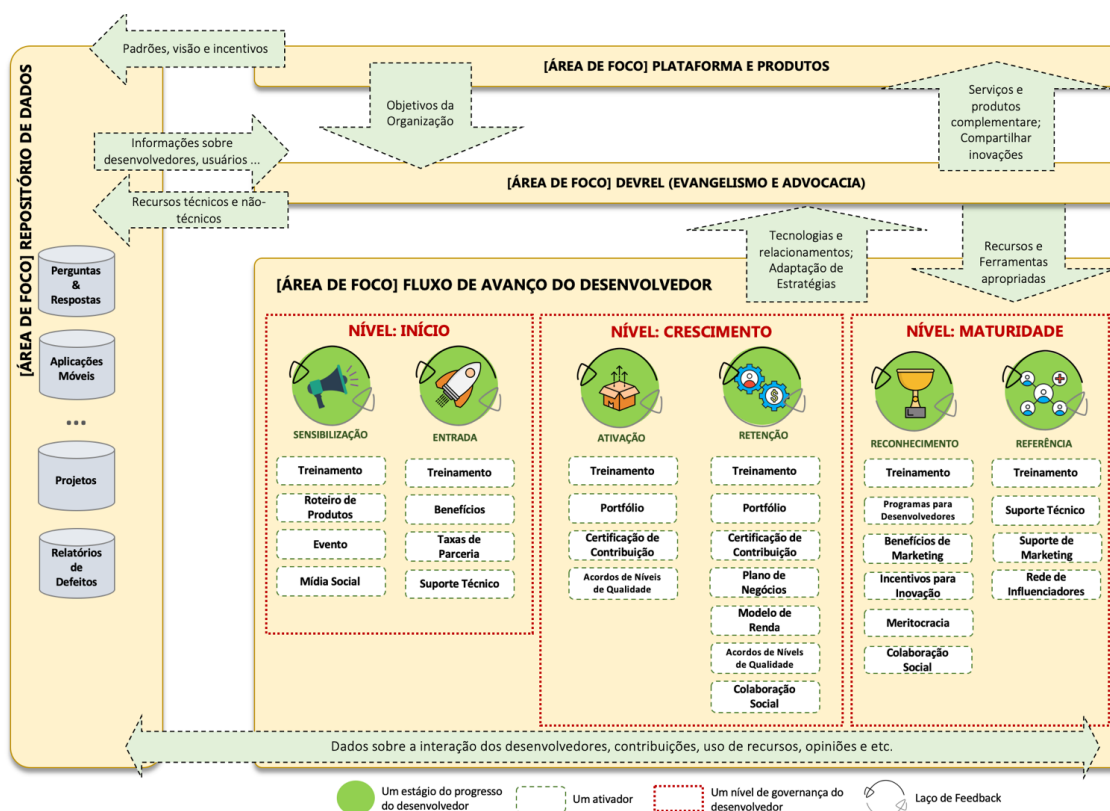


Figura 4. Segunda versão do modelo DevGo.

Um aspecto necessário para refinar o DevGo é de alguma forma recuperar experiências adquiridas em DevRel. Para isto, foram analisadas lições aprendidas de profissionais de DevRel como descrito na próxima seção.

3.4 Identificação de lições aprendidas em DevRel

Uma lição aprendida, segundo a definição do PMBOK [Bayona *et al.* 2018], descreve a experiência adquirida ao longo de um projeto. Registrar, documentar, e principalmente, divulgar as lições aprendidas é uma maneira de evitar que tais problemas voltem a ocorrer em projetos futuros. Este cenário também é aplicável à governança de desenvolvedores em MSECO por meio de atividades de DevRel.

No caso desta tese, as lições aprendidas estão relacionadas aos estágios que compõem o DevGo. As lições aprendidas poderão ser utilizadas por profissionais de DevRel para ajudar desenvolvedores externos a avançar dentro do MSECO e a mitigar possíveis riscos.

Como forma de encontrar lições aprendidas relatadas por profissionais de DevRel, foi escolhido o repositório de compartilhamento de experiências, o *Medium*¹⁵, por também se tratar de um espaço onde profissionais de DevRel compartilham suas experiências resultantes do trabalho de governança de desenvolvedores.

3.4.1 Objetivo do Estudo

O mapeamento sistemático de lições aprendidas no *Medium* foi planejado e executado com o objetivo de **identificar** as lições aprendidas de DevRel **com o propósito de** caracterizar respeito à governança de desenvolvedores **do ponto de vista** dos pesquisadores **no contexto de** ecossistemas de software.

3.4.2 Estratégia de busca e extração de dados

Para a busca e extração de dados foram utilizados procedimentos para mapeamento de literatura cinza indicados por Garousi *et al.* (2016). A base digital para este estudo, como mencionado na seção anterior, foi o *Medium* que é um ambiente de compartilhamento de histórias, perspectivas e ideias. Dentro do *Medium* foi aplicada seguinte *string* de busca: “*developer relations*”. Foram definidos como critérios de inclusão:

1. Compartilhar lição aprendida, conceito ou estratégia (técnica, metodologia, abordagem, *framework*) ou dentro da área de DevRel;
2. Ser escrito em inglês;
3. O autor do artigo ter experiência comprovada em DevRel por meio da análise de perfil no LinkedIn¹⁶.

¹⁵ <https://medium.com/>

¹⁶ <http://www.linkedin.com>

Com a utilização da *string* de busca foi possível encontrar 229 artigos de profissionais da área. Destes artigos, após serem aplicados os critérios de inclusão, foram selecionados 100 artigos para análise e extração das lições aprendidas. O processo de filtragem dos artigos passou por revisão por pares com três especialistas em Engenharia de Software Experimental, Ecossistema de Software e Revisões Sistemáticas de Literatura.

Os artigos, como apresentado na **Figura 5**, estão distribuídos entre os anos de 2011 e 2018. Como é possível perceber, o interesse por entender a governança de desenvolvedores, do ponto de vista de DevRel, é crescente no meio da indústria, assim como, identificado no mapeamento sistemático (Capítulo 2).

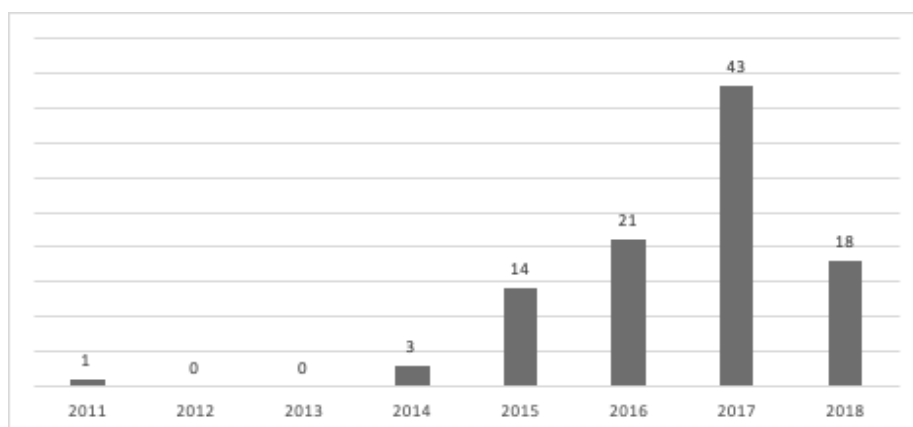


Figura 5. Quantidade de artigos selecionados do *Medium* por ano.

A partir dos artigos identificados foram extraídas 267 lições aprendidas dos artigos (**Figura 6**) que tivessem relação o relato de lições aprendidas de DevRel em ações que estejam relacionadas a governança de desenvolvedores. Na próxima seção são descritas a análise e a síntese das lições por estágios do modelo.

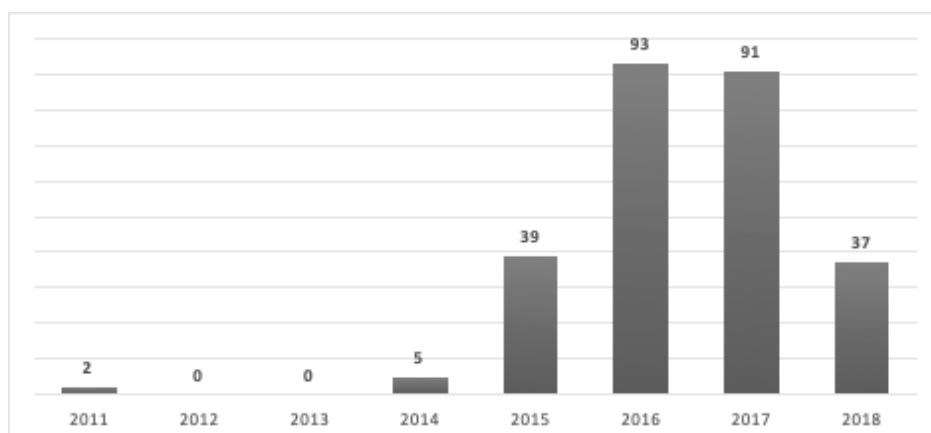


Figura 6. Quantidade de lições aprendidas extraídas por ano.

3.4.3 Análise dos resultados

Para sintetizar o conjunto de 267 lições aprendidas foram utilizados procedimentos de análise temática [Cruzes e Dyba 2011] para classificar as lições em categorias que representassem os estágios do modelo DevGo: sensibilização, entrada, ativação, retenção, reconhecimento e referência. As lições aprendidas similares foram agrupadas para formar uma única. Assim, chegou-se a um total de 63 lições aprendidas com a distribuição indicada na (**Figura 7**) e a seguir em ordem decrescente: 27% para sensibilização (17), 17,5% para referência (11), 17,5% para reconhecimento (11), 15,8% para retenção (10), 11,1% para entrada (7) e 11,1% para ativação (7).



Figura 7. Distribuição de lições aprendidas por estágio do modelo.

As lições aprendidas são apresentadas no próximo capítulo a medida em que são apresentados os estágios que compõem o modelo DevGo.

3.4.4 Ameaças à Validade

Validade de conclusão: realizada por meio da análise do conjunto de artigos. O artigo deveria tratar de alguma lição aprendida para a governança de desenvolvedores e ser escrito por profissional de DevRel.

Validade interna: as lições são baseadas na experiência de profissionais de DevRel envolvidos em governança de desenvolvedores. Deste modo, assumiu-se que as lições e os profissionais são representativos para a população de profissionais de DevRel em MSECO.

Validade de *constructo*: o estudo está identificação e análise das lições aprendidas envolvidas nos estágios do modelo de governança de desenvolvedores. Para a condução do mapeamento e da análise das lições foram utilizados procedimentos indicados, respectivamente, por Garousi *et al.* (2016) e Cruzes e Dyba (2011). Os estágios do modelo concebido a partir de estratégias relevantes para a

governança de desenvolvedores em MSECO, passaram por pesquisa de opinião e entrevistas por especialistas na área.

Validade externa: os participantes e as lições aprendidas são considerados representativos em relação a população de profissionais de DevRel. As lições ainda passaram por análise em relação aos estágios do modelo que, como indicado por profissionais em estudos anteriores, representam o fluxo real de engajamento de um desenvolvedor dentro de um MSECO.

3.4.5 Conclusões do Estudo

O estudo sobre a identificação e análise de lições aprendidas em DevRel para governar desenvolvedores conduz ao entendimento que a indústria também tem investido em um corpo de conhecimento na área. É possível perceber ainda pela análise das recomendações que não há uma estrutura em que se discuta a governança de desenvolvedores dentro de ecossistemas.

A grande maioria das experiências dos profissionais está relacionada aos estágios de sensibilização, ou seja, de comunicar os benefícios do ecossistema como forma de atrair desenvolvedores. Claro que há experiências em cada um dos outros estágios que são indicados no DevGo, porém a sensibilização se destaca na indústria.

3.5 Estudo Exploratório sobre Repositórios

3.5.1 Objetivo do Estudo

Esta seção apresenta um estudo exploratório planejado e executado **com o objetivo de** analisar a estrutura dos repositórios de conteúdo do MSECO, **a fim de** caracterizar **em relação às** suas características comuns do **ponto de vista** de pesquisadores **no contexto de** MSECOs da Apple, Google e Microsoft.

A análise da estrutura dos repositórios de MSECO pode ajudar a conhecer suas características comuns em relação, por exemplo, à forma como os ecossistemas competem, comparação entre estratégias de governança e efeitos em aplicações móveis ou comunidades de desenvolvedores. Como tal, uma organização central pode adaptar as estratégias de governança para manter o ecossistema vibrante, bem como para atrair desenvolvedores e usuários [Axelsson e Skoglund 2016]. Todo detalhamento deste estudo se encontra publicado em [Fontão *et al.* 2018c].

3.5.2 Instrumentação

Como instrumentos para análise neste estudo foram utilizados o conjunto de repositórios discutido em [Fontão 2016]. Além desses foram analisados aqueles

identificados nos seguintes estudos que mapeiam a literatura existente para a mineração de repositórios de software: [Cosentino *et al.* 2017][Meldrum *et al.* 2017].

3.5.3 Execução

Na primeira etapa, um pesquisador de engenharia de software (especialista em MSECO) acessou cada repositório e analisou a estrutura dos repositórios por meio de um mapa mental. Em seguida, na segunda etapa, os elementos comuns foram inseridos em um novo mapa mental para produzir uma estrutura comum dos repositórios.

Como a primeira etapa da execução envolveu um único pesquisador, o mapa mental precisou passar por um processo de revisão por pares. Desta forma, quatro pesquisadores (especialistas em MSECO), avaliaram o mapa mental construído em conjunto com as informações dos instrumentos. Por meio de reuniões de consenso definiram um conjunto de características que se agrupavam. Então, foi possível gerar um conjunto de categorias para a classificação de repositórios visando a governança de desenvolvedores em MSECO.

3.5.4 Análise dos resultados

Com base neste estudo, foi observado que os repositórios utilizados nos MSECO são: Stack Overflow, Android Developers, Apple Developers, Discourse, Github, Bitbucket, Reddit, Hacker News, Facebook, Twitter, Confluence, Gitter, Slack, Telegram, WhatsApp, App Store, Google Play e Android Developers e Apple Developers.

Para que o texto não se estenda, a descrição de cada repositório e suas categorias são apresentadas no Capítulo 5, na Seção 5.1.1.4, que trata sobre o monitoramento como área de foco do modelo DevGo.

3.5.5 Ameaças à Validade

Validade de conclusão: realizada por meio da análise de estruturas de repositórios para derivar categorias. Para isto, foram envolvidos especialistas na área de MSECO e com conhecimento em repositórios.

Validade interna: as categorias são derivadas de características comuns dos repositórios que foram extraídas de trabalhos com evidência experimental. Deste modo, assumiu-se que as características e as categorias são representativas para a população de repositórios em MSECO.

Validade de constructo: o estudo está identificação de características comuns de repositórios de software em MSECO. Para a condução da identificação e revisão das características os pesquisadores utilizaram atividades de revisão por pares.

Validade externa: as categorias identificadas cobrem diferentes tipos de repositórios existentes para o monitoramento de MSECO. Para a inclusão de outras repositórios basta se realizar a análise das características por categorias.

3.5.6 Conclusões do Estudo

Um total de sete categorias foram identificadas: (1) Perguntas e Respostas, (2) Lista de E-mails e Fóruns, (3) Ambiente Social de Codificação, (4) Website Social de Notícias, (5) Redes Sociais, (6) Comunicação do Time, (6) Loja de Aplicações e (7) Painel de Controle do Desenvolvedor.

Estas categorias envolvendo suas descrições e exemplos de repositórios associados serviram para refinar o DevGo, especificamente, a área de foco de monitoramento do DevGo. Desta forma, o modelo (**Figura 8**) passou a conter uma especificação de repositórios que podem ser utilizados para o monitoramento da governança de desenvolvedores em MSECO. As lições aprendidas, identificadas no estudo anterior, foram associadas aos estágios.

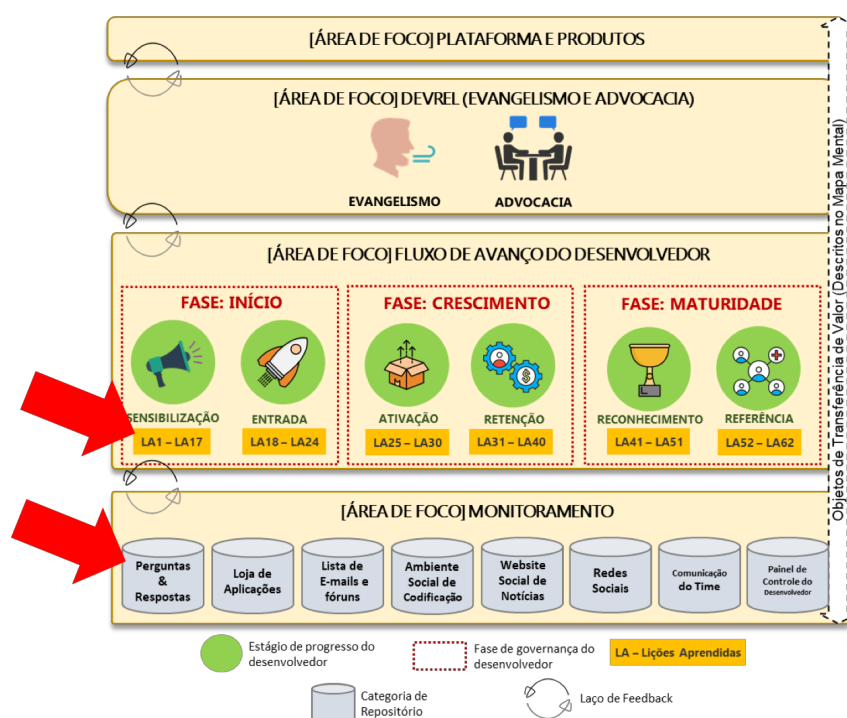


Figura 8. A terceira versão do modelo DevGo.

As categorias, também, foram utilizadas para entender as motivações dos profissionais de DevRel para utilizar repositórios para a criação de valor dentro do ecossistema. O estudo sobre a percepção de valor é descrito na próxima seção.

3.6 Pesquisa de opinião sobre Percepção de Valor

A criação de valor é definida por Bowman e Ambrosini (2000) como a diferença entre o uso e troca de valor em vários níveis de análise. Hyrynsalmi *et al.* (2014) analisam valor a partir do nível de análise das aplicações móveis e os desenvolvedores delas. Neste estudo, o nível de análise é de DevRel.

Este estudo se utilizou de um conjunto de repositórios, assim como de um conjunto de motivações para o entendimento da percepção de valor em DevRel. Desta forma, aprimorar os objetos de transferência de valor do modelo DevGo. Em relação ao conjunto de repositórios adotados, baseou-se nas categorias de repositórios do modelo DevGo apresentadas no Capítulo 5, Seção 5.1.1.4, na **Tabela 26**. O conjunto de motivações (**Tabela 7**) foi construído em cima das motivações apresentadas nos estudos de Roberts *et al.* (2006), Ze e zhang (2008) e Mendes *et al.* (2017).

Tabela 7. Lista de motivações.

Motivação	Descrição
Satisfação do desenvolvedor	O impacto gerado na satisfação do desenvolvedor, por exemplo, maior valor percebido do produto, de recurso técnico ou se o recurso atende a aquilo que o desenvolvedor está esperando.
Crescimento da comunidade	O impacto sobre a quantidade de desenvolvedores de produtos e outros recursos do ecossistema, por exemplo, aquisição de novos desenvolvedores.
Custo operacional	O impacto sobre os custos operacionais para a governança de desenvolvedores. Ajuda a analisar aquilo que reduz os custos operacionais <i>versus</i> aumenta os custos operacionais.
Competitividade do produto	O impacto sobre a competitividade dos produtos do ecossistema na perspectiva do desenvolvedor e usuários dos produtos, por exemplo, ser competitivo em avaliações da qualidade dos produtos.
Custo de Manutenção	O impacto sobre os custos de manutenção de produtos do ecossistema, com isso é possível identificar e analisar aquilo reduz os custos de manutenção <i>versus</i> aumenta os custos de manutenção.
Tempo para Comercialização	O impacto no momento de lançamento de produtos no mercado. Ajuda em decisões do tipo: “o lançamento será lançado nos próximos 3 meses ou nos próximos 12 meses?”
Estratégia a longo prazo	O impacto para o que a organização considera seu valor e custo de vida, por exemplo, a perspectiva da organização para um horizonte de investimento de 2 a 3 anos. O pensamento de longo prazo leva em consideração o aspecto operacional.
Valor financeiro	O valor comercial esperado dos produtos e recursos técnicos (entrada/venda de pedidos) no canal direto do desenvolvedor.
Esforço para desenvolvimento do produto	O impacto sobre o esforço para concluir o lançamento de novos produtos e recursos técnicos.
Custo de entrega	O impacto sobre os custos de entrega do produto. Análise sobre aquilo que reduz os custos de entrega <i>versus</i> aumenta os custos de entrega.
ROI (Retorno sobre o investimento)	O impacto sobre o produto, quando se considera o valor estimado trazido pelo produto em relação aos custos necessários para o desenvolvimento do mesmo.
Facilitador de eficiência técnica	O objeto em análise melhora a tecnologia e a arquitetura do ecossistema permitindo, por exemplo, a economia de custo futura, criação de novos produtos/recursos em cima dele.

Tanto a lista de categorias de repositórios quanto as de motivações passaram por filtros com três pesquisadores da área de Engenharia de Software e com três profissionais de DevRel. As categorias e as descrições foram analisadas em reuniões para se chegar a um consenso, primeiramente entre pesquisadores e depois com profissionais de DevRel.

3.6.1 Objetivo do Estudo e Questão de Pesquisa

O objetivo desta pesquisa de opinião, baseado na abordagem GQM [Basili 1994], é **analisar** um conjunto de repositórios e comentários sobre valor em DevRel **com o propósito de** caracterizar **com respeito a** percepção de valor em DevRel e motivações para uso de repositórios **do ponto de vista de** profissionais da indústria **no contexto de** atividades de DevRel em governança de desenvolvedores.

As questões de pesquisa e as métricas que ajudaram a respondê-las são:

Q1. Qual a percepção dos profissionais sobre valor em DevRel?

Métrica: O conjunto de comentários dos profissionais e de itens relacionados a percepção sobre valor em DevRel.

Q2. Quais as motivações que levam você a contribuir/usar os repositórios durante suas atividades de DevRel?

Métrica: Para cada repositório uma lista das motivações com a quantidade e porcentagem de vezes que foram selecionadas. A lista será ordenada de forma decrescente em relação a porcentagem de seleção da motivação pelos profissionais.

3.6.2 Instrumentação

Foi proposto um questionário aos participantes com o objetivo de caracterizar a sua formação no que se refere à sua experiência, aos ecossistemas nos quais atua/atuou, o tamanho da organização que trabalha(ou), além do Termo de Consentimento Livre e Esclarecido – TCLE.

Além disso, foi preparado um segundo questionário para caracterizar que motivações direcionam ao uso de repositórios em atividades de DevRel. A pergunta voltada aos questionários foi a seguinte: “*Quais as motivações que levam você a contribuir/usar os repositórios durante suas atividades de DevRel?*”. Para responder a essa pergunta, o participante foi apresentado a uma matriz onde as linhas listam as motivações e as colunas listam os repositórios, como apresentado na **Figura 9**.

Além disso, havia uma questão aberta “*Qual a sua percepção sobre criação de valor em DevRel?*”. O questionário referente à avaliação está disponível no **Apêndice D**.

	Questions & Answers	Mailing lists/forums	Social Coding Environment	Social News Web Site	Social Network	Team Collaboration	App Store	Developer Dashboard
Long-term strategy	☐	☐	☐	☐	☐	☐	☐	☐
Community Growth	☐	☐	☐	☐	☐	☐	☐	☐
Financial Value	☐	☐	☐	☐	☐	☐	☐	☐
ROI	☐	☐	☐	☐	☐	☐	☐	☐
Delivery Cost	☐	☐	☐	☐	☐	☐	☐	☐
Maintenance Cost	☐	☐	☐	☐	☐	☐	☐	☐
Operational Cost	☐	☐	☐	☐	☐	☐	☐	☐

Figura 9. Matriz com motivações e repositórios da pesquisa de opinião.

3.6.3 Execução do Estudo

Para participar desta pesquisa de opinião, foram contatados 97 profissionais de DevRel identificados no LinkedIn¹⁷ com o título de cargo envolvendo o termo “*developer relations*” - 31 deles responderam à pesquisa de opinião, obtendo uma taxa de resposta de 32%. Este tamanho da amostra leva a 85% de nível de confiança de acordo com a fórmula de [Hamburg 1980]. Estes profissionais não haviam participado ainda de alguma das etapas desta tese. A estratégia aplicada para a seleção dos participantes foi a “amostragem probabilística” com o objetivo de eliminar a subjetividade na escolha e obter uma amostra não enviesada e representativa.

Todos os participantes trabalham ou trabalharam com pelo menos um dos seguintes MSECOS: *Android, iOS, Nokia S40, Nokia S60, Symbian e Windows*. E, também, há participantes que possuem experiência em outros ECOS: *Actions on Google, Amazon Web Services, Facebook, Google Cloud Platform, IBM Cloud, JVM, Linux, Maemo, Microsoft Azure, OSX, Shopify, Twitter e Unity*. Dentre os participantes da pesquisa de opinião há diretores, gerentes, evangelistas, advogados de desenvolvedores e engenheiros de programas de desenvolvedores.

Os participantes possuem uma média de 4,8 ($\pm 3,26$) anos de experiência profissional em DevRel. Em média, eles planejaram e executaram 44,8 ($\pm 64,30$) treinamentos e 44,8 ($\pm 44,62$) eventos (p.ex.: *hackathon*, conferência de desenvolvedores, *meetups*) com desenvolvedores como parte das atividades de DevRel. Considerando o tamanho das organizações nas quais são funcionários: 15 (48,39%) atuam em grandes empresas, 10 (32,26%), em médias e 6 (19,35%), em pequenas empresas.

¹⁷ <http://www.linkedin.com>

3.6.4 Análise sobre a Percepção de Valor em DevRel

Para analisar os comentários dos participantes sobre o que é valor em DevRel foi utilizado como base o framework de Amit e Zott (2001), que trata sobre aspectos de valor em negócios. Este framework foi utilizado por Hyrynsalmi *et al.* (2014) para analisar as fontes de valor em um MSECO a partir das perspectivas de desenvolvedores de aplicações móveis. Além disso, este é um trabalho muito citado na comunidade científica, com cerca de 6357 citações até Abril de 2019.

O framework se organiza em torno de quatro fontes de criação de valor (**Tabela 8**): (1) Eficiência; (2) Retenção; (3) Inovação; (4) Complementaridade. Cada fonte é composta por itens que permitem a sua operacionalização. O produto tratado aqui se refere às contribuições de desenvolvedores e profissionais de DevRel, que envolvem recursos técnicos, como por exemplo, código fonte, eventos para desenvolvedores, soluções técnicas, portais de perguntas e respostas. Já as transações se referem a trocas entre a organização e a comunidade de desenvolvedores. A seguir, são descritos as fontes e os itens de operacionalização.

Eficiência: sugere que a eficiência das transações aumenta quando o custo por transações dentro do ecossistema diminui. Vários mecanismos existem para reduzir custos, por exemplo, os custos de pesquisa por produtos para desenvolvedores e profissionais de DevRel.

Retenção: os desenvolvedores são motivados a participar de transações continuamente e estão dispostos a continuar seu relacionamento com a organização. Uma situação de retenção pode resultar no aumento da vontade dos desenvolvedores consumirem mais produtos do ecossistema.

Complementaridade: envolve os cenários de agrupamento de vários produtos como forma de gerar mais valor do que oferecer o mesmo conjunto de produtos separadamente, ou seja, valor agregado.

Inovação: refere-se à exploração com sucesso de novos produtos e serviços, bem como a introdução de novos métodos de condução e organização do negócio.

Os itens mencionados na **Tabela 8** foram utilizados para guiar a análise dos 26 comentários gerados pelos participantes da pesquisa de opinião. A análise levou à identificação de 54 trechos dos comentários ligados aos itens que operacionalizam as fontes de valor. A distribuição ficou da seguinte forma: 19 (35,2%) para Retenção, 12 (22,2%) tanto para Eficiência quanto para Inovação e 11 (20,4%) para Complementaridade.

Tabela 8. Itens operacionalizadores de fontes de valor.

EFICIÊNCIA	<i>Custo de Pesquisa:</i> os custos de pesquisa são diminuídos pelo suporte a várias plataformas/dispositivos do ecossistema. Assim como, pela definição clara de todos os recursos e funcionalidades dos produtos disponíveis no mercado do ecossistema.
	<i>Faixa de Seleção:</i> o profissional e DevRel ou o desenvolvedor oferece uma gama de produtos para atender às necessidades específicas de membros da comunidade.
	<i>Informação simétrica:</i> o profissional e DevRel ou o desenvolvedor fornece informações atualizadas e completas sobre produtos.
	<i>Simplicidade:</i> as transações dentro do ecossistema são simples do ponto de vista do desenvolvedor ou profissional de DevRel.
RETENÇÃO	<i>Programa de fidelidade:</i> o profissional de DevRel recompensa o uso contínuo ou aquisições de produtos no ecossistema.
	<i>Confiança:</i> o profissional de DevRel cria valor para o desenvolvedor, estabelecendo diretrizes de segurança e proteção.
	<i>Customização:</i> os produtos podem ser customizados pelo desenvolvedor.
	<i>Ponto de Contato:</i> a organização oferece um ponto de contato para o desenvolvedor se comunicar com o profissional de DevRel.
	<i>Comunidade virtual:</i> a organização cria uma comunidade virtual para agregar valor ao profissional de DevRel e ao desenvolvedor.
	<i>Efeito na rede:</i> o profissional de DevRel cria valor para a organização e para a comunidade de desenvolvedores aumentando ativamente o número de desenvolvedores.
COMPLEMENTARIDADE	<i>Entre produtos e serviços:</i> uma variedade de produtos e serviços são agrupados para criar valor e fornecer uma solução completa ao desenvolvedor em forma de serviço.
	<i>Entre recursos online e off-line:</i> o valor é criado por meio de uma combinação de recursos on-line (p.ex.: portal de desenvolvedores) e off-line (p.ex.: conferência de desenvolvedores).
	<i>Entre tecnologias:</i> o valor é criado combinando recursos de várias tecnologias (p.ex.: aplicação móvel com biblioteca de aprendizagem de máquina).
	<i>Entre atividades:</i> o desenvolvedor recebe acesso a produtos e serviços complementares ao produto principal ou serviço de interesse vendido no ecossistema.
INOVAÇÃO	<i>Novo conteúdo:</i> o desenvolvedor fornece à organização o valor através de um novo produto, serviço ou informação.
	<i>Novas funcionalidades:</i> o desenvolvedor fornece à organização valor através de novos recursos ou processos em produtos existentes, serviços de informação.
	<i>Reestruturação de processos e transações:</i> o desenvolvedor reestrutura um processo ou transação existente oferecido anteriormente no mercado, gerando valor para a organização.
	<i>Migração:</i> o desenvolvedor disponibiliza, pela primeira vez, para a comunidade um produto ou serviço que já existia, criando valor para o usuário.

3.6.4.1 Fonte de valor – Retenção

Como mencionado anteriormente, esta fonte de valor está relacionada à motivação do desenvolvedor em manter uma participação contínua gerando contribuições para a expansão da plataforma do ecossistema. Os itens identificados seguem a seguinte distribuição dentro dos 19 trechos de comentários: 5 (26,3%) para Ponto de Contato, 4 (21,1%) tanto para Programa de Fidelidade quanto para Confiança, 3 (15,8%) para Efeito na Rede, 2 (10,5%) para Comunidade Virtual e 1 (5,2%) para Customização.

Em relação ao **Ponto de contato**, os comentários indicam o bom relacionamento entre os profissionais de DevRel e o desenvolvedor. Isto facilita o entendimento das expectativas do desenvolvedor. A ação do profissional de DevRel deve aproximar mais o desenvolvedor do produto como forma de obter feedback. O feedback também leva a compreensão da probabilidade de recomendação para outro desenvolvedor. O ponto de contato ainda permite manter o desenvolvedor mais próximo da organização e produzindo o que ela, realmente, precisa.

“É um bom relacionamento entre o evangelista e o desenvolvedor.” – P13

“Envolver desenvolvedores durante eventos, oferecendo-lhes camisas e depois arrastando-as para um computador e dando-lhes uma demonstração, não para envolvê-las no produto, mas para obter seus comentários sobre o produto.” – P23

“E isso define como você precisa abordar seu público, como você precisa estar presente para seu público, mas também as expectativas que você pode definir com ele e os entendimentos que você tem quando está trabalhando com ele.” – P17

“É aqui que você pergunta a alguém que se inscreveu na probabilidade de recomendar seu serviço para outra pessoa.” – P25

“Mantê-los animados, dando-lhes valor através de novas oportunidades, e esse valor retornará para você.” – P30

Em relação ao **Programa de Fidelidade**, os comentários indicam como valor o reconhecimento do desenvolvedor de forma oficial pela organização como forma de impulsionar o engajamento do dele. O valor ainda está ligado aos relacionamentos duradouros dentro do ecossistema. Isto leva a uma comunidade de especialistas que além de gerar contribuições relevantes pode servir como apoio para os outros desenvolvedores. Os desenvolvedores precisam perceber que a organização os apoia dentro do ecossistema, o acesso a programas de acesso a versões beta de software, SDK e APIs avançadas, ferramentas de teste e análise de app, por exemplo, fornecem ao desenvolvedor o que é necessário para essa percepção.

“Reconhecimento público. Elogiar alguns desenvolvedores específicos em suas mídias pode ser um grande impulso para seu ego e seus negócios...” – P24

“... construção de relacionamentos de longo prazo com a comunidade, que é um aspecto essencial da economia de aprovação.” – P28

“Quando os desenvolvedores podem aumentar seu valor, seu alcance e seu reconhecimento, desenvolvendo sua plataforma e aumentando seu público-alvo, eles serão muito mais dedicados a você e sua marca...” – P30

“Apps especialmente bem-sucedidos ou inovadores merecem um tapinha nas costas, portanto, dê um e deixe que seus desenvolvedores saibam que você está prestando atenção.” – P7

Em relação ao item **Confiança**, os comentários indicam a credibilidade dos profissionais de DevRel como um valor relevante. Como resultado da confiança de empresas que compõem o ecossistema no trabalho dos profissionais de DevRel, é a interação entre os desenvolvedores mais experientes com os parceiros da indústria. A escala e tamanho das comunidades e, conseqüentemente, do ecossistema é um valor que ajuda na confiabilidade por outros desenvolvedores, pois um ecossistema só se mantém sustentável e em crescimento com atores e interações que permitam isto.

“...engajar-se com credibilidade, mesmo que isso signifique recomendar uma solução concorrente.” – P28

“Apresentações para os pares da indústria. Um e-mail sobre aquele desenvolvedor mais experiente – com informações de contato apropriadas compartilhadas – pode ajudar muito a criar bons sentimentos.” – P24

“... é sobre construir a confiança com os desenvolvedores.” – P3

“...Outro ponto é em escala e tamanho. Então, falar sobre o tamanho do nosso ecossistema é um ponto de confiabilidade...” – P19

Em relação ao item **Efeito de Rede**, os comentários estão voltados a ações que permitam o contato presencial e por meio da web com a comunidade de desenvolvedores, como seminários, eventos, demonstrações de ferramentas e webinários. O envolvimento de desenvolvedores em conjunto com os profissionais de DevRel em ações de contato com a comunidade é valioso. Isto impulsiona os próprios desenvolvedores a atuar como embaixadores do ecossistema e apoiaram a expansão das ações de DevRel, principalmente, para atrair mais desenvolvedores.

“Seminários, eventos falados, seminários on-line, demonstrações – seja o que for que você faça para chamar sua atenção para o mundo, convide alguns desenvolvedores para compartilhar os holofotes e ajudar a explicar as coisas enquanto se expõem...” – P24

“Eles querem ir lá e ensinar ao mundo o porquê é tão bom.” – P26

“... mais desenvolvedores entrarão no mercado” – P2

Em relação ao item **Comunidade Virtual** o comentário do participante 07 fortalece o uso da comunidade para favorecer o *“networking”* entre os desenvolvedores e outros atores do ecossistema. E isso fortalece o trabalho do profissional de DevRel, tornando a ação mais ampla, como sugerido pelo participante 26: *“E assim, eles se tornam parte de sua comunidade e tornam-se parte de suas relações de desenvolvedor mais ampla do que o que você pode alcançar.”*

Em relação ao item **Customização**, pelo conceito de um MSECO, sabe-se que um dos objetivos principais é expandir a plataforma, o que já inclui naturalmente essa customização. O participante 22 comentou que valor em torno disto é permitir por meio de infraestrutura que sejam criados novos produtos: *“ajudar os desenvolvedores a criar software para clientes que usam as APIs e garantir que eles possam ganhar a vida com esses projetos é muito satisfatório”*.

3.6.4.2 Fonte de valor – Eficiência

Os itens identificados seguem a seguinte distribuição dentro dos 12 trechos de comentários: 5 (41,7%) para Faixa de Seleção, 4 (33,3%) para Informação Simétrica, 2 (16,7%) para Simplicidade e 1 (8,3%) para Custo de Pesquisa.

Em relação ao item **Faixa de Seleção** os comentários indicam como valor os recursos desejados por desenvolvedores que podem gerar transações monetárias para o uso de produtos. Então, outro ponto é o conjunto de produtos que são destaque dentro do ecossistema que ajuda a promover a qualidade da plataforma tanto para desenvolvedores quanto para usuários. Para que os recursos desejados sejam oferecidos e produtos sejam promovidos, um aspecto que ajuda e é valoroso do ponto de vista dos participantes é a capacitação técnica dos desenvolvedores. A capacitação técnica pode ser promovida por meio de treinamentos planejados e executados pelos profissionais de DevRel.

“Quando alguém paga pelo seu produto, sua motivação é obter os recursos desejados para que eles paguem pelo produto. E o seu principal desafio é descobrir o que são.” – P17

“Promova seus aplicativos. Impulsionar os downloads de aplicativos gera muito valor em termos de relações com desenvolvedores e também ajuda a vender sua plataforma.” – P24

“Concentra em treinar desenvolvedores para aproveitar nossas APIs...” – P22

“Capacitação técnica.” – P5

Em relação ao item **Informação Simétrica** que aborda aspectos relacionados à área de DevRel como agente articulador do fluxo da informação sobre os produtos do ecossistema entre profissionais de DevRel, organização e desenvolvedores. Isto favorece a comunicação e confiança em torno das informações produzidas por desenvolvedores a partir das ações de DevRel. A conscientização sobre os produtos do ecossistema para todos possíveis interessados no uso e expansão. Para que essa expansão possa ser favorecida é necessário a produção e disponibilização de conteúdo com qualidade para o ecossistema.

“Há muitas peças móveis (engenharia, PM, marketing, serviços profissionais, desenvolvimento de negócios, etc.) e o DevRel é a graxa que mantém a máquina em movimento.” – P12

“...aumentando a conscientização sobre o seu produto.” – P28

“É um conteúdo de qualidade que pode ajudar um desenvolvedor a crescer... ajudar o desenvolvedor a aprender com mais eficiência, proporcionando o máximo de valor a ele.” – P22

“Forte comunicação e confiança” – P13

Em relação ao item **Simplicidade** os comentários indicam como valor a comunicação da visão e expectativas da organização além de clareza sobre a robustez do ecossistema.

“Na verdade, vender a visão e expectativas em torno da plataforma para clientes. Assim, as vendas são o primeiro ponto de contato para os clientes e precisam entender a visão da sua plataforma.” – P19

“...você precisa ter clareza sobre como voltar atrás caso ocorra algum problema.” – P16

Em relação ao item **Custo de Busca** o participante 13 comentou que valor é fazer o desenvolvedor perceber o quão vantajoso é contribuir para o ecossistema.

3.6.4.3 Fonte de valor – Inovação

Os itens identificados seguem a seguinte distribuição dentro dos 12 trechos de comentários: 5 (41,6%) para Reestruturação de Processos e Transações, 3 (25%) para Novos Conteúdos e 2 (16,7%) tanto para Novas Funcionalidades quanto para Migração.

Em relação ao item **Reestruturação de Processos e Transações** os comentários indicam o valor de DevRel como uma área essencial na estrutura

organizacional, que ajuda a organização a se concentrar em custos e também na maturidade da organização. A área de DevRel impacta diretamente a experiência do desenvolvedor (as expectativas e percepções geradas a partir do uso de produtos do ecossistema), gera receita por meio do uso de serviços e ajuda no reconhecimento da marca, por isso, deve ser inserida estrategicamente como parte dos negócios da organização.

“Você precisa se concentrar no custo.” – P16

“É completamente necessário para a equipe de desenvolvimento, para os gerentes de projeto, para as ações diárias e principalmente para a entrega do produto.” – P6

“...a maturidade da empresa, a maturidade do produto, a quantidade de adoção que você está ganhando na comunidade, isso vai mudar suas estratégias e moldar a forma como você pensa sobre as coisas.” – P27

“...são uma estratégia de entrada no mercado, a experiência do desenvolvedor é uma função do produto e deve ser medida como uma unidade de negócios.” – P29

“Gerar receita por meio do uso de seu serviço ou talvez esteja implícito porque você está criando reconhecimento de marca para alguns.” – P16

Em relação ao item **Novos Conteúdos** os comentários indicam a disponibilização de conteúdo gratuito aos desenvolvedores, as novas contribuições dos desenvolvedores para o ecossistema e os novos serviços para desenvolvedores que são gerados a partir do engajamento deles.

“O conteúdo pode ser fornecido aos desenvolvedores sem qualquer custo para eles.” – P22

“As entregas produzidas pelos desenvolvedores participantes de um programa.” – P10

“As pessoas estão agora consumindo mais do seu sistema operacional, o que está gerando mais serviços.” – P16

Em relação ao item **Novos Funcionalidades** os aspectos de valor são relacionados a contribuição para produtos já existentes, contribuições de extensões para produtos gratuitos e pagos.

“...entregas produzidas pelos desenvolvedores participantes de um programa.” – P10

“Alguém compra o seu produto e depois ele desenvolve...” – P21

Em relação ao item **Migração** o participante 15 comentou que é possuir desenvolvedores ativos no ecossistema. Já o participante 22 como descrito abaixo menciona as novas oportunidades que permitam o desenvolvedor avançar dentro do ecossistema.

“Ajudá-los a criar negócios que aproveitem as oportunidades que existem em nossa tecnologia” – Participante 22

3.6.4.4 Fonte de valor – Complementaridade

Os itens identificados seguem a seguinte distribuição dentro dos 11 trechos de comentários: 5 (45,5%) para Produtos e Serviços, 3 (27,3%) para Atividades, 2 (18,2%) para Tecnologias e 1 (9%) para Recursos *on-line* e *off-line*.

Em relação ao item **Produtos e Serviços** os comentários indicam o investimento financeiro dos desenvolvedores nos recursos oferecidos pelos profissionais de DevRel e pelos produtos/serviços da organização. As informações precisas sobre o roadmap da organização para que os desenvolvedores possam adequar suas necessidades.

O cenário apresentado no parágrafo anterior pode permitir entender aquilo que a organização espera da área de DevRel e da comunidade de desenvolvedores. Os produtos com código aberto também foram mencionados como valor. Outro aspecto abordado se refere aos dados gerados por uso de produtos e serviços para entender o que os desenvolvedores querem e como utilizam os recursos do ecossistema.

“É sobre o retorno de desenvolvedores e o investimento com mais dinheiro por parte deles.” – P21

“Estão entusiasmados por estarem associados a alguém que lhes fornece informações precisas sobre suas necessidades em relação ao lançamento de produtos de uma empresa.” – P28

“Dados sobre quão bem o nosso produto faz, o que as pessoas querem, desenvolvedores de código aberto. Você só quer que as pessoas usem as coisas para que você possa mostrar que tem valor.” – P27

“O passo mais importante para quem é profissional de DevRel é realmente entender o que sua empresa espera de você e o que é que você pode trazer para sua comunidade.” – P17

Em relação ao item **Atividades** os comentários indicam o equilíbrio para ações com tipos diversificados de comunidades de desenvolvedores. Outra percepção sobre

valor e atividades é a condução à contribuição para outros produtos do ecossistema. O profissional de DevRel ainda conecta desenvolvedores de maneira escalável aos produtos do ecossistema.

“Trata-se de equilibrar as necessidades de chegar aos desenvolvedores e de trabalhar com as comunidades de desenvolvedores livres e de código aberto.” – P17

“A ideia é que você crie usuários de algum outro produto por meio de uma estratégia de DevRel” – P21

“Ela conecta os desenvolvedores ao produto de maneira escalável.” – P11

Em relação ao item **Tecnologias** os comentários indicam a combinação de tecnologias do ecossistema nas contribuições dos desenvolvedores. Isto gera diferenciação para o mercado atraindo usuários e criando nichos dentro do ecossistema. Os desenvolvedores, desta forma, aumentam sua reputação pois criam novos e competitivos produtos em um ambiente altamente competitivo.

“Incentivar os desenvolvedores a usarem suas tecnologias para entregar usuários para você.” – P21

“Os desenvolvedores são notados por apostar no uso de novas tecnologias para criar novos produtos.” – P28

Em relação ao item **Recursos on-line e off-line** o participante 21 discorre que o valor de DevRel está na disponibilização de conteúdo acessível que permita a produção de novas contribuições: *“Ter certeza de que há conteúdo para as pessoas encontrarem. E por meio do fornecimento de APIs que permitam o desenvolvedor adicionar uma contribuição”.*

Com o resultado da análise dos comentários foi realizada uma expansão do framework de valor proposto por Amit e Zott (2001) como apresentado na **Figura 8**. Na Figura os itens operacionalizadores estão ordenados de forma decrescente considerando os aspectos identificados nos comentários dos profissionais de DevRel.

A extensão do framework com a percepção de valor em DevRel permitiu refinar o modelo DevGo, no que está relacionado a dimensão de negócios, e os objetos de transferência de valor. Em seguida, como forma de caracterizar a área de foco de Monitoramento do DevGo, analisou-se dentro da dimensão de negócios o valor a partir da perspectiva de motivação para uso e/ou contribuição para repositórios.

Os itens operacionalizadores identificados para cada fonte, como apresentado na **Figura 10**, foram aprimorados os objetos de transferência de valor do modelo DevGo.

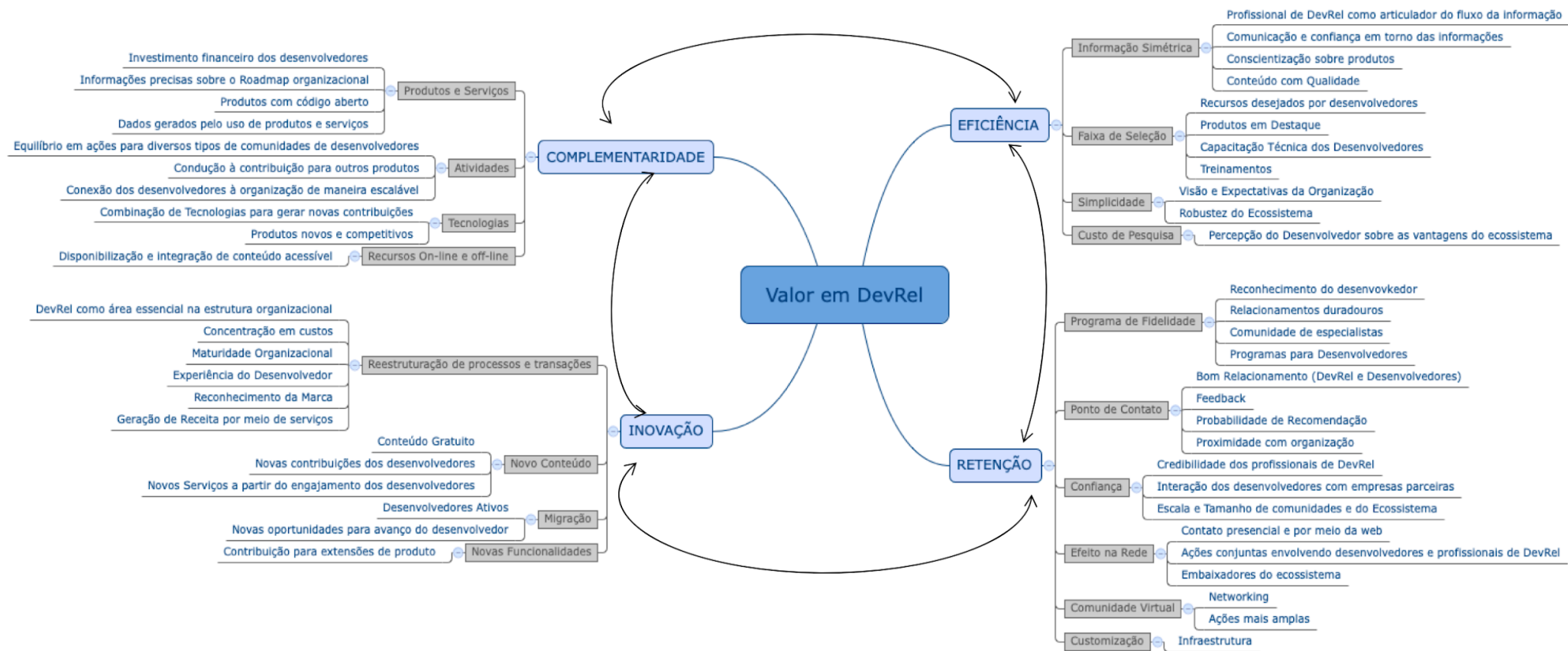


Figura 10. Fontes e itens operacionalizadores em DevRel expansão do framework de Amit e Zott (2001).

3.6.5 Análise dos Resultados dos Repositórios e Motivações

Na **Figura 11** são apresentados os gráficos de barras contendo, para cada tipo de conjunto de repositórios, a porcentagem das motivações indicadas pelos participantes. Todas as motivações foram ordenadas de forma decrescente (porcentagem de indicação pelos profissionais de DevRel).

Começando a análise pelos repositórios de **Perguntas & Respostas** as motivações com valor acima da mediana (36%) são: (1) Satisfação dos Desenvolvedores, (2) Crescimento da Comunidade, (3) Custo Operacional, (4) Competitividade do produto e (5) Custo de Manutenção.

Com isso é possível perceber que os profissionais de DevRel, participantes da pesquisa, usam este tipo de repositório para capturar o valor percebido dos produtos do ecossistema pelo desenvolvedor e com isso favorecer o crescimento de desenvolvedores para a expansão destes produtos e, conseqüentemente, da plataforma do ecossistema.

Os repositórios de Perguntas & Respostas guardam dúvidas técnicas e soluções em torno do desenvolvimento de contribuições que se utilizam dos recursos técnicos do ecossistema. Esses fatores ajudam a organização a analisar aquilo que reduz ou aumenta os custos tanto operacionais quanto de manutenção do ecossistema. Isto fortalece os insights que foram extraídas nos estudos de análise de questões técnicas no Capítulo 4.

Para os repositórios com característica de **Comunicação do Time** as motivações acima da mediana (31%) são: (1) Esforço de Desenvolvimento do Produto, (2) Estratégia de longo prazo, (3) Satisfação do desenvolvedor, (4) Facilitador de eficiência técnica, (5) Crescimento da Comunidade e (6) ROI.

Esse tipo de repositório consiste no registro de comunicação entre uma equipe de desenvolvedores e pode ser utilizado pelos profissionais de DevRel para uma comunicação mais aproximada com desenvolvedores do ecossistema. A primeira motivação está relacionada ao esforço para concluir o desenvolvimento de alguma contribuição para o ecossistema.

Dentro dessas contribuições há aquelas que terão impacto mais a longo-prazo como, por exemplo, desenvolvedores que estão inseridos em programas de acesso a SDKs que ainda não foram liberados pra comunidade. Isto ajuda na satisfação do desenvolvedor, por estar envolvido de forma mais aproximada com o ecossistema.

Logo, quando o produto for liberado para os outros desenvolvedores estes poderão perceber um maior valor do produto. Tudo o que foi mencionado está relacionado à quarta motivação indicada pelos participantes, pois permite a economia de custos, além da criação de novos nichos para contribuições de desenvolvedores. Isto favorece o crescimento da comunidade, assim como, sua especialização e o retorno sobre o investimento.

Quando se analisa os repositórios voltados a **Redes Sociais** as motivações acima da mediana (24%) são: (1) Crescimento da Comunidade, (2) Satisfação dos Desenvolvedores, (3) Estratégia de longo prazo, (4) Competitividade do produto, (5) Tempo para comercialização e (6) Custo operacional.

Na categoria de ferramentas de rede sociais se encaixam as motivações que envolvem a expansão de forma sustentável das comunidades de desenvolvedores dentro de um ecossistema. Essas interações permitem analisar aspectos positivos e negativos de produtos, assim como, agilizar o tempo para criação e evolução de produtos e contribuições.

Para os repositórios de aplicações móveis, dentro da categoria de **Lojas de Apps**, as motivações acima da mediana (23%) são: (1) Competitividade do produto, (2) Custo de manutenção, (3) Custo de entrega, (4) Valor financeiro, (5) Satisfação do desenvolvedor e (6) Esforço de desenvolvimento do produto.

As lojas de apps são onde os usuários podem consumir e avaliar as contribuições, neste caso as apps, dos desenvolvedores. Isto ajuda a compreender as motivações acima da mediana indicadas pelos profissionais de DevRel para o uso desta categoria. Um portfólio de apps variado, bem avaliado e que atende às necessidades dos usuários enquanto consumidor gera renda dentro do ecossistema. A satisfação do desenvolvedor e o esforço de desenvolvimento de produto, também, podem ser aspectos para a evolução das apps dentro do repositório.

Para os repositórios que envolvem informações geradas em **Listas de E-mail/Fóruns** as motivações acima da mediana (24%) são: (1) Competitividade do produto, (2) Esforço de desenvolvimento do produto, (3) Custo operacional, (4) Estratégia de longo prazo, (5) Custo de manutenção e (6) Crescimento da comunidade.

As listas de e-mail ou fóruns são espaços para perguntas, ideias, dicas e truques relacionados a um ecossistema. O uso de repositórios dentro deste conjunto é motivado pelo entendimento da competitividade dos produtos e recursos do ecossistema a partir da perspectiva dos desenvolvedores. Podem ser utilizados tanto para compreender o esforço para o desenvolvimento e publicação de novas contribuições quanto os custos operacionais.

Estes elementos próprios de repositórios desse conjunto apoiam na adequação de estratégias para longo prazo para manter o ecossistema vibrante. As informações ainda podem ser úteis para o entendimento do custo de manutenção de contribuições por membros da comunidade favorecendo assim o crescimento e especialização das comunidades de desenvolvedores dentro do ecossistema.

Nos repositórios que apoiam o **Ambiente Social de Codificação** as motivações que estão acima da mediana (23%) são as seguintes: (1) Valor Financeiro, (2) Crescimento da

Comunidade, (3) Estratégia de longo prazo, (4) Custo de Manutenção, (5) Satisfação do Desenvolvedor e (6) Esforço de Desenvolvimento do Produto.

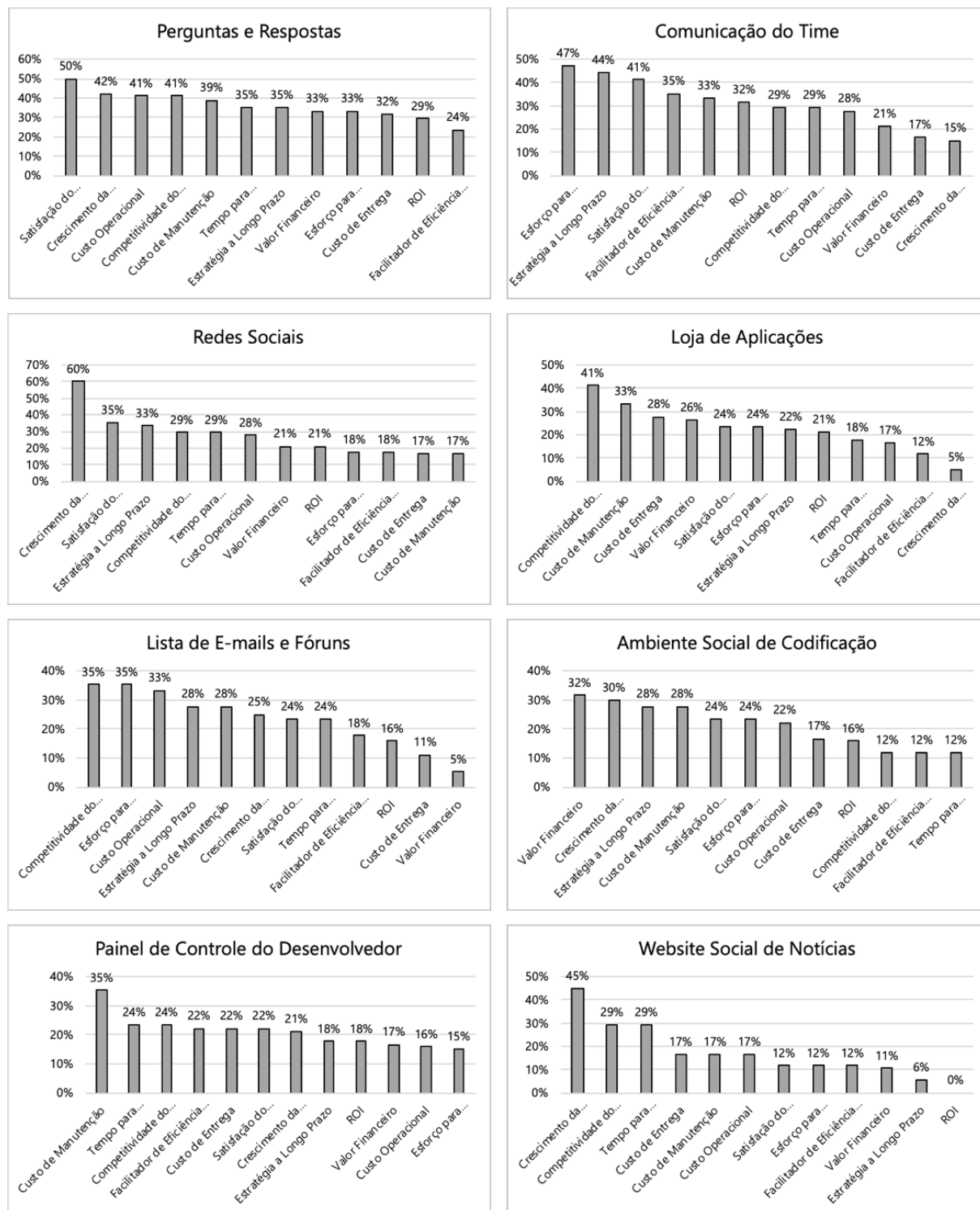


Figura 11. Repositórios e motivações.

Os repositórios que permitem ambientes sociais de codificação e compartilhamento de código têm associado a eles motivações voltadas ao valor comercial esperado dos produtos e a comunidade que cresce em torno do desenvolvimento, verificação e validação de código e projetos.

Grande parte das contribuições geradas nestes repositórios servem como insumos para estratégias de longo prazo, permitem a análise de manutenção de produtos. Permitem ao desenvolvedor perceber o valor de suas contribuições para outros desenvolvedores e para a organização detentora da plataforma do ecossistema. E reduzem o esforço da organização e dos profissionais de DevRel no desenvolvimento e entrega de novos produtos ou componentes reutilizáveis, por exemplo.

As motivações relacionadas aos repositórios dentro do conjunto de **Painel de Controle de Desenvolvedor** acima da mediana (22%) são: (1) Custo de Manutenção, (2) Tempo para comercialização e (3) Competitividade do produto.

Os repositórios relacionados a informações do Painel de Controle de Desenvolvedor contêm informações sobre as contribuições sejam elas, aprovadas, reprovadas, publicadas e o status de andamento para aprovação em uma loja de aplicações móveis. Os relatórios permitem uma análise do custo de manutenção e o momento para distribuição da contribuição a usuários. Os participantes também indicaram que estes repositórios ajudam na análise da competitividade dos produtos do ecossistema.

Para os repositórios relativos ao conjunto de **Website Social de Notícias** as motivações que direcionam à utilização e estão acima da mediana (14%) são: (1) Crescimento da Comunidade, (2) Competitividade do produto, (3) Tempo para comercialização, (4) Custo de entrega, (5) Custo de manutenção e (6) Custo operacional.

As motivações listadas no parágrafo anterior indicam que o uso de informações de websites como o Reddit e Hacker News estão ligadas ao entendimento do crescimento da comunidade tanto de usuários como desenvolvedores. Logo, essa motivação está ligada também àquela voltada a análise da competitividade dos produtos em torno da plataforma do ecossistema.

As informações ajudam a adequar estratégias para o lançamento de novos produtos e recursos técnicos para os variados tipos de desenvolvedores e usuários. As outras três motivações acima da mediana estão ligadas a custos: (a) de entrega dos produtos a essas comunidades; (b) manutenção a partir das informações extraídas nesses repositórios; (c) e, operacional, os custos envolvidos para a operacionalizar a comunidade.

3.6.6 Ameaças à Validade

As ameaças à validade foram organizadas em 4 categorias: validade de *constructo*, interna, externa, e de conclusão.

Validade de conclusão: realizada por meio de simples demonstração da ligação entre motivações e um conjunto de repositórios. E da análise de comentários dos participantes sobre a percepção de valor em DevRel.

Validade interna: propôs-se selecionar profissionais de DevRel que atuam/atuaram em MSECO. Deste modo, assumiu-se que eles são representativos para a população de profissionais de DevRel em MSECO e que possam dar a perspectiva da organização central sobre a governança de desenvolvedores.

Validade de constructo: as motivações foram extraídas de estudos primários sobre motivações de desenvolvedores em projetos de software e em ECOS de código aberto. As categorias dos repositórios foram classificadas em cima das características em comum. Isso também foi feito como base em outros trabalhos já publicados que abordam esses repositórios e no estudo disponível na Seção 3.5 sobre repositórios de MSECO. Os questionários passaram por avaliação em estudo piloto com quatro pesquisadores da área de engenharia de software e um profissional da área de DevRel.

Validade externa: os participantes do estudo em geral podem ser considerados representativos para a população de profissionais de DevRel. Para a avaliação do nível de envolvimento na governança de desenvolvedores no contexto de MSECO, os dados do questionário sobre a experiência dos participantes foram analisados.

Os materiais utilizados no estudo podem ser considerados representativos e atuais para o problema em análise. Tais materiais são formados pelas estratégias que estão relacionadas à governança de desenvolvedores em MSECO.

3.6.7 Conclusão da Pesquisa de Opinião

Com os resultados desta pesquisa de opinião foi possível refinar os objetos de transferência de valor do modelo DevGo, estes são descritos na Seção 5.1.3. A partir do framework foram extraídos os objetos de transferência de valor que são providos e consumidos por cada área de foco, como apresentado na **Figura 12**.

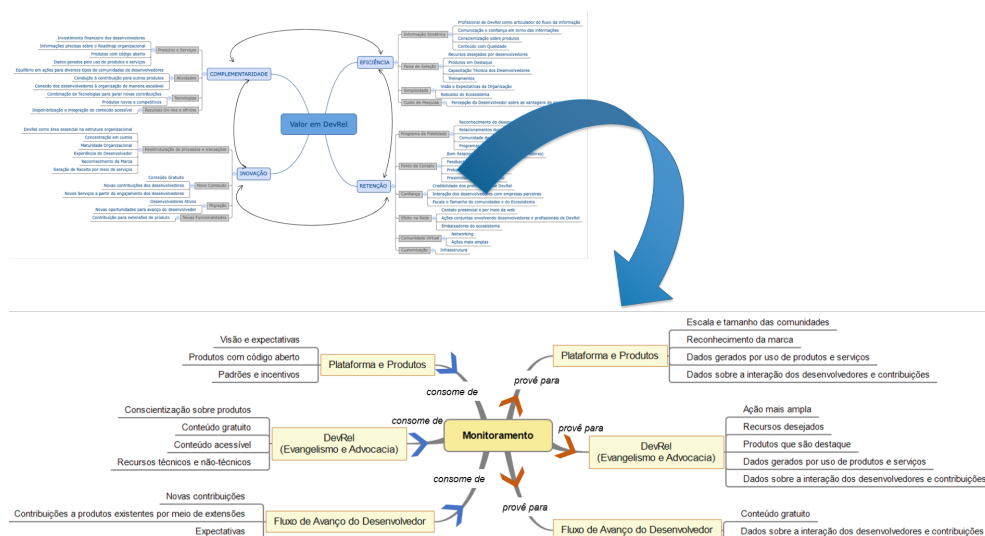


Figura 12. Quarta versão do modelo DevGo – objetos de transferência de valor.

Outro aspecto refinado foi a inserção de motivações para uso de categorias de repositórios que compõem a área de foco de monitoramento do DevGo.

3.7 Considerações do Capítulo

Foi possível avançar no estabelecimento de uma definição de governança do desenvolvedor: os resultados mostram que a governança do desenvolvedor depende da área de DevRel alinhada às lições aprendidas da organização central (por exemplo, roteiro da plataforma do ecossistema). O DevGo foi considerado pelos participantes como uma maneira clara de apoiar relacionamentos estruturados e orgânicos entre a organização e os desenvolvedores.

Com os estudos executados foi possível refinar os elementos estruturais do DevGo, um conjunto de lições aprendidas, categorias de repositórios pra monitoramento e os objetos de transferência de valor entre as áreas de foco do modelo.

Os estágios do DevGo provavelmente, como indicado pelos participantes, serão influenciados por diferentes fatores e taxas de dispersão do desenvolvedor e mortalidade. Mesmo em um estágio, os desenvolvedores podem se diferenciar em relação à “qualidade” ou “condição de participação”. Outra opinião capturada é que a área de foco “Monitoramento” dá suporte a insights precisos e ajuda a identificar sinais de alarme que indicam necessidades de reações.

Os aspectos sociais envolvidos no controle e monitoramento dos desenvolvedores também precisam da atenção dos pesquisadores. É necessário entender como monitorar o engajamento de desenvolvedores usando repositórios como fontes de informação (dados de mineração, mecanismos de integração e feedback de requisitos).

Esses estudos de concepção e refinamento do modelo, assim como de identificação de lições aprendidas baseadas em DevRel, representam uma perspectiva da organização central sobre a governança de desenvolvedores em MSECO. Para obter uma visão sinérgica sobre a governança se faz necessário obter a percepção do desenvolvedor. Isto é discutido no próximo capítulo.

CAPÍTULO 4 – CAPTURANDO AS EXPECTATIVAS DO DESENVOLVEDOR

Como forma de buscar a sinergia dentro da governança é necessário, além de entender os objetivos da organização central, capturar as expectativas e percepções dos desenvolvedores durante o engajamento em um ecossistema. Para isto, neste capítulo são discutidos resultados obtidos a partir da aplicação de técnicas de mineração de dados no repositório da categoria Perguntas e Respostas, Stack Overflow, com foco no Android, iOS e Windows Phone.

4.1 Definição do Estudo

Parte das contribuições tanto da organização central como dos desenvolvedores são armazenadas em um repositório oficial do MSECO, como *Android Developers*¹⁸ e *Apple Developers*¹⁹. Existem ainda ligações entre MSECO e repositórios externos, como de ambiente social de codificação (p.ex.:, GitHub) [Casalnuovo *et al.* 2015] e perguntas e respostas (Q&A, por exemplo, o *Stack Overflow*) [Lin e Serebrenik 2016].

Um repositório como o *Stack Overflow* tem comunicações arquivadas entre os desenvolvedores de ecossistema e pode ser usado para investigar alguns aspectos (p.ex.: engajamento do desenvolvedor) [Fontão *et al.* 2018d]. Isto pode ajudar a obter uma visão geral do ecossistema, fornecendo medições eficazes para analisar o desempenho do desenvolvedor [Eckhardt *et al.* 2014] e, como indicado nos estudos anteriores desta tese, ajudar no engajamento dos desenvolvedores como um instrumento para apoiar a governança deles dentro do MSECO.

Neste estudo foi utilizado o *Stack Overflow* (SO), da categoria de Perguntas e Respostas como indicado no modelo DevGo, para capturar as percepções dos desenvolvedores a partir do engajamento em um MSECO. Além disso, foram identificados insights para a área de foco de monitoramento do DevGo.

4.2 Minerando questões técnicas

4.2.1 Planejamento e projeto do estudo

As questões de pesquisa (QP) são baseadas no princípio “Representação” dentro da governança da comunidade [O’Mahony 2007]. O princípio da representação significa que os contribuidores, neste caso os desenvolvedores, podem ser representados por decisões ou perguntas da comunidade. A representação pode ser examinada pelo grau em que os

¹⁸ developer.android.com

¹⁹ developer.apple.com

membros podem exercer a voz na comunidade. As questões de pesquisa e planejamento experimental foram inspirados nos estudos de [Bajaj e Mesbah 2014] (foco: desenvolvedores da web), [Barua *et al.* 2014] (foco: testes) e [Rosen e Shihab 2016] (foco: desenvolvedores de aplicações móveis).

4.2.2 Questões de pesquisa

Como forma de apoiar a principal questão de pesquisa “O que pode ser entendido a partir dos três principais MSECOS (ou seja, Android, iOS e Windows Phone) com base em questões técnicas no Stack Overflow?”, foi definido um conjunto de subquestões de pesquisa.

As respostas a essas perguntas ajudam a investigar informações sobre como identificar e oferecer suporte aos mecanismos de governança do desenvolvedor no MSECO a partir do Stack Overflow. Essas percepções são importantes para auxiliar na governança de desenvolvedores, a partir do aspecto de monitoramento, com base na enorme quantidade de dados disponíveis. As subquestões de pesquisa são descritas usando a abordagem GQM (*Goal-Question-Metric*) [Basili 1994]:

OBJETIVO #1: Analisar como a atividade dos desenvolvedores está evoluindo em relação ao número de perguntas, ao número de respostas e ao tempo de resposta. A intensidade da atividade corresponde à frequência com a qual as perguntas e respostas são postadas, incluindo o tempo médio de resposta de cada tópico.

QP1. Quais são os tópicos relevantes extraídos de perguntas técnicas feitas por desenvolvedores de cada MSECO?

- **Razão:** obter uma visão geral de quais tópicos são abordados e se há alguma diferença entre os MSECOS analisados;
- **Métricas:** Número de agrupamentos; Nome dos tópicos.

QP2. O que pode ser entendido a partir dos três principais MSECOS baseado na relação entre questões e emoções?

- **Razão:** identificar e monitorar as emoções da comunidade de desenvolvedores, permitindo que a organização central antecipe e resolva possíveis ameaças em suas comunidades, além de descobrir e promover fatores que geram produtividade no MSECO;
- **Métricas:** quantidade de postagens referentes a cada emoção.

QP3. Quais são os tópicos relevantes extraídos a partir de questões técnicas sem resposta em cada MSECO?

- **Razão:** obter uma visão geral sobre os tópicos que cobrem as perguntas sem resposta e se há alguma diferença entre os MSECOs analisados;
- **Métricas:** Número de agrupamentos e Nome de tópicos.

OBJETIVO #2: Analisar o engajamento do desenvolvedor.

QP4. Quais são as perguntas em que mais os desenvolvedores se engajam?

- **Razão:** entender o quanto o envolvimento em determinados tópicos contribui para explorar o fluxo de conhecimento dentro de um MSECO. Assim, podemos identificar os desenvolvedores mais comprometidos com base nas perguntas mais comentadas/visualizadas;
- **Métricas:** número de respostas; Contagem de visualizações; Número de clusters; Número de tópicos.

QP5. Existe alguma relação entre perguntas e eventos oficiais?

- **Razão:** uso de séries temporais para identificar as questões frequentes do ecossistema, a fim de entender se os tópicos têm alguma relação com eventos oficiais, como o lançamento de plataformas. Como tal, é importante saber como analisar o efeito de eventos externos na comunidade;
- **Métrica:** Frequência de postagem dentro de 12 meses.

QP6. Qual é o ranking do número de medalhas recebidas pelos desenvolvedores em cada plataforma?

- **Razão:** obter informações sobre os emblemas dos desenvolvedores MSECO, bem como para explorar informações sobre os principais desenvolvedores;
- **Métrica:** Número de medalhas recebidas pelos desenvolvedores em cada MSECO.

4.3 Seleção dos dados

O SO disponibiliza seus dados publicamente no formato XML licenciado sob a licença CC BY-SA 3.0. Para este estudo foi utilizado o posts.xml, que contém o conteúdo do texto das postagens, as respostas/contagem de visualizações, tags, contagem de favoritos e data de criação. Como o objetivo era recuperar conjuntos de dados de três MSECOs, foram mapeadas as tags que poderiam representar os MSECOs do Android, iOS e Windows Phone. Essa análise nos permitiu adotar as tags: android, windows-phone e ios.

Um total de 1.568.377 postagens relacionadas a MSECO foram extraídas para compor o conjunto de dados de janeiro de 2008 a dezembro de 2017, sendo distribuído da

seguinte forma: Android 62,9% (986.099), iOS 34,2% (535.876) e Windows Phone 2,9% (46.402).

4.4 Execução do Estudo

Nas próximas subseções são descritos os procedimentos usados para detectar tópicos relevantes, selecionar o número de agrupamentos e filtrar o conjunto de dados para responder as QPs.

4.4.1 Detectando tópicos relevantes

Conforme os estudos anteriores baseados na mineração do SO, usamos uma abordagem não supervisionada para extrair tópicos de suas perguntas. O procedimento de mineração foi automatizado desde a construção do conjunto de dados até a análise dos tópicos. O framework Spark²⁰ foi utilizado por apoiar a análise de grandes volumes de dados (*Big Data*). Nossa metodologia é composta de 4 etapas:

- **Coleta de Dados:** descrito na Seção 4.3;
- **Pré-processamento:** eliminar termos não representativos (p.ex., palavras irrelevantes, URLs, *emoticons* e *hashtags*) na coleta e o processo de extração de recursos para esta etapa o corpo das postagens foi analisado. Primeiro, foram descartados qualquer fragmento de código (entre tags HTML `<code>`), porque a sintaxe do código fonte (por exemplo, instruções `if` e loops `"for"`) introduz ruído na fase de análise. Em seguida, foram removidas todas as tags HTML (por exemplo, `<p>` e `<a href="...">`). Terceiro, foram removidas as palavras comuns (*stopwords*) em inglês, como "a", "the" e "is". Neste contexto, foi utilizada a ferramenta NTLTK [Loper e Bird 2002] para eliminar os termos (frequência menor que 5) com a abordagem *bag-of-words* com o TF-IDF [Larson 2010]. Esta etapa foi aplicada em toda a base para a análise de todas as questões;
- **Extração de tópicos:** para esta etapa foi utilizado o modelo estatístico de tópicos, o LDA (*Latent Dirichlet Allocation*) [Krestel *et al.* 2009], pois ajuda a recuperar automaticamente tópicos em vários domínios de grandes volumes de documentos de texto. Além disso, foi aplicada uma técnica de particionamento chamada *Silhouette* [Aranganayagi *et al.* 2007], baseada em um número e análise de um gráfico, para identificar o número apropriado de tópicos (e não uma escolha aleatória) para o LDA. *Silhouette* fornece valores no intervalo de -1 a 1, em que 1 significa que as amostras pertencentes ao agrupamento estão distantes dos outros clusters, 0 significa que a divisão entre os agrupamentos já está na borda da

²⁰ <https://spark.apache.org/>

separação e -1 significa que algumas amostras têm uma chance de serem atribuídas ao cluster errado;

- **Resultados:** na Seção 4.5, para cada questão de pesquisa apresentada na Seção 4.2.2, os resultados serão analisados.

4.4.2 Filtrando a base de dados

Com o objetivo de responder QP1, QP2, QP4 e QP5, foi realizado um filtro do conjunto de dados original para extrair apenas o corpo de questões técnicas. Para criar o conjunto de dados específico para a análise QP3 (perguntas não respondidas), foram extraídas as perguntas com o valor da contagem de respostas igual a zero. Para o QP6, foi criado um ranking do número de medalhas recebidas pelos desenvolvedores dentro dos três MSECOs. Para isto, foram utilizadas as informações de conjuntos de dados sobre 9.795 desenvolvedores de cada MSECO classificados por reputação – número de medalhas conquistadas.

4.5 Análise dos resultados e discussões

No final de cada análise em alguns QPs (QP1, QP3, QP4, QP5 e QP6), são apresentados insights resultantes das respostas às QPs que podem orientar os pesquisadores e profissionais da indústria a usar dados para estudar MSECOs e que podem ser avaliados em estudos posteriores. Os insights foram criados em cima das motivações indicadas por profissionais de DevRel para o uso desta categoria de repositórios: satisfação dos desenvolvedores, crescimento da comunidade, custo operacional, competitividade do produto e custo de manutenção.

4.5.1 (QP1) Quais são os tópicos quentes extraídos de perguntas técnicas feitas por desenvolvedores de cada MSECO?

O número de tópicos e o valor de silhueta para cada MSECO foram: Android - $n = 4$ (0,64), iOS - $n = 4$ (0,86) e Windows Phone - $n = 3$ (0,87). Usamos o número de tópicos como entrada para o algoritmo LDA. A **Tabela 9** mostra os resultados.

No MSECO Android, as questões relacionadas ao tópico “Projeto” envolvem os projetos básicos do Android, como iniciar novos projetos, importar/exportar projetos e criar/manipular atividades. O tópico Interface do usuário aborda questões sobre posicionamento, alinhamento e justificação de objetos em relação a um elemento de contêiner. As perguntas relacionadas ao tópico “Exceções” abordam dúvidas relacionadas a uma condição que requer desvio do fluxo normal do programa Android. Por fim, o tópico “Notificações” aborda questões técnicas relacionadas a um elemento da interface do usuário

que um desenvolvedor pode exibir fora da interface da aplicação móvel para indicar que um evento ocorreu.

Tabela 9. Tópicos mais frequentes por MSECO.

Android	<i>Project</i>	<i>file, project, activity, class</i>
	<i>User Interface</i>	<i>activity, view, layout, xml</i>
	<i>Notifications</i>	<i>service, device, data, notification,</i>
	<i>Exceptions</i>	<i>androidruntime, lang, thread, method</i>
iOS	<i>User Interface</i>	<i>cell, table, tableview, uitableview</i>
	<i>Data Binding</i>	<i>view, data, controller, screen</i>
	<i>Project</i>	<i>xcode, project, device, iphone</i>
WP	<i>Services</i>	<i>wSDL, keys, resx, reminder</i>
	<i>Data Binding</i>	<i>page, data, xaml, control</i>
	<i>Frameworks</i>	<i>xna, dll, native, reflection</i>

Em relação ao MSECO iOS, o tópico “Ligação de dados” abrange os mecanismos usados para sincronizar uma interface do usuário com um modelo de dados. O tópico “Interface do usuário” abrange o controle da interface do usuário e a adaptação a qualquer alteração de tamanho. O tópico “Projeto” envolve perguntas semelhantes dentro da comunidade do Android, com foco na criação de aplicações para dispositivos de hardware.

As perguntas no MSECO do Windows Phone têm os seguintes tópicos: “Serviços” - que envolvem serviços da Web e de dados que usam uma linguagem baseada em XML aberta para descrever sua API baseada na Web; “Ligação de dados” - uma conexão/vinculação entre a interface do usuário e um objeto de dados permite o fluxo de dados entre essas camadas; e “Frameworks” - perguntas referentes a bibliotecas de links dinâmicos, suporte ao desenvolvimento de jogos, funcionalidades nativas do sistema, etc.

Há pontos comuns entre os tópicos de diferentes MSECOS: mecanismos de vinculação de dados, programação da interface do usuário (UI) e infraestrutura de suporte ao desenvolvimento. Isso nos leva ao seguinte *insight*:

Insight #1: Os tópicos podem indicar as barreiras mais frequentes enfrentadas por desenvolvedores dispostos a participar de um MSECO. Esse cenário pode servir como uma estratégia de monitoramento para apoiar a organização central no recrutamento e no treinamento de desenvolvedores. Do ponto de vista do custo operacional, de manutenção e a competitividade, os tópicos comuns: vinculação de dados, programação de UI e infraestrutura de desenvolvimento, ajudam na análise.

4.5.2 (QP2) O que pode ser entendido dos três principais MSECOS baseado na relação entre questões e emoções primárias?

Para classificar as emoções, cada corpo de questão foi submetido à API *AlchemyLanguage Emotion Analysis*²¹. *AlchemyLanguage* é uma coleção de APIs que oferece análise de texto através do processamento de linguagem natural e que conta com bases treinadas para a classificação de emoções primárias. Ele detecta raiva, desgosto, medo, alegria e tristeza implícitas no texto em inglês. A análise de emoções pode detectar emoções associadas a frases, entidades ou palavras-chave específicas, ou pode analisar o tom emocional geral do conteúdo.

A **Figura 13** mostra a porcentagem relacionada a cada emoção das questões analisadas. As três emoções mais presentes nos MSECOS analisados são: tristeza, raiva e alegria.

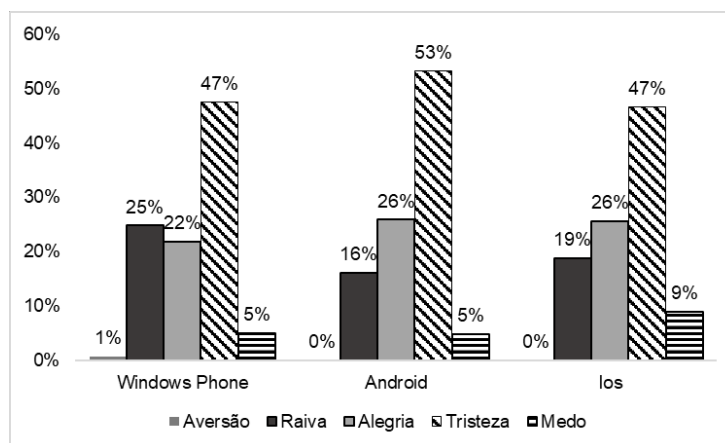


Figura 13. Emoções primárias identificadas em cada MSCO (Android, iOS e Windows Phone).

4.5.2.1 Investigação sobre a emoção tristeza

Como forma de auxiliar na análise, foram extraídas as cinco questões (**Tabela 10**) mais representativas que representam a tristeza da emoção. Essas perguntas estão relacionadas ao *AVD manager*, que não disponibiliza o dispositivo do desenvolvedor, o que causa uma dúvida sobre se alguma etapa básica foi esquecida. A segunda pergunta mais vista menciona a comunicação entre o dispositivo que está executando alguma aplicação móvel com o computador do desenvolvedor. A terceira questão aborda a recuperação do número de série de um dispositivo. As duas últimas questões tratam da manipulação de elementos da interface do usuário: *EditText* e *Status Bar*, especificamente, para valores de propriedade desses elementos.

No MSCO Windows Phone, as palavras mais frequentes são: *use*, *install*, *work*, *know*, *try* e *emulator*. As cinco questões mais visualizadas relacionadas à tristeza são apresentadas na **Tabela 11**. As duas primeiras perguntas mais visualizadas estão

²¹ <https://www.ibm.com/watson/developercloud/alchemy-language.html>

relacionadas ao uso de elementos da lista suspensa e componentes do tipo data e hora. Nestas questões, a emoção está relacionada a não encontrar o componente com o comportamento esperado ou não saber como implementá-lo.

Tabela 10. Android – tristeza – As cinco questões mais visualizadas.

The AVD manager in Android Studio doesn't show my device but "adb devices" does show it. Am I missing something obvious here?
I would like to pull the log file from a device to my PC. How can I do that?
I need to use a unique ID for an Android app and I thought the serial number for the device would be a good candidate. How do I retrieve the serial number of an Android device in my app?
I have 5 EditTexts in android for users to input. I would like to know if I could have a method for checking all the 5 EditTexts if they are null. Is there any way to do this??
What's the height of the status bar in Android? Is it always the same? From my measurements it seems that it's 25dp, but I'm not sure if it has the same height on all platforms. (I want to know this to properly implement a fade transition from an activity that doesn't have status bar to one that does)

A terceira pergunta está relacionada ao mecanismo de virtualização necessário do *Hyper-V* para o emulador do Windows Phone funcionar. O desenvolvedor realizou os procedimentos, mas não pode ativar o *Hyper-V*. A quarta questão aborda a indisponibilidade de uma biblioteca para o desenvolvimento de aplicações móveis que fazem uso da estrutura do Silverlight. A quinta questão descreve lançando exceções ao tentar acionar o evento de clique em um botão.

Tabela 11. Windows Phone – tristeza – cinco questões mais visualizadas.

I'm not able to find a <i>Drop Down</i> list on windows phone 7. Am I missing something?
How do I get the first day and the last day of the current year in c# for windows phone 7?
I don't see Hyper-V at "Windows Futures". I turned on virtualization in BIOS. I also check that my computer is a SLAT capable machine.
In C# one can use System.Version.Assembly to get the version of a running app. However, this doesn't appear to exist in Silverlight for Windows Phone. Is there an alternative?
I completed coding my application. But when I click start button, my app raised a exceptions.. :(I just make my class using default data types.... please help me

Em relação ao MSECO iOS, as palavras mais frequentes relacionadas à tristeza são: *Image*, *Nsstring*, *Device*, *Request* e *VC*. As cinco questões do iOS mais relacionadas à tristeza são apresentadas na **Tabela 12**.

Tabela 12. iOS – tristeza – cinco questões mais visualizadas.

Just got a notice that the provisioning profile for one of my apps is about to expire. Is there some way I can renew the existing one or must I recreate a new one?
As someone that's new to Objective-C can someone give me an overview of the retain, assign, copy and any others I'm missing, that follow the @property directive? What are they doing and why would I want to use one over another?
My application has a dark background, but in iOS 7 the status bar became transparent. So I can't see anything there, only the green battery indicator in the corner. How can I change the status bar text color to white like it is on the home screen?
I'm customizing a Tableview. I want to hide the line separating on the last cell ... can I do this?
I saw that this is a property of the UITableView but I would like work only on a particular cell.
So my application is very basic but I'm new, new to the Xcode platform. The application basically calculates acceleration by inputting initial and final velocity and time and then use a formula to calculate acceleration. However, since the values in the text boxes are string, I am unable to convert them to integers

A primeira pergunta está relacionada ao uso do serviço da plataforma iOS, o perfil de provisionamento é o processo de preparar e configurar uma aplicação móvel para iniciar em dispositivos e usar serviços. A segunda questão aborda diretivas específicas da

linguagem Objective-C que é padrão de plataforma. A terceira questão aborda o aspecto da interoperabilidade, pois os elementos de interface programados devem estar relacionados ao sistema. Ainda na área de interface do usuário, a quarta questão aborda a manipulação de um elemento de organização de layout. Finalmente, a quinta questão explora os passos básicos da codificação de operações de conversão de valor.

Em relação à **emoção da tristeza**, as questões em geral estão relacionadas ao uso de emuladores e dispositivos, e manipulação de elementos da interface do usuário. Ao analisar a tristeza, no MSECO Android as palavras mais frequentes são: *error*, *found*, *constructor*, *resource* e *match*.

4.5.2.2 Investigação sobre a emoção raiva

Ao analisar a raiva, no MSECO Android as palavras mais frequentes são: *use*, *image*, *wish*, *method* e *card*. As cinco perguntas Android mais vistas relacionadas à **emoção da raiva** são apresentadas na **Tabela 13**.

A primeira questão está relacionada à implementação de um método que manipula a recuperação de estado da aplicação móvel quando um evento do sistema operacional é acionado. Neste ponto, o desenvolvedor não pode implementar o comportamento de recuperação de estado. Na segunda questão, o desenvolvedor pode baixar uma imagem, mostrar na tela, no entanto, ele precisa encontrar uma maneira de salvar a imagem em uma pasta específica. O desenvolvedor faz uso de uma biblioteca que permite salvar em um cartão SD, mas não pode personalizar a pasta.

Tabela 13. Android - raiva - cinco questões mais visualizadas.

I am launching an activity to make a phone call, but when I pressed the 'end call' button, it does not go back to my activity. Can you please tell me how can I launch a call activity which comes back to me when 'End call' button is pressed?
My issue is that I can download the image, display it on screen as a Bitmap. The only way I have been able to find to save an image to a particular folder is to use FileOutputStream, but that requires a byte array. Which works fine to save to SD card, but does not allow you to customize the folder.
Where can I find step-by-step instructions on how to parse a JSON feed in Android? I'm just an Android beginner wanting to learn.
I found a sample code, but it seems that the classes and string constants used in them are outdated and are no longer provided.
I am developing an app using Eclipse IDE Juno and Android SDK. How do I change my app's icon?

A terceira pergunta envolve um desenvolvedor iniciante que deseja manipular arquivos JSON para extração de dados. Este desenvolvedor precisa de um guia passo a passo. Na quarta questão, o desenvolvedor pode encontrar um exemplo de código disponível nos canais oficiais do Google para resolver o problema, mas alguns elementos do código estão desatualizados ou não são mais fornecidos. A quinta questão indica a troca de informações relativas à aplicação móvel, neste caso o ícone. O desenvolvedor, mesmo usando o ambiente de desenvolvimento compatível e o Android SDK, não pode executar uma tarefa simples, que é a troca de ícones.

No MSECO Windows Phone as palavras mais frequentes são: *file*, *error*, *value*, *construtor*, *element* e *button*. As cinco questões mais vistas relacionadas à emoção da raiva são apresentadas na **Tabela 14**. A primeira pergunta mais vista está relacionada ao problema de compatibilidade da plataforma. O desenvolvedor comprou um laptop apenas para o desenvolvimento de aplicações móveis, porém a versão do Windows não é Pro, o que não permitiria a execução das ferramentas necessárias. A segunda pergunta aborda a indisponibilidade do primeiro SDK do Windows Phone (7.0) para integração com outros serviços, a documentação em si não suporta a recomendação sobre como realizar a integração. A terceira questão se refere ao uso do conhecimento em uma tecnologia para iniciar na plataforma Windows Phone; no entanto, o desenvolvedor está totalmente perdido, porque a documentação menciona poucas informações sobre isso.

Na quarta questão, um desenvolvedor iniciante tem problemas com *namespaces* ao usar os padrões de projeto para o Windows Phone, já que a indicação é apenas para adicionar a biblioteca; no entanto, o desenvolvedor não pode resolver o problema. A quinta questão é relacionada a compatibilidade do SDK com uma versão do sistema operacional Windows que não suporta o desenvolvimento de aplicações móveis, o que força o desenvolvedor a atualizar a versão do sistema operacional.

Tabela 14. Windows Phone - raiva - cinco questões mais visualizadas.

I recently bought a Windows 8 laptop just for the sake of developing Windows Phone Apps. Mine is not a Pro version hence Hyper-V does not run. (I realized that Hyper-V needs WinPro after I bought the laptop). I am not capable of upgrading to pro edition. Can anyone please suggest me any alternatives to Hyper-V where my emulator can run?
I am trying to post to google so I can log into Google Reader and download subscription list, but I am unable to find a way to post to google in the windows 7 phone sdk, does anyone have an example on how to do this?
I was totally lost at sea. I once read a line or two on Microsoft website that says you can build WP8 apps using just HTML5, CSS3 and JavaScript but could never find it any more. So my question is if it's possible? I tried to find something to get started but it seems to me it all requires C#, C/C++, etc.? I thought they said the web is native in WP8? So in this sense, I should be able to develop apps using just HTML5, CSS3, and JavaScript? Am I right?
I'm using Mvvm light expression, and it does not support that namespace or at least I can't add it. I'm a newbie in using VS and c# atm, maybe I'm doing something wrong, but simply when I'm trying to add the reference to my project it does not list.
I downloaded the Windows Phone 7 SDK yesterday but when I installed it on my Windows XP it complaint that it needs to be installed on Windows 7. My question is, is it possible to install the Windows Phone 7 SDK or develop for Windows Phone 7 on WinXP? Is there any workaround that has been made to overcome this?

Em relação ao MSECO iOS, as palavras mais frequentes relacionadas à raiva são: *use*, *data*, *access*, *localization* e *visualization*. As cinco questões mais vistas relacionadas à emoção da raiva são apresentadas na **Tabela 15**. A primeira pergunta está relacionada ao processo de como realizar um teste beta com os usuários. O ponto aqui é a necessidade do desenvolvedor de um processo que não seja caro. Esta informação está presente na documentação oficial, mas não foi encontrada pelo desenvolvedor.

Na segunda questão, um desenvolvedor novato na plataforma deseja implementar solicitações HTTP. O desenvolvedor tem o conhecimento técnico, mas não pode aplicá-lo de maneira prática. A terceira questão envolve a conversão de dados que são obtidos de um servidor e serão consumidos por uma aplicação no iPhone. A quarta pergunta aborda os

serviços de notificação do usuário em uma aplicação da Web, no entanto, o serviço está disponível apenas para aplicações nativos ou híbridos (da Web e nativos). A quinta questão diz respeito à criação de localização e diretório no sistema de arquivos do iOS.

Tabela 15. iOS - raiva - cinco questões mais visualizadas.

How can you beta test an iPhone app? I can get it on my own device, and anyone that gives me a device, I can run it on theirs, but is there a way to do a limited release via the app store for beta testing
I'm approaching iOS development and I'd like to have one of my first applications to perform a HTTP POST request as far as I can understand, I should manage the connection which handles the request via a <code>NSURLConnection</code> object, which forces me to have a delegate object, which in turn will handle data events. Could someone please clarify the task with a practical example? I should contact an https endpoint sending authentication data (username and password) and getting back a plain text response.
I have UTF-8 encoded NSData from windows server and I want to convert it to NSString for iPhone. Since data contains characters (like a degree symbol) which have different values on both platforms, how do I convert data to string?
I'm working on a web app. How can I send push notifications to iOS users when there is new content?
Can someone explain to me what the documents directory is on an iOS app and when to use it? Here is what I believe at present: To me it seems to be a central folder where the user can store any files needed for the app. This would be a different location than where core data stores its data? It seems like each app gets its own documents directory. I am free to create a subdirectory of the documents directory, like documents directory/images, or documents directory/videos? If my app wants to write to the filesystem

Em relação à **emoção da raiva**, os pontos em comum entre os três ecossistemas envolvem a indisponibilidade ou incompatibilidade de recursos ao usar bibliotecas ou customizar métodos.

4.5.2.3 Investigação sobre a emoção alegria

Um ponto que chamou a atenção foi a classificação de parte do corpo de perguntas como **alegria**, uma vez que as questões técnicas inicialmente se referem a uma emoção envolvendo tristeza ou raiva. Com isso, seguimos os mesmos procedimentos da análise anterior para investigar o motivo do aparecimento dessa emoção.

Ao analisar a **alegria**, no Android MSECO as palavras mais frequentes são: *layout*, *content*, *eclipse*, *studio* e *background*. As cinco questões Android mais vistas relacionadas à emoção da alegria são apresentadas na **Tabela 16**.

Tabela 16. Android - alegria - cinco questões mais visualizadas.

To set Background: RelativeLayout layout =(RelativeLayout)findViewById(R.id.background); layout.setBackgroundResource(R.drawable.ready);Is the best way to do it?
I want to be able to change the background color to white in my android app in the simplest way possible.
How can I pass an object of a custom type from one Activity to another using the putExtra() method of the class Intent?
I finally managed to obfuscate my Android application, now I want to test it by installing the APK file and running it on the emulator. How can I install an APK file on the Android Emulator?
What does the following exception mean; how can I fix it?

Na primeira pergunta, o desenvolvedor sabe como implementar a troca de plano de fundo e quer encontrar a melhor maneira de realizar essa tarefa. Percebe-se que não há foco no problema de alterar a imagem de fundo, mas em como implementá-lo. A segunda pergunta segue a mesma linha de raciocínio, mas o desenvolvedor se concentra em

encontrar a maneira mais simples de implementar a mudança de cor. Essas duas perguntas estão relacionadas aos comportamentos da interface do usuário.

Na terceira questão, o desenvolvedor aborda métodos para realizar a passagem de parâmetros especificamente um objeto entre atividades. Pode-se analisar que os passos básicos para passar uma variável primitiva fazem parte do conhecimento do desenvolvedor. E agora ele quer avançar refinando a implementação para passar um objeto que contém mais informações para outra tela. A quarta questão aborda um desenvolvedor que já se preocupa em ofuscar o código para fornecer segurança contra engenharia reversa. A partir disso, ele precisa saber como fazer com que o emulador carregue o APK para realizar atividades de teste. Esses procedimentos estão relacionados à preocupação com a qualidade do produto. A quinta questão envolve lidar com exceções que podem ocorrer durante a execução da aplicação móvel. Aqui, o desenvolvedor procura maneiras de resolver a exceção.

No Windows Phone MSECO, as palavras mais frequentes são: *answer*, *page*, *develop*, *button* e *list*. As cinco perguntas mais visualizadas do Windows relacionadas à emoção da alegria são apresentadas na **Tabela 17**.

Tabela 17. Windows Phone - alegria - Cinco questões mais visualizadas.

I'm looking for the best practice on how to pass data from page to page. I want to pass that data back to Page A. I've seen suggestions to: build XML documents and save to Isolated Storage; use the App class to store information in properties; pass it like a query string. I'm looking for the Best practice. Is there one that Microsoft recommends or one that is generally accepted as the best way? Thanks!
Recently we started to work on a new project which includes clients for Windows 8 Metro, Windows Phone and Desktop application. It was decided to use MVVM pattern as main Architecture because sharing ViewModels between projects is much more acceptable solution for us. Maybe some of u already done this and knows the way we could achieve that functionality? please provide us right way to develop this task as using SQLite and having reusable code is very important on this stage of the development
I wanted to ask how it's possible to achieve something like this (first wide tile). I know that there are only a limited number of live tile templates and this is the list I found on Microsoft's website. There isn't any template that looks like what I'm looking for, so is there any other way to create a custom tile specifying text size and position? Thank you!
I have the following code: The response = await part just continues an ongoing loop and nothing happens. Any ideas what I am doing wrong? The question really is, how do I send the following header: Authorization: OAuth2 ACCESS_TOKEN to an external web API.
I'm trying to set timeout to HttpClient object in Windows Phone App. But if request is not completed before timeout, GetAsync never returns a value. How can I properly set the timeout to get result from GetAsync?

Na primeira pergunta, assim como no MSECO Android, o desenvolvedor procura as melhores práticas para implementar um método. Nesse caso, o método envolve a passagem de dados entre as páginas (telas) da aplicação móvel. Ele até sugere uma prática e pergunta se é algo indicado pela Microsoft. A segunda pergunta, que envolve decisões de arquitetura e reuso de código, é de um desenvolvedor que está desenvolvendo um novo projeto para Windows Phone, Windows 8 e Desktop, algo conhecido como multiplataforma.

A terceira questão envolve o desenvolvimento da funcionalidade “Tile” que funciona como um ícone na tela principal do Windows Phone que contém informações de notificação e permite acesso ao conteúdo dentro da aplicação móvel. Para que um desenvolvedor

implemente um “tile”, ele já deve ter desenvolvido os recursos da aplicação móvel que serão usados para preencher o bloco. Ainda neste caso, ele procura algum modelo que o ajude a implementar da maneira que ele quer.

A quarta questão envolve mecanismos de autenticação para o uso de uma API externa. O desenvolvedor precisa saber por que ele não obtém respostas quando solicita com o *token* de acesso. A quinta questão também está relacionada a solicitações de serviços da web. Nesse caso, o desenvolvedor deseja modificar a implementação para usar um mecanismo de tempo limite, isso indica que o desenvolvedor deseja tratar um comportamento específico do aplicativo.

Em relação ao MSECO iOS, as palavras mais frequentes relacionadas à alegria são: *Xcode*, *UIWebView*, *Build*, *Develop* e *SDK*. As cinco perguntas do iOS mais relacionadas à emoção da alegria são apresentadas na **Tabela 18**.

Tabela 18. iOS - alegria - cinco questões mais visualizadas.

I want to be able to create a GUID/UUID on the iPhone and iPad. The intention is to be able to create keys for distributed data that are all unique. Is there a way to do this with the iOS SDK?
I started taking a look of the new Swift on Xcode 6, and I tried some demo projects and tutorials. Now I am stuck at: Instantiating and then presenting a viewController from a specific storyboard. How to achieve this on Swift?
I have a very simple UIWebView with content from my application bundle. I would like any links in the web view to open in Safari instead of in the web view. Is this possible?
I'm trying to explore the possibilities with running a native web-app on different platforms using HTML5. Currently, an input type="date" field just opens the standard soft keyboard on Android and iOS. I suppose that in the future the mobile OS's soft keyboards will include date pickers and such - just as select invokes the native select today.
I have really great wish to set my own color to UITextField border. But so far I could find out how to change the border line style only. I've used background property to set background color in such way: self.textField.backgroundColor = textFieldColor; But I have to change color of UITextField border too. And my question was about how to change the border color.

A primeira questão envolve a implementação de identificadores que permitem o rastreamento de dados do usuário, permitindo a identificação de dispositivos para fórmulas de análise, segmentação e crescimento. A segunda questão envolve a linguagem de programação, especificamente mudanças de sintaxe na implementação de controladores. O desenvolvedor já tem experiência e usa como exemplo um trecho de código no Objective-C.

A terceira questão, relacionada à interface do usuário, tem a ver com a modificação do comportamento de um elemento de visualização da página da web. A quarta questão ainda sobre a interface do usuário busca estratégias para a interoperabilidade de aplicações móveis híbridas. Na quinta questão, o desenvolvedor precisa de ajuda para manipular propriedade específica de um elemento de interface com o usuário.

Com relação à **emoção da alegria**, identificou-se, a partir da análise, que os desenvolvedores procuram as melhores práticas e estratégias para manipular os comportamentos da interface do usuário, tomar decisões arquiteturais e integrar o aplicativo móvel aos serviços da web. Eles ainda estão procurando maneiras de implementar recursos avançados, como notificações e personalização de blocos.

A partir da análise das três emoções mais frequentes foi possível chegar ao seguinte *insight*:

Insight #2: As emoções primárias além de serem úteis pra obter a satisfação do desenvolvedor, também cobrem, aspectos de custo, competitividade e manutenção de produtos. A tristeza se relaciona a produtos como os emuladores e dispositivos, assim como, a manipulação de elementos de UI. A raiva está associada a indisponibilidade de recursos. E a alegria, pode impactar na competitividade do produto, pois há relação com as melhores práticas para gerar uma contribuição e a implementação de recursos avançados.

4.5.3 (QP3) Quais são os tópicos relevantes extraídos a partir de questões técnicas sem resposta em cada MSECO?

Com relação às perguntas do desenvolvedor não respondidas e mais visualizadas no, o número de tópicos e o valor da silhueta para cada MSECO são respectivamente: Android - n = 2 (0,45), iOS - n = 2 (0,58) e Windows Phone - n = 3 (0,47). Usamos o número de tópicos como entrada para o algoritmo LDA. Os resultados são mostrados na **Tabela 19**. Ao responder a essa pergunta é possível explorar as questões mais frequentes que a comunidade aponta, mas que, ao mesmo tempo, não detém o conhecimento. Por outro lado, as perguntas podem ser muito óbvias para a comunidade, ou podem já ter respostas e não foram moderadas.

No MSECO Android, as perguntas estão relacionadas a: *Deployment Issue* - problemas detectados no código que influenciam a instalação, a depuração e o teste de aplicativos em um dispositivo; e *Menu oculto* - envolve a ativação do menu com comandos de fábrica que fornecem uma grande quantidade de informações sobre o hardware e o sistema do dispositivo, já que um usuário comum pode ter dificuldade de acesso.

Tabela 19. Tópicos quentes para questões não respondidas.

Android	Deployment Issues	Code, using, file, problem, android, device, error, one, help, image
	Hidden Menu	Android, java, rtn, sensor, 11, 12, 22, 05, com, float
iOS	Facebook Login Error	Facebook, user, using, safari, device, login, notification, keyboard, code, server
	Video	View, code, using, file, error, problem, set, controller, works, video
WP	Analytics Integration	List, item, column, listpicker, pull, textblock, database, name, localytics, radius
	Libraries Issues	System, dll, exception, Microsoft, windows, speech, error, threading, mscorlib, sk
	Design Tool	Expression, remoting, designhost, isolation, tribe ,vimeo, score, verlightplatform, Microsoft, canceltoken

No MSECO iOS, o primeiro tópico (*Facebook Login Error*) se refere à dificuldade de fazer login no Facebook dentro do navegador Safari e ao feedback de informações na aplicação móvel. Isto está relacionado aos mecanismos de autenticação do Facebook para

dispositivos móveis. Outro tópico abrange problemas com a manipulação de componentes de vídeo.

No Windows Phone MSECO, o tópico *Analytics Integration* refere-se ao uso de SDKs analíticos nos componentes da aplicação móvel como uma forma de mapear o comportamento do usuário de maneira mais detalhada. O tópico *Libraries Issues* abrange exceções e outros erros no uso de bibliotecas; sua interação com o Windows Phone pode ser difícil devido à troca frequente de versões do sistema operacional, causando vários problemas de incompatibilidade de biblioteca. O tópico *Design Tool* está relacionado a problemas que ocorreram durante o uso de uma ferramenta para projetar e desenvolver uma aplicação móvel.

A partir da análise desses MSECO, é possível observar que mesmo uma moderação dos problemas não evita problemas que são frequentemente informados durante todo o desenvolvimento de uma contribuição para um MSECO. A análise para este ponto levou em consideração os problemas mais visualizados para que os tópicos possam estar relacionados a problemas enfrentados regularmente pelos desenvolvedores. Os tópicos da MSECO compartilham como característica comum o fato de serem termos relacionados àqueles que já avançaram no desenvolvimento: implantação, análise, autenticação de serviço, bibliotecas e projeto de uma aplicação móvel.

Analisar as questões de pesquisa anteriores mostra que parece haver pouca intersecção de interesses. Isto pode ser uma evidência de que as MSECOs são diferentes em termos de "interesses, desafios, dificuldades". Isso pode indicar que as estratégias realmente variam muito, o que se refere à necessidade de pesquisar as semelhanças como forma de estabelecer um modelo geral de interações entre desenvolvedores e sua governança dentro de MSECOs. Este cenário nos leva ao seguinte *insight*:

Insight #3: explorar como a falta de interseção de interesses impulsiona a troca de informações entre desenvolvedores trabalhando em mais de um MSECO. A pequena interseção mostra que as estratégias tomadas em nível gerencial podem afetar as interações e motivações dos desenvolvedores.

4.5.4 (QP4) Quais são as perguntas em que mais os desenvolvedores se engajam?

Para analisar esta questão, foram utilizadas duas perspectivas de engajamento: 1) pelo número de respostas dos desenvolvedores que participam de um dos MSECO e do SO; e 2) pelo número de visualizações de perguntas, ou seja, desenvolvedores que não participam do SO, mas visualizam perguntas.

Para cada caso (*NumAnswers* e *ViewCount*, respectivamente), como nos procedimentos de resposta às perguntas anteriores, o método LDA foi utilizado junto com

algoritmo de silhueta. Quanto ao engajamento do desenvolvedor e as perguntas mais populares respondidas, o número de tópicos e o valor da silhueta foram, respectivamente, para cada MSECO: Android - $n = 4$ (0,64), iOS - $n = 4$ (0,86) e Windows Phone - $n = 3$ (0,87). Os resultados são mostrados na **Tabela 20**.

No MSECO Android, os tópicos mais frequentemente respondidos estavam relacionados a: *Ligação de dados* - para escrever layouts declarativos e minimizar o código de ligação necessário para vincular a lógica e os layouts da aplicação móvel; *IDE* - o uso de edição de código, depuração e ferramentas de desempenho; *Interface do usuário* - para criar uma interface de usuário dinâmica e com vários painéis para encapsular componentes de interface do usuário e comportamentos de atividade em módulos de atividades; e *infra-estrutura de back-end* - o uso de uma plataforma que ajuda a aumentar a base de usuários e monetizar o aplicativo.

No MSECO Windows Phone, os tópicos estavam relacionados a: *Event Handler* - o uso de manipular métodos de manipulação de eventos para processar entrada de toque; *IDE* - ferramentas para suportar o desenvolvimento de aplicativos, incluindo emuladores e uso de ferramentas de migração (WP7.X para WP8.X); e *diretrizes de controle* de elementos da interface do usuário.

No MSECO iOS, alguns tópicos eram os mesmos dos outros. Tópicos adicionais foram: *Interface do usuário*; *Ligação de dados*; *Notification Services* - notificações locais e por *push* para manter os usuários informados com conteúdo relevante, se a aplicação móvel está sendo executada em segundo plano ou inativo; e *Linguagem de Programação* - o uso de Swift e Objective-C no XCode para desenvolver aplicações móveis.

Tabela 20. Tópicos quentes - as questões mais respondidas.

MSECO	Topic	Words
Android	Data Binding	Android, Text, Java, API
	IDE	Studio, File, Adb, Device
	User Interface	Activity, Fragment, View, Image
	Back-end Infrastructure	Firebase, Data, Notification, Time
WP	Event Handler	Event, Develop, Silverlight, Visual
	IDE	WP7, file, service, emulator
	User Interface	Listbox, control, page, image
iOS	User Interface	Swift, View, Change, Image
	Data Binding	Data, View, Control, Swift
	Notification services	Can, Call, Work, Notification
	Programming Language	Swift, Object, Xcode, Value

Em relação ao engajamento das questões mais visualizadas no SO, o número de tópicos e o valor da silhueta para cada MSECO são: Android - $n = 4$ (0,62), iOS - $n = 3$ (0,81) e Windows Phone - $n = 4$ (0,75). Os resultados são mostrados na **Tabela 21**.

No MSECO Windows Phone, os desenvolvedores trabalham para atualizar suas aplicações móveis para a plataforma mais recente, como forma de oferecer suporte a novos recursos (por exemplo, dados do sensor). Um desenvolvedor do Android deve decidir se deseja criar uma única aplicação móvel ou várias versões para ser executada nos mais diversos dispositivos da plataforma por meio do uso de *Fragments*. No MSECO iOS, as perguntas mais visualizadas estão relacionadas ao uso do ambiente de desenvolvimento integrado, programação e projeto/desenvolvimento da interface do usuário.

Tabela 21. Tópicos quentes - questões mais visualizadas.

MSECO	Topic	Words
Android	IDE	Device, android, java, string
	User Interface	View, set, button, listview
	Basic Steps	File, create, project, use
	Interface behavior	Activity, fragment, call, service
WP	Platform	App, 8, 1, T, work
	Notification Services	Data, call, service, notification, method
	Data Binding	Image, item, bind, listbox, control
	Page Navigation	File, page, button, navigation, wp7
iOS	IDE	Xcode, Imag, View, iPhone, Chang
	Programming Language	Object, Swift, c, text, io
	User Interface	Uitableview, view, uiview, cell, anim

Os tópicos mais visualizados podem indicar as barreiras frequentes enfrentadas pelos desenvolvedores, pois essas perguntas podem ser encontradas por qualquer desenvolvedor usando um mecanismo de pesquisa como o Google, por exemplo. A análise do engajamento a partir da perspectiva das questões mais comentadas/visualizadas nos permitiu definir o seguinte *insight*:

Insight #4: Os tópicos mais visualizados, bem como os tópicos em que os desenvolvedores estão mais empenhados em responder, podem indicar uma comunidade de especialistas que pode ajudar a reduzir as barreiras frequentes à participação no MSECO. Isto favorece o crescimento da comunidade e o aumento da satisfação dos desenvolvedores.

4.5.5 (QP5) Existe alguma relação entre perguntas e eventos oficiais?

Para responder a esta pergunta, foi escolhido um período entre fevereiro/2015 e janeiro/2016, uma vez que abrange os anúncios oficiais de cada um dos MSECOs. Uma primeira análise permitiu verificar se existia um comportamento semelhante na frequência de postagem entre os diferentes MSECO (**Tabela 22**).

Para análise estatística, os dados foram normalizados para um intervalo [0,1] e se definiu a seguinte hipótese nula: “*Não há diferença entre a frequência de publicações dos desenvolvedores entre MSECOs em um período de tempo selecionado*”. Então, foi

calculada a frequência de lançamento para cada dia do ano e depois se dividiu cada elemento pelo elemento máximo. Finalmente, calculou-se a média de cada mês.

O teste de *Mann-Whitney* foi aplicado para verificar a normalidade das três amostras com nível de confiança de 95%. Identificou-se que as amostras seguem a distribuição normal. Houve uma diferença estatisticamente significativa entre os grupos, conforme determinado pelo *one-way ANOVA*, $p = 0,0001$. Um teste *post hoc* de *Tukey* revelou que a frequência com que os desenvolvedores do Windows Phone publicam perguntas foi estatisticamente menor do que os desenvolvedores do iOS ($0,334 \pm 0,018$, $p = 0,0001$) e desenvolvedores do Android ($0,37 \pm 0,018$, $p = 0,0001$). Não houve diferença estatisticamente significativa entre os desenvolvedores Android e iOS ($p = ,233$).

Tabela 22. Frequência de postagens entre fevereiro/2015 e janeiro/2016.

Mês/Ano	Windows Phone	Android	iOS
Fevereiro/15	0,43 $\wedge$	0,62	0,59
Março/15	0,34	0,71	0,66
Abril/15	0,39	0,72	0,67
Mai/15	0,37	0,70	0,65
Junho/15	0,35	0,69	0,67
Julho/15	0,39	0,74	0,73 $\wedge$
Agosto/15	0,32	0,71	0,70
Setembro/15	0,31	0,67	0,71
Outubro/15	0,22	0,70	0,69
Novembro/15	0,33	0,67	0,62
Dezembro/15	0,27	0,72	0,66
Janeiro/16	0,32	0,77 $\wedge$	0,70

Na **Tabela 22**, podem ser identificados pontos sazonais dentro da série temporal formada pela frequência de postagens. O ponto mais alto para o Windows Phone no período estudado foi o primeiro mês (fevereiro/2015); foi o último mês da série para postagens relacionadas ao Android (janeiro/16); e foi o sexto mês para iOS (julho/15).

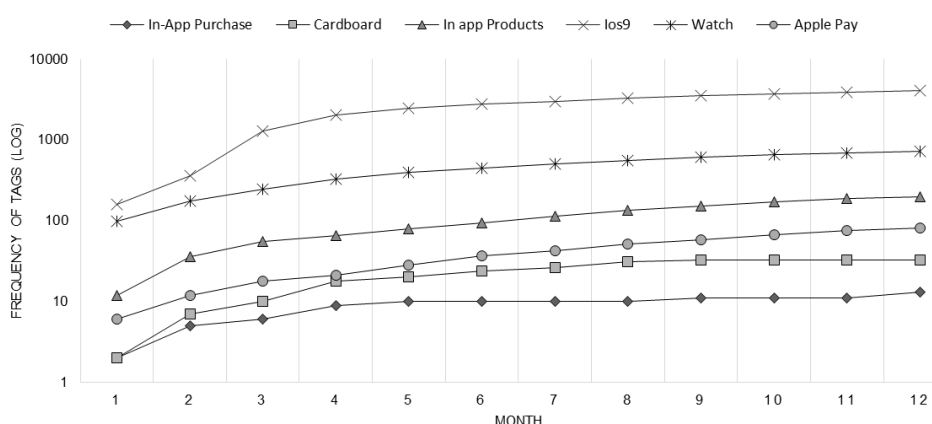


Figura 14. Eventos por MSECO e frequência de tags.

Quando se faz a análise da **Figura 14**, em fevereiro/15, a Microsoft anunciou melhorias para os desenvolvedores do Windows Phone relacionadas a “*In-app Purchase*” e

o “*Windows App Studio*”. Para analisar o impacto da “In-app Purchase”, foi descoberto que as perguntas estão relacionadas a: erro de recuperação de código em uma aplicação móvel na plataforma WP8.0; expectativa ao tentar adicionar moedas ao saldo atual; ao usar um método, um desenvolvedor não pode encontrar o item no catálogo; os desenvolvedores usam um método que está obsoleto na nova versão; e uma vez que a compra tenha sido feita, o código não será executado. O filtro para perguntas relacionadas “*Windows App Studio*” não retornou informações; isso pode estar relacionado ao fato de que o App Studio é um serviço de criação de aplicativos on-line.

Em janeiro/16, o Google anunciou o *SDK do Cardboard para Unity* que envolve suporte a áudio espacial para ajudar a criar experiências de áudio igualmente imersivas em uma aplicação de realidade virtual (VR). Em relação ao “*Cardboard*”, os desenvolvedores publicaram perguntas envolvendo: criação de um player estéreo 360 usando o *cardboard*; um erro envolvendo a interface de usuários em tela reduzida no dispositivo Android; erro no cartão SDK; e erros ao usar um código de demonstração. Quando o termo é “*In app Products*”, os desenvolvedores descrevem perguntas relacionadas a: o faturamento falhou; permissão de compra de um “produto de preço zero”; comportamento da interface do usuário; técnicas para implementar a compra de produto dentro da aplicação móvel; uso de Cordova/PhoneGap; e testar a compra na aplicação móvel não funciona.

No MSECO iOS (julho/15), os anúncios incluíram: *Apple Previews iOS 9*, App de Notícias para iPhone e iPad, *OS X El Capitan*, Novo Software da Apple *Watch - watchOS 2* (Aplicativos de Terceiros Nativos, Novos Faces de Relógio e Recursos Aprimorados de Comunicações) e Expandindo Benefícios com *Merchant Rewards* e *Store Cards (Apple Pay)*. Com o anúncio do *iOS 9*, os desenvolvedores publicaram perguntas relacionadas a: XCode 7, Swift 2.0 e configurações de interface, que envolvem atualização de SDKs, falhas de renderização de fonte, falhas ao tentar iniciar emuladores, uso de TouchID e métodos obsoletos. No caso do *Apple Watch*, as questões estão relacionadas à implementação de recursos, uso de gestos, teste do emulador XCode, interface do usuário e como usar os sensores. Durante o uso do *Apple Pay*, os desenvolvedores questionaram falhas envolvendo o Swift Apple Pay, como usar o Apple Pay com um PayPal SDK (BrainTree) e a integração com o Apple Passport.

A **Figura 14** mostra que o uso de tags relacionadas a anúncios mantém um crescimento acumulado de perguntas até o quarto mês. Depois disso, o comportamento permanece quase constante com base na diferença entre o último mês e o atual. A partir da análise, é possível perceber que questões técnicas surgem quando uma organização central fornece novas tecnologias; a organização deve fornecer efetivamente novas tecnologias, processos ou ideias aos participantes do MSECO. A análise nos levou ao seguinte *insight*:

Insight #5: As perguntas postadas em um repositório de perguntas e respostas em períodos próximos a anúncios oficiais de um MSECO podem ajudar a definir estratégias para adicionar novos recursos do MSECO (por exemplo, plataformas, SDKs, APIs, linguagens de programação). Quando essas novas tecnologias são lançadas no mercado, uma organização central deve ser capaz de gerenciá-las facilmente.

4.5.6 (QP6) Qual é o ranking do número de medalhas recebidas pelos desenvolvedores em cada plataforma?

Para esta questão de pesquisa, foi criado um ranking (**Tabela 23**) do número de medalhas recebidos pelos desenvolvedores dentro dos três MSECOS. Foram utilizadas informações de conjuntos de dados sobre 9.795 desenvolvedores desses MSECOS classificados por reputação - número de medalhas conquistadas. Devido à análise estatística, os dados foram normalizados seguindo os procedimentos adotados na questão de pesquisa anterior.

Uma primeira ação foi testar a seguinte hipótese: “*Não há diferença entre o número de medalhas recebidas pelos desenvolvedores dos diferentes MSECO*”. Aplicando o teste *one-way ANOVA*, obteve-se o valor de significância $p = 0,0001$ que está abaixo de 0,05. Portanto, pode-se dizer que há indício de uma diferença estatisticamente significativa no número de medalhas entre os diferentes MSECO. Um teste post hoc de Tukey revelou que o número de medalhas adquiridas pelos desenvolvedores do Windows Phone foi estatisticamente menor do que os desenvolvedores do iOS ($.00052 \pm 0,00019$, $p = 0,015$) e dos desenvolvedores do Android ($0,00089 \pm 0,00019$, $p = 0,0001$). Não houve diferença estatisticamente significativa entre os grupos Android e iOS ($p = 0,139$).

A partir dos dez primeiros desenvolvedores no ranking, foi possível identificar que alguns deles atuam como *multi-homing*, ou seja, eles atuam em mais de um MSECO, ajudando a responder perguntas e gerenciar comunidades no Stack Overflow. Por exemplo, veja o primeiro: Jon Skeet (**Figura 15**). Do ponto de vista do ecossistema, esse perfil de desenvolvedor é importante porque promove a troca de conhecimento adquirido a partir das interações entre ecossistemas e desenvolvedores.

Também foram analisadas as medalhas conquistadas pelos desenvolvedores em cada MSECO e foi criado um ranking com as cinco medalhas mais frequentes, como mostra a **Tabela 24**. A medalha *Mortarboard* é a única presente no Android (1º), iOS (1º) e Windows Phone (3º). É uma medalha de participação de bronze conquistada quando os desenvolvedores conquistam pelo menos 200 pontos de reputação em um único dia (200 é o máximo diário).

Analisando o MSECO Android, o segundo emblema no ranking é o *Multithread*, ou seja, um emblema de participação ganho quando pelo menos 400 pontos totais de pelo menos 80 respostas do wiki não pertencentes à comunidade forem conquistados. O terceiro distintivo, *Legendary*, é um distintivo de participação de ouro ganho quando 200 pontos de reputação diária é conquistada 150 vezes.

Tabela 23. Ranking - número de medalhas conquistadas.

Ranking	NumBadges (Android)	NumBadges (iOS)	NumBadges (Windows Phone)
1º	14,779	14,779	14,779
2º	5,576	4,074	5,166
3º	4,202	3,093	4,202
4º	4,074	2,588	3,093
5º	3,093	2,130	3,035
6º	3,035	2,062	2,755
7º	2,588	1,865	2,732
8º	2,421	1,650	2,421
9º	2,150	1,623	2,279
10º	2,062	1,621	2,130

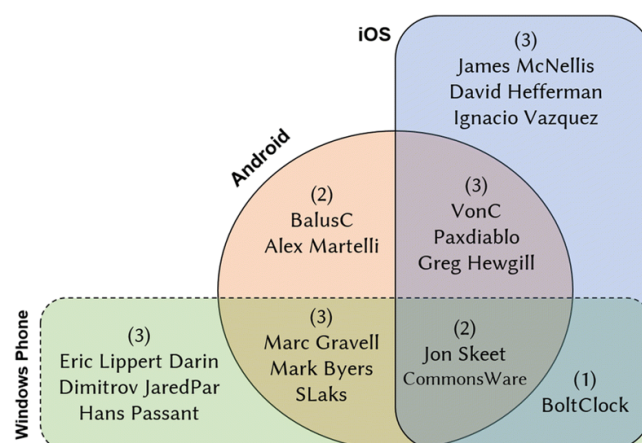


Figura 15. Diagrama de Venn - Desenvolvedores que mais recebem medalhas.

Por sua vez, o *Quorum* é uma medalha de participação de bronze conquistada quando um desenvolvedor alcança uma postagem com pontuação de dois no *Meta Stack Exchange* (ou seja, parte do site em que os usuários discutem os mecanismos e políticas do Stack Overflow). Finalmente, *Great Answer* é uma medalha de resposta de ouro quando uma pontuação de 100 ou mais é conquistada. As cinco medalhas mais frequentes no Android MSECO são relacionadas à participação e um é focado em respostas.

Tabela 24. As cinco medalhas mais recebidas em cada MSECO.

Ranking	Android	iOS	Windows Phone
1º	<i>Mortarboard</i>	<i>Mortarboard</i>	<i>Enthusiast</i>
2º	<i>Multithread</i>	<i>Reviewer</i>	<i>Good Answer</i>
3º	<i>Legendary</i>	<i>Great Answer</i>	<i>Mortarboard</i>
4º	<i>Quorum</i>	<i>Editor</i>	<i>Talkative</i>
5º	<i>Great Answer</i>	<i>Cleanup</i>	<i>Excavator</i>

Para revisões de perguntas e repostas, o filtro foi expandido por um período extra, porque alguns anúncios ocorreram no final do mês indicado. No MSECO iOS, o *Reviewer* é uma medalha de moderação de prata ganha quando o desenvolvedor conclui pelo menos 250 tarefas de revisão. Em seguida, *Great Answer* é uma medalha de resposta de ouro, conforme explicado acima. A quarta medalha, *Editor*, é relacionada a moderação e é conquistada quando o desenvolvedor faz algumas edições pela primeira vez. Finalmente, *Cleanup*, também, é uma medalha de moderação de bronze quando o desenvolvedor fez sua primeira revisão. As cinco medalhas mais ganhas no MSECO iOS estão relacionados à moderação e uma delas se refere a repostas.

No MSECO Windows Phone, a primeira medalha mais frequente é *Enthusiast*, ou seja, uma medalha de participação de prata ganha quando um desenvolvedor visita o Stack Overflow todos os dias por 30 dias consecutivos. A segunda medalha é *Good Answer*, ou seja, uma medalha de prata recebida quando uma resposta recebe pontuação igual ou superior a 25. A medalha *Talkative* faz parte da categoria de participação – essa medalha é obtida quando o desenvolvedor publica dez mensagens avaliadas com uma ou mais estrelas. A última medalha é *Excavator*, ou seja, uma medalha de moderação obtida quando um desenvolvedor editar uma primeira postagem que ficou inativa por seis meses. As cinco medalhas mais frequentes no MSECO Windows Phone estão relacionadas à participação, resposta e moderação.

O último *insight* se refere à identificação de desenvolvedores e comunidades técnicas dentro do Stack Overflow – as comunidades técnicas podem funcionar como “uma extensão” da organização central. Esta extensão se eleva do fluxo de conhecimento técnico e controle da comunidade:

Insight #6: Medalhas podem ajudar a organização central no gerenciamento de estratégias relacionadas a exploração de recurso técnico, desenvolvedores ativos na comunidade e o controle da comunidade por meio do cultivo de relacionamentos com desenvolvedores que se destacam dentro do ecossistema. Isto pode ajudar a reduzir custos operacionais para a governança de desenvolvedores.

4.6 Ameaças à Validade

Validade de *constructo*: a base teórica deste estudo considerou as fragilidades apontadas em recentes revisões de literatura publicadas no campo ECOS, ou seja, estudos aprofundados. A escolha do SO como repositório de perguntas e respostas se deve à presença de desenvolvedores que também postam perguntas e respostas relacionadas ao domínio da plataforma móvel.

Validade interna: os conjuntos de dados não foram selecionados aleatoriamente, mas foram relacionados ao MSECO estudado. Para reduzir o efeito da expectativa dos pesquisadores, as análises do estudo seguiram os procedimentos indicados por algoritmos ou análises estatísticas.

Validade externa: o ambiente não é diferente do real, pois o SO é um repositório com perguntas de desenvolvedores que de alguma forma participam de um MSECO. Além disso, nossa análise considerou os três principais MSECO no mercado: Android, iOS e Windows Phone.

Validade de conclusão: As análises estatísticas e/ou interpretação dos resultados foram baseadas em algoritmos de extração de tópicos (LDA), contagem de palavras e procedimentos para testes de hipóteses com um nível de confiança de 95%.

4.7 Considerações do Capítulo

As contribuições deste estudo envolvem um conjunto de seis *insights* que podem ajudar a organização central a entender o envolvimento do desenvolvedor em um MSECO. A identificação e a compreensão das emoções envolvidas no contexto de MSECO são relevantes para capturar as expectativas e o nível de engajamento dos desenvolvedores.

A partir da perspectiva de DevRel, essas informações podem ajudar a melhorar as estratégias para favorecer o recrutamento e o envolvimento de desenvolvedores dentro do MSECO. O envolvimento do desenvolvedor impacta no aumento ou diminuição do número de contribuições (p.ex.: aplicações móveis, artigos técnicos) de qualidade. Os insights foram utilizados para aprimorar a área de foco de monitoramento do modelo DevGo. Eles também servem para indicar procedimentos de análise e mineração que podem ser utilizados em outras categorias de repositórios.

CAPÍTULO 5 – DEVGO: MODELO DE GOVERNANÇA DE DESENVOLVEDORES

Neste capítulo é apresentado um modelo para a governança de desenvolvedores em MSECO, assim como os estudos envolvidos na concepção e refinamento do modelo. O modelo surge como forma de ajudar na identificação da estrutura que apoia a governança de desenvolvedores em MSECO e estágios do fluxo dos desenvolvedores.

5.1 Estrutura do DEVGO

O DevGo (do inglês, **Developer Governance**) consiste em um modelo composto elementos estruturais e por um conjunto de lições aprendidas para a criação e manutenção de um MSECO próspero para a organização central e para os desenvolvedores. As organizações centrais em MSECO podem se beneficiar do DevGo para saber quais elementos da estrutura do modelo têm sido abordados, ajudando a identificar lições aprendidas e favorecendo a colaboração e a competitividade. Dessa forma, elas poderão obter uma visão sobre a adequação do seu modelo de governança de desenvolvedores.

Na **Figura 16**, é apresentada a estrutura do modelo proposto neste trabalho que envolve o modelo DEVGO e um conjunto de lições aprendidas baseadas na experiência de profissionais de DevRel. O modelo DevGo, com exceção do conjunto de lições aprendidas e os marcos que são descritos de forma textual, é apresentado na **Figura 17**. A descrição dos elementos que compõem o modelo é abordada na próxima seção.

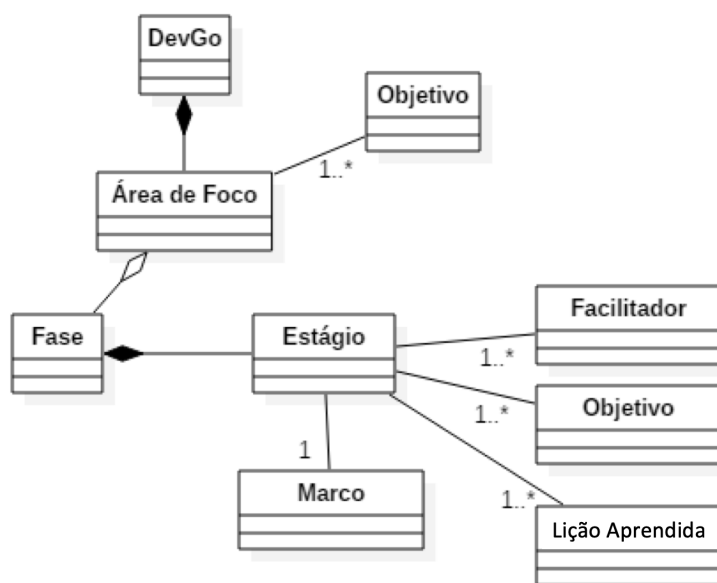


Figura 16. Estrutura do modelo DevGo.

O DevGo, como indicado na **Figura 17**, é composto por **áreas de foco**, que indicam as áreas que uma organização central precisa cuidar para governar desenvolvedores e manter padrões. Cada área de foco possui, pelo menos, um objetivo organizacional. Uma área de foco pode ser composta por **Fases**. Cada uma destas fases está relacionada à mudança no fluxo de avanço do desenvolvedor dentro do ecossistema que é impulsionada por um conjunto de passos a serem dados.

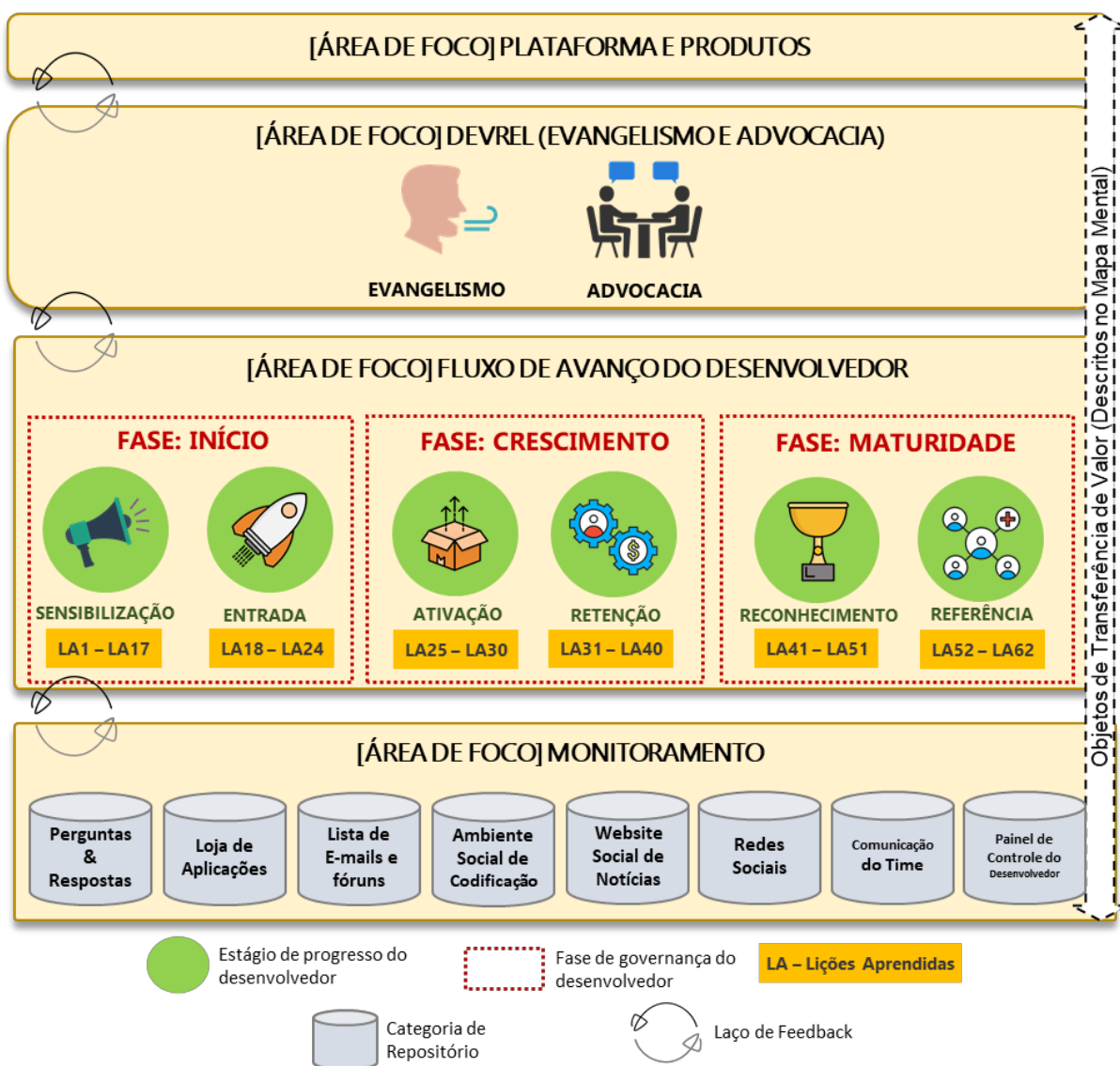


Figura 17. DevGo - modelo para governança de desenvolvedores.

Cada uma das fases é composta por **Estágios**, que compreendem um período de desenvolvimento do desenvolvedor. Um estágio é formado, como mostrado na **Figura 13**, por: um objetivo, um conjunto de facilitadores que permitem a relação com o desenvolvedor, um marco do estágio que representa quando o desenvolvedor poderá sair daquele estágio e um conjunto de lições aprendidas.

O modelo compreende **Quatro áreas de foco: Plataforma e Produtos, DevRel (Evangélismo e Advocacia), Fluxo de Avanço do Desenvolvedor e, por fim, Monitoramento.** Estas áreas de foco ajudam a suportar uma abordagem estruturada e/ou descentralizada (orgânica) para governar desenvolvedores. A seguir, cada área de foco será descrita, juntamente com o seu objetivo.

O conjunto de setas no modelo representam objetos de transferência de valor entre as áreas de foco. Para cada área de foco, que são descritas nas próximas seções, são apresentados mapas mentais onde é possível identificar alguns desses objetos de transferência entre as áreas de foco. Os objetos de transferência de valor são mecanismos de criação de valor para gerar e distribuir valor para todo o ecossistema baseando-se em inovação, investimentos e compartilhamento de custos.

5.1.1 Áreas de foco

5.1.1.1 Plataforma e produtos

Esta área de foco tem como objetivo fornecer informações e recursos que apoiem as metas de uma organização em relação à produtividade, criação de nicho e qualidade das contribuições. Compreende a plataforma, infraestrutura, orçamento, produtos (p.ex.: APIs, SDKs, IDEs) e serviços da organização central, uma vez que uma organização investe em um ecossistema com o objetivo de atrair usuários para consumir seus produtos. A visão e os incentivos da organização central podem ser armazenados em repositórios proprietários, como as lojas de aplicações móveis e portais de apoio aos desenvolvedores (por exemplo, *Android Developers*²²). Assim como, podem fazer uso de repositórios de código aberto para favorecer um ambiente social de codificação. Na **Figura 18**, são apresentados os objetos de transferência de valor que são consumidos e providos por esta área de foco.

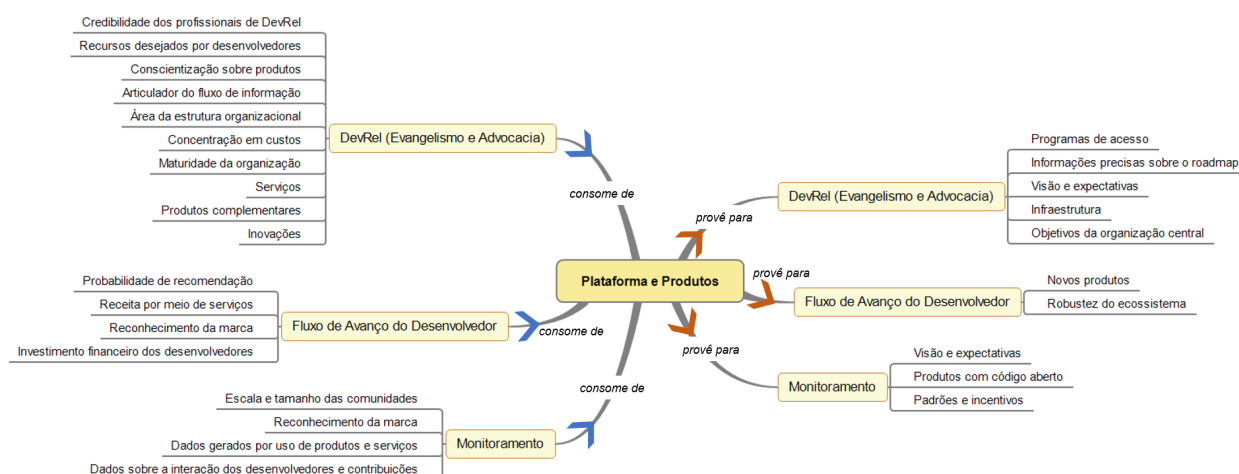


Figura 18. Objetos de transferência de valor – Plataforma e Produtos.

²² developer.android.com

Como exemplo de ações dentro desta área de foco, podem ser destacados os portais de apoio aos desenvolvedores da Google e da Apple, que possuem informações sobre ferramentas, recursos disponíveis que apoiam o desenvolvimento e produtos focados nos usuários de dispositivos da plataforma. É importante deixar claro que esta área está fortemente direcionada ao que a organização tem como visão em relação aos seus negócios.

5.1.1.2 DevRel (Evangelismo e Advocacia)

Esta área de foco tem como objetivo ajudar a incorporar contribuições com potencial (ou seja, produtos complementares, serviços e inovações) que surgem dos desenvolvedores para a plataforma do ecossistema na área de foco da Plataforma e Produtos. Isto ajuda a manter o equilíbrio entre as expectativas dos desenvolvedores e as necessidades da organização central. Incluindo o balanceamento entre o roadmap interno da organização com as necessidades dos desenvolvedores.

Esta área é composta por evangelismo e advocacia. A parte de advocacia trabalha com desenvolvedores existentes, ou seja, com o ganho de interesse e a fidelização de potenciais desenvolvedores. Isto se dá por meio de conteúdo de qualidade e específico para esses desenvolvedores. A parte de advocacia está relacionada aos estágios de retenção, reconhecimento e referência dentro do DevGo. O evangelismo foca na divulgação da “palavra” da organização, ou seja, na prospecção de desenvolvedores, na influência de desenvolvedores externos. A parte de evangelismo está associada aos estágios de sensibilização, entrada e ativação do DevGo. Como característica comum, tanto advocacia quanto o evangelismo devem construir confiança entre os setores da organização central e os desenvolvedores.

Nesta área de foco são fornecidos os recursos apropriados, como componentes e ferramentas de código aberto, para dar suporte aos desenvolvedores, dividindo as metas de uma organização de acordo com vários públicos-alvo. Estes recursos e ferramentas são desenvolvidos com base em diretrizes organizacionais que incluem especificação de plataforma, ideias emergentes, melhores práticas, tecnologias, ferramentas de desenvolvimento e marketing, critérios de qualidade e projeto de interface com o usuário. Os objetos de transferência de valor que são consumidos e providos por esta área de foco são apresentados na **Figura 19**.

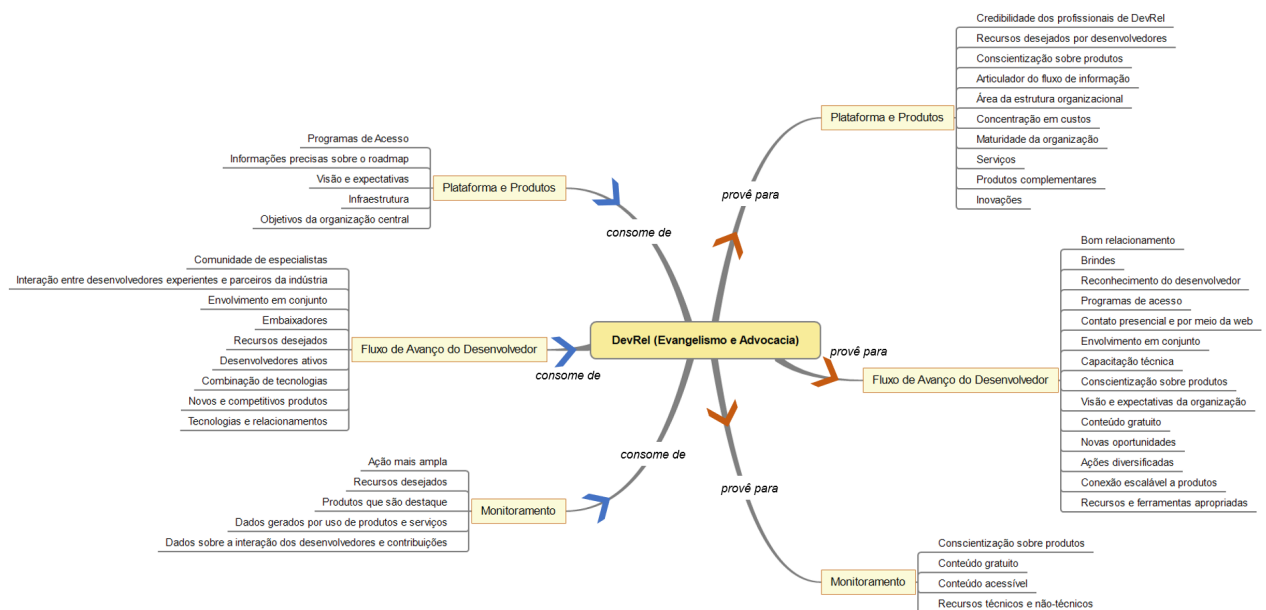


Figura 19. Objetos de transferência de valor – DevRel (Evangelismo e Advocacia).

5.1.1.3 Fluxo de avanço do desenvolvedor

Esta área de foco tem como objetivo apoiar no acompanhamento e análise do avanço dos desenvolvedores dentro do ecossistema. Entender como os desenvolvedores estão se movimentando e gerando contribuições dentro do ecossistema é importante para uma organização direcionar suas estratégias de atuação para governá-los. De fato, se há muitos desenvolvedores com problemas para a entrada no ecossistema, a organização passará por problemas de ativação e retenção, por exemplo, estágios necessários para a geração de contribuições com qualidade.

A **Figura 20** apresenta os objetos que ajudam na transferência de valor nesta área de foco (os consumidos e os providos pela área).

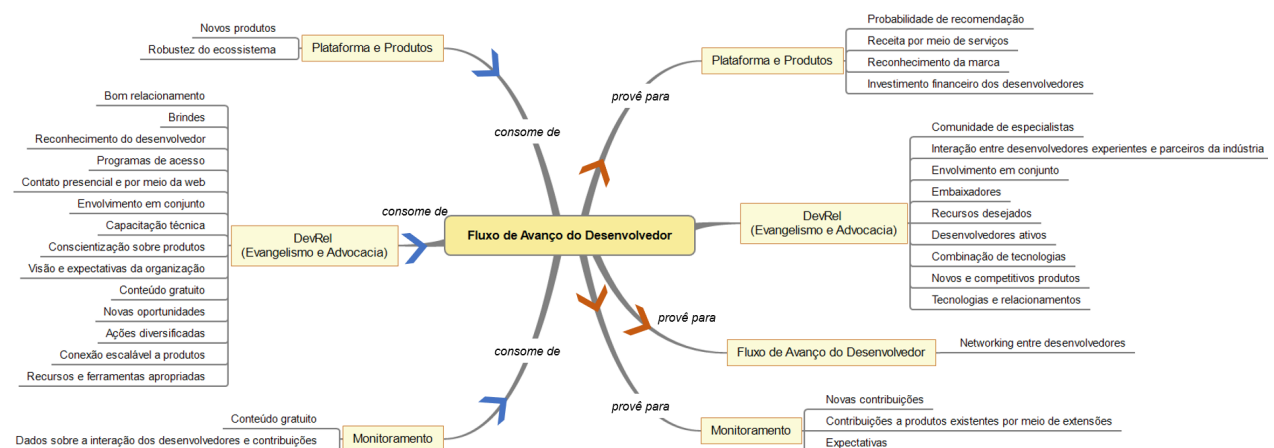


Figura 20. Objetos de transferência de valor – Fluxo de Avanço do Desenvolvedor.

Especificamente nesta área há um conjunto de fases, como explicado na estrutura do modelo (**Figura 16**), e que são descritos mais a frente. Há três fases para apoiar o

progresso do desenvolvedor, essas fases consideram modelos de negócio e de gerenciamento de parceria. Tem-se, desta forma, as seguintes fases: Início, Crescimento e Maturidade.

5.1.1.4 Monitoramento

Esta área de foco tem como objetivo servir como uma estratégia para monitorar o engajamento do desenvolvedor, dando suporte à transparência tanto para a organização quanto para os desenvolvedores. Desta forma é possível garantir que todos tenham a chance de entender e fornecer feedback. Além disso, apresenta mecanismos para armazenar dados sobre as contribuições e interações dos desenvolvedores. Isto ajuda a promover o ciclo de feedback para adaptar as estratégias de governança do desenvolvedor na área de foco da Plataforma e Produtos e na DevRel (Evangelismo e Advocacia).

Na **Tabela 25**, são apresentadas as categorias, descrição e exemplo de repositórios que podem ser utilizados, pelos profissionais de DevRel, dentro da área de foco de monitoramento.

Tabela 25. Categorias de Repositórios.

Categoria	Descrição	Exemplo
Perguntas e Respostas	Guardam perguntas e respostas em torno de problemas e soluções técnicas descritas por desenvolvedores	Stack Overflow
Lista de E-mails e Fóruns	Armazenam discussões em torno de qualquer assunto relativo aos ecossistemas.	Android Developers, Apple Developers e Discourse
Ambiente Social de Codificação	Armazenam projetos, códigos, defeitos e outros artefatos.	Github e Bitbucket
Website Social de Notícias	Guardam informações sobre notícias compartilhadas pelos usuários em sub-comunidades.	Reddit e Hacker News
Redes Sociais	Ferramentas de interação social que envolvem conteúdo como imagens, textos, vídeos e <i>emoticons</i> .	Facebook e Twitter
Comunicação do Time	Ferramentas de comunicação que integram informações entre equipes e que ajudam a se organizar melhor.	Confluence, Gitter, Slack, Telegram e WhatsApp
Loja de Aplicações	Armazenam produtos de software que podem ser adquiridos e revisados por usuários. São fontes de informação sobre a qualidade dos produtos de um ecossistema.	App Store e Google Play
Painel de Controle do Desenvolvedor	São os portais de desenvolvedor que apoiam a submissão de novas contribuições para a Loja.	Android Developers e Apple Developers

A categorização destes repositórios foi descrita no **Capítulo 3, Seção 3.5**. Já na **Tabela 26** são apresentadas as principais motivações para o uso das categorias de

repositórios. As motivações foram identificadas no estudo sobre percepção de valor na **Seção 3.6**.

Os repositórios (p.ex.: Loja de Apps, Questões e Respostas, Relatórios de Defeitos e Projeto de Software) armazenam dados gerados a partir do uso de recursos e objetivos fornecidos por uma organização central para ajudar na expansão do ecossistema. Esta área de foco permite com que o DevGo possua um canal de comunicação bidirecional, fornecendo informações sobre conhecimento e experiência a partir das direções *bottom-up* e *top-down* (alimentando todas as áreas de foco). Os repositórios suportam o alinhamento entre os ecossistemas locais (p.ex.: uma comunidade específica na África do Sul) e o ecossistema global de desenvolvedores.

Tabela 26. Repositório e Motivações.

Categoria	Motivações para Uso
Perguntas e Respostas	Satisfação dos Desenvolvedores; Crescimento da Comunidade; Custo Operacional; Competitividade do Produto; Custo de Manutenção
Lista de E-mails e Fóruns	Competitividade do Produto; Esforço de Desenvolvimento do Produto; Custo Operacional; Estratégia de Longo Prazo; Custo de Manutenção; Crescimento da Comunidade
Ambiente Social de Codificação	Valor Financeiro; Crescimento da Comunidade; Estratégia de Longo Prazo; Custo de Manutenção; Satisfação do Desenvolvedor; Esforço de Desenvolvimento do Produto
Website Social de Notícias	Crescimento da Comunidade; Competitividade do Produto; Tempo para comercialização; Custo de Entrega; Custo de Manutenção; Custo Operacional
Redes Sociais	Crescimento da Comunidade; Satisfação dos Desenvolvedores; Estratégia de Longo Prazo; Competitividade do Produto; Tempo para Comercialização; Custo Operacional
Colaboração de Time	Esforço de Desenvolvimento do Produto; Estratégia de longo prazo; Satisfação do Desenvolvedor; Facilitador de Eficiência Técnica; Crescimento da Comunidade; ROI
Loja de Aplicações	Competitividade do Produto; Custo de Manutenção; Custo de Entrega; Valor Financeiro
Painel de Controle do Desenvolvedor	Custo de Manutenção; Tempo para Comercialização; Competitividade do Produto

Os objetos de transferência de valor para esta área de foco, que consome e provê valor para todas outras áreas de foco, são descritos na **Figura 21**.

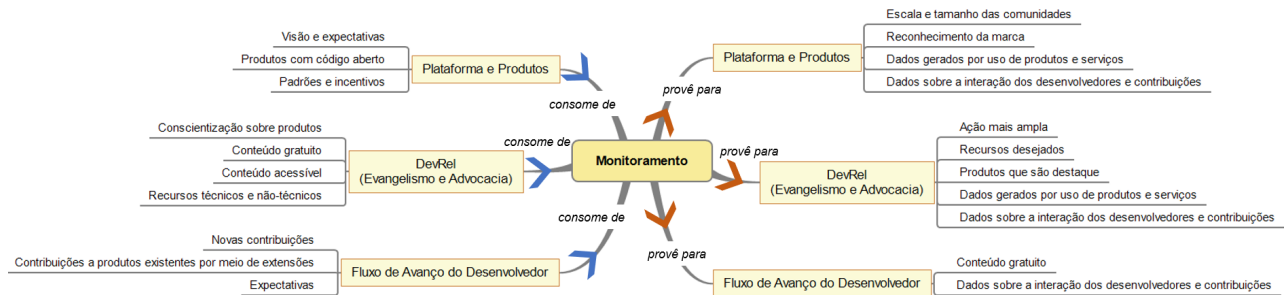


Figura 21. Objetos de transferência de valor – Monitoramento.

Especificamente, nesta tese, analisamos um repositório da categoria Perguntas e Respostas, o Stack Overflow, como forma de obter insights para o monitoramento a partir desta categoria. Esta análise de insights é descrita no **Capítulo 4** e são baseadas em mineração de repositórios de software como forma de obter informações que ajudem na governança de desenvolvedores. Os insights são os seguintes:

- **Insight #1:** os tópicos podem indicar as barreiras mais frequentes enfrentadas por desenvolvedores dispostos a participar de um MSECO. Esse cenário pode servir como uma estratégia de monitoramento para apoiar a organização central no recrutamento e no treinamento de desenvolvedores. Do ponto de vista do custo operacional, de manutenção e a competitividade, os tópicos comuns: vinculação de dados, programação de UI e infraestrutura de desenvolvimento, ajudam na análise;
- **Insight #2:** as emoções primárias além de serem úteis pra obter a satisfação do desenvolvedor, também cobrem, aspectos de custo, competitividade e manutenção de produtos. A tristeza se relaciona a produtos como os emuladores e dispositivos, assim como, a manipulação de elementos de UI. A raiva está associada a indisponibilidade de recursos. E a alegria, pode impactar na competitividade do produto, pois há relação com as melhores práticas para gerar uma contribuição e a implementação de recursos avançados;
- **Insight #3:** é importante explorar como a falta de interseção de interesses impulsiona a troca de informações entre desenvolvedores trabalhando em mais de um MSECO. A pequena interseção mostra que as estratégias tomadas em nível gerencial podem afetar as interações e motivações dos desenvolvedores;
- **Insight #4:** os tópicos mais visualizados, bem como os tópicos em que os desenvolvedores estão mais empenhados em responder, podem indicar uma comunidade de especialistas que pode ajudar a reduzir as barreiras

frequentes à participação no MSECO. Isto favorece o crescimento da comunidade e o aumento da satisfação dos desenvolvedores;

- **Insight #5:** as perguntas postadas em um repositório de perguntas e respostas em períodos próximos a anúncios oficiais de um MSECO podem ajudar a definir estratégias para adicionar novos recursos do MSECO (por exemplo, plataformas, SDKs, APIs, linguagens de programação). Quando essas novas tecnologias são lançadas no mercado, uma organização central deve ser capaz de gerenciá-las facilmente;
- **Insight #6:** as medalhas podem ajudar a organização central no gerenciamento de estratégias relacionadas a exploração de recurso técnico, desenvolvedores ativos na comunidade e o controle da comunidade por meio do cultivo de relacionamentos com desenvolvedores que se destacam dentro do ecossistema. Isto pode ajudar a reduzir custos operacionais para a governança de desenvolvedores.

5.1.2 Fases, estágios e facilitadores

Cada fase compreende um conjunto de **Estágios de Avanço do Desenvolvedor**. Cada um destes estágios colabora para um laço de feedback que aumenta tanto o conhecimento da organização sobre o MSECO quanto dos desenvolvedores. Este laço de feedback apoia a organização central, assim como os desenvolvedores por meio da solução de problemas e da redução de riscos (situação de cooperação). O laço de feedback consiste no exercício contínuo de colher as percepções/expectativas dos desenvolvedores e facilitar com que sejam respondidas. Esta resposta pode ser tanto pela organização central, por meio dos profissionais de DevRel, quanto pela comunidade desenvolvedores.

Cada um dos estágios está associado a um conjunto de facilitadores. Os **Facilitadores** são mecanismos da organização associados a cada estágio do progresso do desenvolvedor para ajudar os desenvolvedores a alcançarem seus próprios objetivos. O treinamento é um facilitador comum a todos os estágios que compõem o DevGo, uma vez que é um dos mecanismos principais de formação de desenvolvedores e de troca de conhecimento entre os profissionais de DevRel e os desenvolvedores. A cada um dos estágios ainda há um **marco** que pode ser uma das indicações de que o desenvolvedor já passou por um determinado estágio e avançará para o próximo.

5.1.2.1 Fase de início

Nesta fase, o desenvolvedor aprende sobre a cultura, costumes e realidades do ecossistema. É a fase da decisão em participar do ecossistema. O desenvolvedor é inserido no contexto da plataforma, dos produtos e da comunidade de desenvolvedores e usuários

do ecossistema. Nesta fase, por exemplo, são importantes as informações sobre a base instalada de usuários (p.ex.: usuários que estão utilizando ativamente os produtos em torno da plataforma), dispositivos e ferramentas para desenvolvimento disponíveis. Para isto, esta fase envolve dois estágios: Sensibilização e Entrada. A fase de início envolve o trabalho de evangelismo pelos profissionais de DevRel.

5.1.2.1.1 Estágio de sensibilização

Neste estágio, a organização central, por meio de sua equipe de DevRel, deve mostrar ao desenvolvedor que o ecossistema é atrativo para as expectativas dele. Essa sensibilização pode acontecer, também, por meio dos seguintes facilitadores: roadmap de produtos, eventos (*hackathons*, conferências, palestras, *meetups*) e mídias sociais.

Objetivos: Anunciar roadmap de dispositivos que serão lançados no mercado e ferramentas de desenvolvimento; Atrair desenvolvedores por meio de conferências, *hackathons*, palestras, redes sociais e propagandas; Comunicar ajustes na plataforma; Envolver líderes técnicos e desenvolvedores que atuam como influenciadores dentro do ecossistema.

O marco que indica que o desenvolvedor já passou por este estágio é quando um desenvolvedor se associa ao ecossistema por meio de inscrição (p.ex.: cadastro em portal de desenvolvedores ou cadastro para submissão de contribuição).

Para este estágio há um conjunto de 17 lições aprendidas:

- LA1. Divulgue os eventos (mídias sociais). Para isso, crie uma matriz de proposição de valor - para os desenvolvedores, organização e comunidade. Os eventos são como um produto, em vez de uma “atividade” única. Determine se isso ajuda a dar suporte e aumentar as metas da sua organização;
- LA2. Defina uma abordagem estruturada para a aprendizagem avançada, por exemplo, ferramentas como Udacity pode ser uma tentativa;
- LA3. Fomente grupos de discussão online para abordar novos produtos. Podcasts, blogposts, boletins informativos são bons para compartilhar uma mensagem. É fundamental para entender os desenvolvedores quer para melhorar seus produtos de desenvolvedor. Você descobrirá rapidamente os principais pontos problemáticos;
- LA4. Planeje iniciativas educacionais como laboratórios de codificação, workshops, *hackathons*, webinars, etc. Permita um tempo de planejamento adicional caso o evento aconteça durante uma época movimentada do ano;

- LA5. Crie listas de discussão segmentadas com base em experiência e interesse. Isto ajudará a personalizar e-mails e informações favorecendo a criação e suporte de variados nichos. Isto facilitará sua resposta rápida aos desenvolvedores;
- LA6. Execute a codificação ao vivo ao conduzir uma palestra pois é uma maneira incrível de mostrar como é fácil implementar e usar um determinado SDK ou API.
- LA7. Forneça links para seus produtos e canais oficiais de comunicação tanto em material online quanto em ações presenciais. Isto deve ser acessível sem registro;
- LA8. Liste e divulgue todos os eventos que você está apoiando, planejando visitar, etc;
- LA9. Envolver-se com seus desenvolvedores em seu habitat nativo (ou seja, universidades, conferências de desenvolvedores) priorizando a criação de nicho. Você precisa colher feedback para equipe do produto e ajudar a priorizar o roadmap;
- LA10. Comunique-se com seu público (crescente) através de mídias sociais, blogs, fóruns e canais do Slack. Qualquer que seja a sua escolha, você precisa administrá-la bem e se comunicar ainda melhor. Conceda uma mensagem de para cada produto e grupo de desenvolvedores. Estabelecer periodicidade é uma ótima ideia, porque as pessoas saberão que você está mantendo um canal de comunicação sério;
- LA11. Considere o trabalho conjunto com desenvolvedores e empresas parceiras durante o planejamento de eventos presenciais. Isto imediatamente expande a comunidade antes mesmo de você realizar o evento. Crie um plano de divulgação mostrando a equipe de DevRel que estará presente e o logotipo da organização. As pessoas se acostumarão com essas duas imagens e, quando o evento acontecer, os desenvolvedores reconhecerão a sua equipe de DevRel e já confiará nela.
- LA12. Considere lançar e divulgar os produtos sob uma licença de código aberto, a menos que haja uma razão convincente para não o fazer;
- LA13. Converse com os desenvolvedores durante um evento. Mesmo que não resulte em conversão em contribuição, qualquer interação ou assistência positiva, no mínimo, cria uma imagem positiva de sua empresa. Isto pode resultar indiretamente na adoção futura de produtos por essas mesmas pessoas ou por alguém que eles recomendam seu produto;

- LA14. Disponibilize seus slides de apresentações de forma editável à comunidade onde entregou um evento. Dessa forma, a comunidade pode reutilizá-los e traduzi-los;
- LA15. Saiba o máximo que puder sobre seus produtos e os represente de uma forma tecnicamente confiável. Pois os desenvolvedores sempre têm a opção de encontrar artigos sobre seus produtos em canais de terceiros;
- LA16. Pergunte aos seus desenvolvedores o que eles querem ler. Isto pode ser feito em mídias sociais. Além de suas mídias sociais, use as contas da área de DevRel para que outros profissionais da equipe possam dar suporte;
- LA17. Priorize divulgação de conteúdo relacionado aos produtos por meio de ferramentas SEO e de palavras-chave, como: keyword.io, keywordtool.io, Google AdWords Keyword Planner e Google Webmaster Tools.

5.1.2.1.2 Estágio de entrada

Este estágio está relacionado ao objetivo do desenvolvedor de gerar alguma contribuição para o ecossistema ou para sua carreira profissional. Aqui o desenvolvedor passou pelo estágio de sensibilização e possui alguma motivação em atuar no MSECO. Neste estágio o desenvolvedor é considerado novato em algum produto do ecossistema pois está iniciando sua possível contribuição. Desta forma é importante reduzir as barreiras para a participação deste desenvolvedor.

O profissional de DevRel pode se utilizar do seguinte conjunto de facilitadores: pacote de benefícios, taxas de parceria e suporte técnico.

Objetivos: Estabelecer parceria; Manter capacidade de absorção de novos e/ou potenciais desenvolvedores; Apoiar o desenvolvimento; Apoiar as negociações; Propiciar a conexão entre desenvolvedores novatos e desenvolvedores experientes.

O marco que indica que o desenvolvedor já passou por este estágio é quando um desenvolvedor está com uma contribuição pronta em nível de submissão para algum dos repositórios do MSECO.

Para este estágio há um conjunto de 7 lições aprendidas:

- LA18. Construa, atualize e compartilhe um conjunto bem projetado de ferramentas de apoio ao desenvolvedor;
- LA19. Concentre-se em indivíduos que tenham interesse perceptível em uma tecnologia ou abordagem específica entre os produtos do ecossistema;
- LA20. Comunique os benefícios econômicos de construir e desenvolver contribuições;

- LA21. Ajude os desenvolvedores a se sentirem em casa quando interagirem com a equipe DevRel ou com os serviços e produtos da MSECO, mesmo se o desenvolvedor já for contribuidor de outro ecossistema;
- LA22. Direcione os desenvolvedores a se envolver em comunidades em torno de algo tangencial aos seus produtos;
- LA23. Crie e mantenha a documentação de seus produtos. Torne as informações fáceis de encontrar e entender;
- LA24. Insira o desenvolvedor um programa de fidelização completo para desenvolvedores - que ofereça suporte e engaje os desenvolvedores, adicionando valor tanto para o desenvolvedor quanto para a organização.

5.1.2.2 Fase de crescimento

Nesta fase o desenvolvedor deve ter a sua disposição os recursos necessários para avançar na aquisição de conhecimento teórico e prático para gerar contribuições para a expansão do ecossistema. Como parte desta fase foram identificados dois estágios: Ativação e Retenção.

5.1.2.2.1 Estágio de ativação

Este estágio funciona como um gatilho que indica se o desenvolvedor gerou sua primeira contribuição para o ecossistema, por exemplo, por meio da publicação de uma aplicação móvel. Neste estágio, desenvolvedor está projetando, desenvolvendo e submetendo sua primeira contribuição. Os seguintes facilitadores podem ser utilizados por profissionais de DevRel: portfólio de dispositivos e aplicações móveis, certificação de contribuição e acordos de níveis de qualidade.

Objetivos: Fornecer impulso de ganho; Gerenciar mudança de plataforma; Apoiar contribuições de nicho; Apoiar o desenvolvimento; e Analisar os desenvolvedores periféricos, ativos e de topo.

O marco que indica que o desenvolvedor já passou por este estágio é quando um desenvolvedor está com uma contribuição publicada (exemplo: aplicação móvel, funcionalidade em projeto open-source, artigo técnico) em algum dos repositórios do MSECO.

Para este estágio há um conjunto de 7 lições aprendidas:

- LA25. Direcione o desenvolvedor durante o processo de desenvolvimento a compartilhar sobre a experiência com o ecossistema em mídias sociais;
- LA26. Envolver os desenvolvedores para falar sobre os pontos problemáticos e as soluções que podem ser implementadas. Os desenvolvedores não precisam que

você os impressione resolvendo todos os problemas deles, concentre-se naqueles que o produto de sua empresa resolve;

LA27. Desenvolva sua comunidade planejando e executando Hackathons. Um evento deste tipo ajuda a engajar os desenvolvedores ativos do ecossistema. Possibilitando que todos se conheçam e dando a oportunidade de mostrar que você se importa com eles;

LA28. Envolve os desenvolvedores em uma coleção de materiais escritos por eles para a comunidade;

LA29. Certifique-se de que a documentação de produtos para o desenvolvedor seja uma parte essencial da experiência no site da plataforma. Deve ser fácil de navegar, limpo e simples para fornecer instruções fáceis de serem avaliadas e seguidas pelos desenvolvedores;

LA30. Forneça modelos de projetos e diretrizes de cenários de uso para que os desenvolvedores possam evoluir as contribuições com os produtos e serviços do ecossistema. Isto permite a evolução das contribuições com a menor resistência possível dentro de um prazo muito curto.

5.1.2.2.2 Estágio de retenção

Neste estágio um desenvolvedor continua a usar a plataforma, bem como recursos novos/adicionais e fornece novas contribuições. Porém o desenvolvedor conta com outras oportunidades em concorrentes, desta forma, neste estágio é importante valorizar o desenvolvedor para retê-lo em relação a monetização e oportunidades de benefícios e a própria cultura dentro do MSECO. São alguns facilitadores que ajudam o profissional de DevRel neste estágio: portfólio de dispositivos e aplicações móveis, certificação de contribuição, plano de negócios, modelo de geração de renda, acordos de níveis de qualidade e colaboração social.

Objetivos: Fornecer impulso de ganho; Gerenciar mudança de plataforma; Apoiar contribuições de nicho; Permitir que os desenvolvedores trabalhem com as mais recentes e melhores tecnologias do ecossistema; Apoiar o desenvolvimento; e Analisar os desenvolvedores periféricos, ativos e de topo.

O marco que indica que o desenvolvedor já passou por este estágio é quando um desenvolvedor gerou contribuição considerada pela organização relevante para o ecossistema. Por exemplo, uma aplicação móvel que alcançou alta visibilidade, um recurso técnico que é muito utilizado pela comunidade ou uma correção de bug crítico em um projeto importante da plataforma.

Para este estágio há um conjunto de 10 lições aprendidas:

- LA31. Concentre-se no valor para o desenvolvedor em qualquer conteúdo que você esteja produzindo, seja em posts de blog, guias, etc. Isso significa que o foco deve estar na resolução de problemas;
- LA32. Reconheça as diversas motivações de cada membro da comunidade, pois não existe um único tipo de desenvolvedor, e busque continuamente alinhar os interesses de todos. Entenda e declare claramente o que a organização por meio de um programa de fidelização de desenvolvedores está tentando alcançar. Isto permitirá o suporte às metas de várias unidades de negócios. Experimente ter um portfólio que inclua uma série de atividades, desde ganhos rápidos a projetos grandes/importantes;
- LA33. Chame a atenção dos desenvolvedores para o fato de que sua própria marca crescerá com a quantidade de contribuições que eles entregaram e a fama que recebem por isso;
- LA34. Esteja presente no Stack Overflow. Você pode ver o que os desenvolvedores escrevem sobre seu produto e, mesmo que outros desenvolvedores não consigam ajudar uns aos outros, você pode entrar e ajudar alguém. Mostre que você está lá para ajudá-los quando precisarem de você;
- LA35. Fique de olho na competição. Faça um benchmark contra os principais concorrentes para acompanhar as tendências ao longo do tempo. Além de ajudá-lo a entender os líderes do setor, isso também ajuda a evitar as armadilhas para a retenção de desenvolvedores;
- LA36. Envolver os desenvolvedores em demonstrações e prévias de novos produtos e ferramentas. Assim, além de terem acesso antecipado, eles ajudarão a encontrar possíveis problemas para uso;
- LA37. Analise as contribuições de seus desenvolvedores, escreva sobre e divulgue em canais oficiais e durante eventos;
- LA38. Incentive os desenvolvedores a lerem “histórias de desenvolvedores”, um site dedicado sobre como os desenvolvedores estão obtendo sucesso dentro ecossistema. Pode ajudar os desenvolvedores a criar um negócio, não apenas uma aplicação móvel;
- LA39. Seja inclusivo compartilhando grandes bibliotecas de código aberto e comunitárias como parte da solução. O Android tornou-se muito mais inclusivo para a comunidade. O exemplo mais óbvio é o “abraço” do Kotlin e o trabalho coletivo para torná-lo o melhor possível no Android;

LA40. Direcione os desenvolvedores, também, a se envolver com as comunidades que fazem o trabalho em áreas específicas da plataforma (por exemplo, fragmentação e construção de bibliotecas de suporte).

5.1.2.3 Fase de maturidade

Nesta fase o desenvolvedor precisa se manter atualizado, compartilhar experiências, estabelecer confiança com a organização central e a comunidade de desenvolvedores. É ainda importante que o desenvolvedor seja reconhecido pelas contribuições e seja preparado para movimentar a comunidade atuando assim como uma extensão da equipe de DevRel.

Um desenvolvedor nesta fase é referência para os outros desenvolvedores do ecossistema e precisa de uma ligação direta com a organização central por meio da equipe de DevRel. Esta fase compreende um estágio de reconhecimento do desenvolvedor (Estágio de reconhecimento) e um estágio que ajuda a identificar e preparar desenvolvedores que são referência para o ecossistema (Estágio de referência), como segue.

5.1.2.3.1 Estágio de reconhecimento

Uma organização deve perceber e destacar dentro do ecossistema seus melhores desenvolvedores em aspectos que demonstrem contribuições impactantes no ecossistema, por exemplo, histórias de sucesso de desenvolvedores, extensões como ferramentas, relatórios de correções da plataforma e aplicações móveis que se destacam na loja. Os seguintes facilitadores ajudam o profissional de DevRel dentro deste estágio: programas de apoio ao desenvolvedor, benefícios de marketing, incentivos para inovação, meritocracia e colaboração social.

Objetivos: Levar interações positivas e relacionamentos mais próximos; Suportar o reconhecimento de desenvolvedores por usuários, organização e comunidade; Identificar os líderes da comunidade; Construir e obter feedback sobre os produtos; Comunicar os impactos reais do trabalho dos desenvolvedores.

O marco que indica que o desenvolvedor já passou por este estágio é quando o desenvolvedor obtém o reconhecimento de uma comunidade local e possui habilidades próximas de um perfil de profissional de DevRel (liderança, comunicação e conhecimento técnico das tecnologias envolvidas no ecossistema).

Para este estágio há um conjunto de 11 lições aprendidas:

- LA41. Trabalhe diretamente com os vencedores do prêmio em hackathons, promovendo os projetos através de blogs usando a contribuição como material promocional, se isso fizer sentido;
- LA42. Promova as contribuições dos desenvolvedores como, por exemplo, as aplicações móveis. Impulsionar os downloads de aplicações móveis criará muito valor em termos de relações com desenvolvedores e também ajudará a vender sua plataforma;
- LA43. Elogie alguns desenvolvedores específicos em seus canais oficiais. Isto pode ser um grande impulso para o reconhecimento público e negócios do desenvolvedor;
- LA44. Conecte desenvolvedores em potencial da comunidade aos colegas de setores organizacionais. É importante, também, ter uma variedade de programas para conectar os desenvolvedores diretamente aos clientes de pequenas e médias empresas;
- LA45. Ajude os desenvolvedores a ajudar a sua área de DevRel. Seminários, eventos, seminários on-line, demonstrações - o que você estiver fazendo para chamar a atenção da sua plataforma para o mundo, convide alguns desenvolvedores para compartilhar conhecimento e experiências;
- LA46. Ajude a criar status e identificar líderes na comunidade. Os desenvolvedores querem construir seu status social, ganhar reputação e compartilhar seus conhecimentos com outros desenvolvedores;
- LA47. Invista na capacitação de líderes de comunidade. Esses líderes poderão trabalhar como extensão da equipe de DevRel recrutando, treinando e capacitando os desenvolvedores do ecossistema;
- LA48. Dar reputação às pessoas certas, demonstrar apreço e recompensar os mais ativos. Quando feito corretamente, as pessoas criam palestras, escrevem livros, ajudam a desenvolver a comunidade junto com você. Quando os desenvolvedores podem aumentar seu valor, seu alcance e seu reconhecimento construindo sua plataforma e aumentando seu público-alvo, eles serão muito mais dedicados a você e sua marca;
- LA49. Ofereça treinamento 1:1 em vários níveis como recompensa - em vez de se cansar treinando todos, tenha materiais de auto treinamento que os desenvolvedores possam usar para chegar ao próximo nível;
- LA50. Ter um código de conduta definido para os líderes e para as comunidades - sua reputação também está em jogo quando um dos membros da sua comunidade "sai da linha";

LA51. Conecte sua equipe de aquisição de Talentos aos desenvolvedores, instrua essa equipe sobre como a comunidade está funcionando. Isto ajuda na identificação de comunidades e os influenciadores podem fazer parceria para criar conteúdo incrível para a área de DevRel.

5.1.2.3.2 Estágio de referência

Neste estágio o foco é fazer com que um desenvolvedor, identificado como líder no estágio anterior, seja capacitado para influenciar a comunidade e atuar na sensibilização de novos desenvolvedores. O influenciador conta a outros sobre a plataforma (neste cenário, relacionamentos são compromissos bidirecionais e representam um investimento do desenvolvedor). Como facilitadores para este estágio se tem: suporte técnico, suporte de Marketing e rede de influenciadores.

Objetivos: Estabelecer e apoiar uma rede de influenciadores especializados para dimensionar as atividades do ecossistema; Liderar parcerias estratégicas; Capacitar os principais desenvolvedores e principais colaboradores como influenciadores.

Especificamente neste estágio não há um marco de mudança do fluxo de avanço do desenvolvedor. Por ser um estágio de referência, o desenvolvedor se potencializa como um influenciador e no trabalho aproximado com os profissionais de DevRel.

Para este estágio há um conjunto de 11 lições aprendidas:

LA52. Estabeleça um forte relacionamento com desenvolvedores influentes locais chamados de influenciadores. Estas são as pessoas que se levantarão durante uma reunião e proporão seu produto como a solução para o problema que estão tentando resolver. Eles fazem parte do meu sistema de apoio. Um influenciador é alguém com quem você criou um relacionamento especialmente profundo e significativo, que você delegou para sair, agir em seu nome e aumentar seu alcance;

LA53. Invista em eventos, preferencialmente organizados pelos influenciadores, garantindo eles compreendam que afetam os objetivos da empresa e o que eles podem fazer para causar impacto positivo. Mantenha sua equipe de influenciadores informadas os eventos que você está planejando;

LA54. Construa uma rede forte entre os diferentes influenciadores da comunidade, uma comunidade de influenciadores. Então, reconheça-as como comunidades oficiais de especialistas e conceda-lhes privilégios expandidos. Isso ajudará a aumentar a comunidade e diminuir a carga nas equipes DevRel;

- LA55. Apresente a sua comunidade de influenciadores aos desenvolvedores e empresas parceiras. Informe aos diversos setores de sua organização sobre seus influenciadores - este é o passo final para transformar os voluntários em colegas;
- LA56. Defina um manual de eventos – como o influenciador de alguma forma fala do nome da sua empresa usando seus materiais, eles também devem ter metas a serem alcançadas;
- LA57. Peça aos influenciadores para submeter palestras para conferências técnicas para que eles entreguem palestras técnicas e ajudem na divulgação do ecossistema;
- LA58. Envolve os influenciadores na codificação de demonstrações técnicas de produtos do ecossistema. Isto ajuda na capacitação do influenciador;
- LA59. Ensine os influenciadores a criar artigos técnicos que informam, convencem e/ou estabelecem a credibilidade do autor;
- LA60. Peça aos influenciadores que planejem e realizem reuniões com as comunidades locais, além de analisar os resultados do engajamento de encontro;
- LA61. Oriente os influenciadores para serem ativos em sites como o GitHub, o Medium e o YouTube;
- LA62. Direcione o influenciador a coletar feedback sobre as tendências de seus produtos.

5.2 Conclusões e Implicações do DEVGO

O DevGo descreve como as relações ganha-ganha entre os desenvolvedores e uma organização central podem favorecer a sustentabilidade do ecossistema por meio dos desenvolvedores e da organização, representada pela equipe de profissionais de DevRel. Esse cenário ganha-ganha é percebido no DevGo por meio do loop de feedback envolvendo repositórios de dados dentro da área de foco “Monitoramento” que contêm a “voz dos desenvolvedores” e os objetivos da organização central. O uso de repositórios de dados no modelo pode formar a base sob mecanismos e ferramentas de avaliação maduros para grandes ecossistemas.

O DevGo pode suportar demandas do setor, tais como: ROI (Retorno sobre o Investimento, do inglês *Return On Investment*) em atividades de relações com desenvolvedores; efeitos na experiência do desenvolvedor; e identificação de barreiras para a participação do desenvolvedor. O uso do DEVGO também pode contribuir para ampliar os esforços nas equipes de DevRel. O uso do modelo do DEVGO pode ajudar quando:

1. **Não há programa de DevRel:** conhecimento das áreas essenciais para planejar e iniciar um programa de governança de desenvolvedores por meio de DevRel;
2. **Estratégias de DevRel em nível básico, documentado e definido):** identificação de onde a organização está e onde ela pode ir;
3. **Estratégias próprias de DevRel:** guia em decisões envolvendo riscos e tendências para manter a competitividade em relação a outras organizações;
4. **Monitoramento e controle:** uso mais focado na área de Monitoramento para formar uma base sólida de mecanismos de avaliação e ferramentas para MSECOS com base mais extensa de desenvolvedores;
5. **Introdução de estratégias inovadoras para melhor atender às metas da organização:** apoio às organizações na evolução das estratégias de governança do desenvolvedor, dimensionar esforços e formar equipes internas.

CAPÍTULO 6 – ESTUDO DE CASO: RETENÇÃO DE DESENVOLVEDORES

Neste capítulo são descritos o planejamento, execução e análise de resultados de um estudo de caso no cenário de retenção de desenvolvedores. O cenário é descrito dentro de MSECO e com o envolvimento de profissionais de DevRel. Enquanto a intervenção no caso, o DevGo é validado em relação a adequação, controle, entendimento e generalidade para a governança de desenvolvedores em MSECO do ponto de vista da retenção.

6.1 A retenção de desenvolvedores em MSECO por profissionais de DevRel

O objetivo do estudo de caso nesta pesquisa é confirmatório, ou seja, é um tipo de estudo de caso que justifica uma intervenção em algum cenário (Runeson e Host, 2009). Para executar atividades que ajudem em investigações da prática é necessário estabelecer um planejamento que irá guiar a execução dessas atividades. Isto é importante para que as análises dos resultados reflitam exatamente o que está sendo observado e levantado nos estudos de caso (Yin, 2009). O caso é relatado a seguir.

O **caso analisado** neste estudo se refere à **retenção de desenvolvedores externos a uma organização central dentro de um MSECO**. O estágio de retenção, como descrito no Capítulo 4, é um estágio relacionado ao envolvimento contínuo do desenvolvedor dentro do ecossistema. Os desenvolvedores já são ativos no ecossistema, ou seja, já submeteram e tiveram aprovada alguma contribuição para a plataforma. Essa contribuição pode ser, por exemplo, um artigo técnico sobre algum produto do MSECO, uma aplicação móvel, uma resolução de defeito relatado no Github.

A perspectiva de análise do caso é do ponto de vista dos profissionais de DevRel (do inglês *Developer Relations*), que desejam utilizar o modelo DevGo (intervenção) para governar desenvolvedores em MSECO com foco no estágio de retenção dentro de um MSECO. Desta forma, nas próximas seções são apresentados o propósito, a perspectiva, o objetivo, as questões e métricas, o planejamento do estudo, a instrumentação e a seleção dos participantes.

6.1.1 Objetivo

O objetivo deste estudo, baseado na abordagem GQM [Basili 1994], é **analisar** o modelo DevGo **com o propósito de avaliar com respeito** aos critérios de adequação, controle, entendimento e generalidade para a governança de desenvolvedores em MSECO

do ponto de vista dos profissionais no contexto de atividades de DevRel voltadas a retenção em governança de desenvolvedores.

6.1.2 Questão de Pesquisa do Estudo de Caso

O propósito deste estudo foi responder à questão: *“A utilização do modelo DevGo por profissionais de DevRel para governar desenvolvedores, durante o estágio de retenção, no contexto de Ecossistema de Software Móvel (MSECO) é adequada, controlável, entendível e generalizável?”*

6.1.3 Questões de Pesquisa e Métricas

As questões para a análise do modelo DevGo foram baseadas em um conjunto de critérios definidos por Strauss e Corbin (1998) para garantir objetividade durante a avaliação de teorias substantivas fundamentadas em dados. A seguir, para cada critério são apresentadas as questões e métricas associadas a eles. As métricas são coletadas a partir das respostas dos participantes. As respostas contêm as percepções dos profissionais de DevRel.

- **Questão 1 – Adequação:** Qual o nível de conformidade do DevGo para a governança de desenvolvedores em MSECO?
 - **Métricas:** (1) Percentual de concordância e discordância, em nível parcial ou total, para a adequação de áreas de foco, das fases, dos estágios, dos objetos de transferência de valor e das lições aprendidas. (2) Percentual de elementos do modelo e de lições aprendidas que têm sido abordados por profissionais de DevRel (lista consolidada a partir do conjunto inicial).
- **Questão 2 – Controle:** O DevGo serve como guia para um profissional de DevRel monitorar e intervir durante a governança de desenvolvedores em MSECO?
 - **Métricas:** (1) Percentual de concordância e discordância, em nível parcial ou total, em relação a percepção de melhoria de desempenho, produtividade e eficácia de profissionais de DevRel. (2) Conjunto de benefícios e limitações do DevGo identificadas pelos profissionais de DevRel.
- **Questão 3 – Entendimento:** O profissional de DevRel consegue utilizar o DevGo para lidar com a governança de desenvolvedores em MSECO?
 - **Métricas:** Percentual de concordância e discordância, em nível parcial ou total, para a aplicabilidade, utilidade, clareza, compreensão do DevGo, inclusive, a relação com a realidade vivida por profissionais.
- **Questão 4 – Generalidade:** O DevGo serve como guia para um grupo de profissionais de DevRel e outros ECOS sem perder sua relevância?

- **Métricas:** Percentual de concordância e discordância, em nível parcial ou total, em relação a relevância do modelo para: (1) outros ECOS, (2) outros profissionais de DevRel e (3) para empresas de pequeno, médio e grande porte.

6.1.4 Planejamento do Estudo

Este estudo objetivou a governança de desenvolvedores por profissionais de DevRel num contexto real de MSECO, neste caso os MSECOS onde os profissionais estão envolvidos, por meio da estrutura de governança de desenvolvedores e as lições aprendidas que compõem o DevGo.

Com os resultados obtidos por meio da execução deste estudo é possível avaliar o modelo proposto em relação a visão e perspectiva dos profissionais de DevRel para a governança de desenvolvedores em MSECO por meio da definição dos elementos gerais do modelo incluindo as lições aprendidas.

O método para a coleta de dados utilizado foi a entrevista. A entrevista é uma conversa guiada, onde o entrevistador pode direcionar a conversa segundo a necessidade da investigação (Yin, 2009). Em relação às entrevistas, foram utilizados questionários semiestruturados envolvendo questões fechadas e questões abertas. Em relação ao método de análise de dados, utilizaram-se procedimentos de análise temática (Cruzes e Dyba, 2011).

6.1.4.1 Instrumentação

Foi proposto um questionário aos participantes com o objetivo de caracterizar a sua formação no que se refere à sua experiência, aos ecossistemas nos quais atua/atuou, o tamanho da organização que trabalha, além do Termo de Consentimento Livre e Esclarecido – TCLE. O estudo de caso e a entrevista seguem o roteiro descrito na Seção 6.1.5 que trata sobre a execução do estudo. As questões estão disponíveis no **Apêndice E** e no link: <https://forms.gle/FydnkGAtCkxeDQyE7>.

Como instrumento para guiar o uso pelo participante foi construído um conjunto de slides para compor uma apresentação com o modelo DevGo. Este instrumento ajudou a guiar o participante no uso do DevGo durante o estudo de caso.

O planejamento, assim como, os instrumentos deste estudo de caso, passaram por etapas de revisão envolvendo quatro pesquisadores com experiência em Engenharia de Software e Engenharia de Software Experimental. Foram ainda realizadas duas rodadas de estudo piloto para refinar o planejamento e os instrumentos antes da realização do estudo com os profissionais de DevRel.

6.1.4.2 Participantes

Os cinco profissionais de DevRel participantes deste estudo possuem experiência nos seguintes MSECOS: Android, Windows, iOS, XNA e em outros tipos de ECOS: Microsoft Azure. Eles, enquanto profissionais de DevRel, já atuaram/atua nos seguintes papéis: evangelista, *developer advocate*, líder de DevRel, gerente de DevRel. Na **Tabela 27** são descritas as características da empresa e dos profissionais.

Tabela 27. Caracterização dos Participantes.

Id	Anos de Experiência	MSECO	Organização	Papeis	Treinamentos	Eventos
P1	4	Windows e XNA	100 – 999 funcionários	Evangelista	15	5
P2	6	Android e iOS	Acima de 1000 funcionários	<i>Developer Advocate</i> e Líder de DevRel	50	10
P3	7	Android, iOS, Symbian, Unity e Windows	Acima de 1000 funcionários	Gerente de DevRel	40	11
P4	9	Android, iOS, Microsoft Azure, Symbian, Unity e Windows	Acima de 1000 funcionários	Evangelista, Especialista em Desenvolvimento de Parceiros, Gerente de DevRel	100	200

6.1.5 Execução do Estudo

Para a execução deste estudo foi definido um roteiro que consiste nas seguintes fases: (1) Contextualização; (2) Caracterização de Perfil; (3) Entendimento da Estratégia; (4) Utilização do modelo DevGo; (5) Entrevista com o Participante; (6) Finalização. Os passos para cada uma das fases são descritos no **Apêndice E**.

6.1.6 Ameaças à Validade

Relacionada ao risco de outros fatores não identificados a priori, terem influenciado um eventual relacionamento de causalidade entre tratamento e resultado, sem o conhecimento prévio do experimentador.

- **Instrumentação:** os instrumentos adequados foram utilizados para todos participantes, desde o roteiro do estudo, o questionário e o instrumento contendo o DevGo. Todos passaram por revisão por pares e foram submetidos a dois estudos piloto. A medição dos resultados de questionários foi feita por meio de contagem das respostas. Os resultados envolvendo elementos selecionados para

retenção foram obtidos diretamente do instrumento que o participante respondeu com sua seleção;

- **Seleção:** como mencionado na Seção 6.1.4.2, os participantes não foram selecionados de maneira aleatória, levou-se em consideração o perfil técnico e a experiência em DevRel;
- **Maduração:** para não houvesse diferença entre a capacidade dos participantes em utilizar o instrumento contendo o modelo DevGo, os mesmos instrumentos foram utilizados, todos seguiram um mesmo roteiro e foram submetidos a mesma meta: selecionar elementos do modelo DevGo visando a retenção de desenvolvedores em MSECO;
- **Contaminação:** os participantes foram instruídos a não realizar comunicação sobre o modelo DevGo até que o estudo fosse considerado concluído;
- **Efeito da expectativa do sujeito:** pela comunicação de que os participantes não estavam sendo avaliados de forma individual e nem que suas estratégias estariam sendo avaliadas em relação a corretude;
- **Efeito da expectativa do experimentador:** para que as crenças do pesquisador não causassem um efeito no sujeito ou nas atividades realizadas por ele, o pesquisador também responsável pela pesquisa seguiu fielmente o roteiro do estudo descrito no *Apêndice E*.

6.1.6.1 Validade Externa

Questões relacionadas à ameaça dos resultados do estudo não serem generalizáveis a projetos reais na indústria. As principais ameaças identificadas foram:

- **Participantes:** foram selecionados participantes, como mencionado na Seção 6.1.4.2, que possuem relação e refletem o comportamento da população de profissionais de DevRel;
- **Tempo:** o participante foi comunicado do tempo médio que deveria dedicar para a participação no estudo. Com isto, reduziu-se a ameaça de, com um tempo reduzido, o participante não analisasse com cuidado os elementos do modelo DevGo;
- **Configuração do experimento:** o caso de estudo analisado caracteriza um ambiente real, pois os participantes estão atuando na área de DevRel e MSECO, logo os elementos foram selecionados em cima da realidade que eles já viveram e ainda vivem.

6.1.6.2 Validade de Construção

Problemas relacionados à ameaça de generalizar os resultados do estudo à teoria que o sustenta. As principais ameaças são:

- **Projeto do Experimento:** o modelo DevGo passou por diversos estudos experimentais para sua concepção e refinamentos. Os instrumentos de todos estudos passaram por revisão e estudos piloto, assim como, o roteiro do estudo de caso e entrevistas para avaliação do DevGo;
- **Fatores humanos (ou sociais):** os participantes não estavam envolvidos em outros experimentos durante a realização deste estudo de caso. Os participantes ainda participaram de forma espontânea como forma de adquirir conhecimento sobre como o modelo DevGo poderia ajudá-los na estratégia de retenção de desenvolvedores.

6.1.6.3 Validade de Conclusão

Relacionada a questões que ameaçam a habilidade de traçar conclusões corretas sobre o relacionamento entre tratamentos e resultados de um estudo experimental:

- **Análise e interpretação do resultado:** realizaram-se os passos necessários para a análise quantitativa, por meio da coleta direta das respostas por formulário on-line, e para a análise qualitativa, agrupamento dos comentários e identificação e aspectos comuns levantados pelos participantes;
- **Confiabilidade das medidas:** os dados foram obtidos diretamente dos instrumentos preenchidos pelos participantes e do formulário online.

6.1.7 Análise Quantitativa

Nas **Figuras 22, 23, 24 e 25** são apresentadas as porcentagens em níveis de concordância, neutralidade ou discordância para cada um dos critérios (adequação, controle, entendimento e generalidade) avaliados de forma quantitativa.

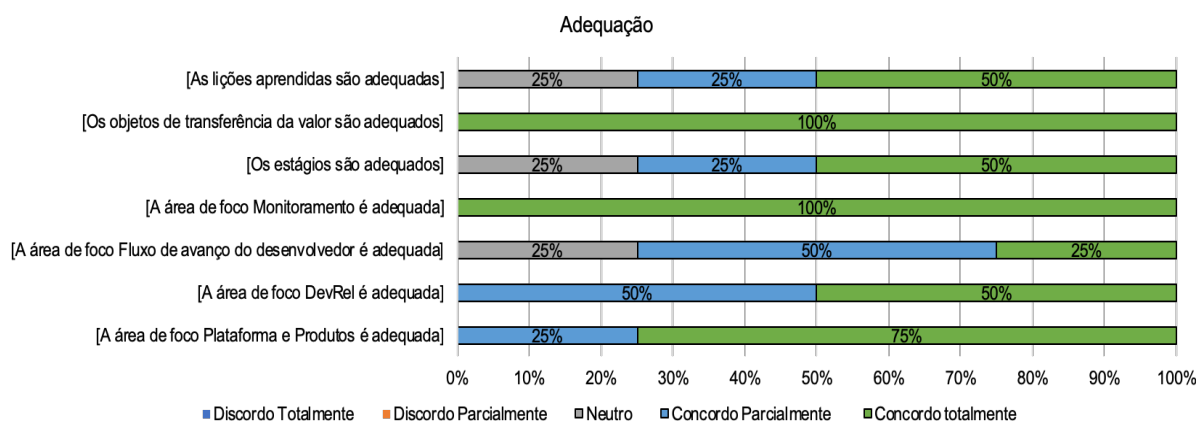


Figura 22. Adequação do DevGo.

Em relação a **adequação**, ou seja, a conformidade do DevGo para a governança de desenvolvedores em MSECO, do ponto de vista de retenção. Todos participantes concordaram, parcialmente ou totalmente, que os objetos de transferência de valor e as áreas de foco de monitoramento, DevRel e plataforma e produtos são adequados para a retenção de desenvolvedores. Em relação a adequação das lições aprendidas, do estágio de retenção, somente um participante (P1) se manteve neutro. Segundo este participante a LA41 é aderente ao estágio de reconhecimento.

Quando o aspecto analisado é o **controle**, todos participantes concordam totalmente que o DevGo pode melhorar, a eficácia, a produtividade e o desempenho dos profissionais de DevRel em estratégias que visem a retenção de desenvolvedores. Houve nível neutro (25%) e de concordância parcial, (25%) quando o profissional foi questionado se o DevGo tornaria mais fácil as atividades de DevRel.

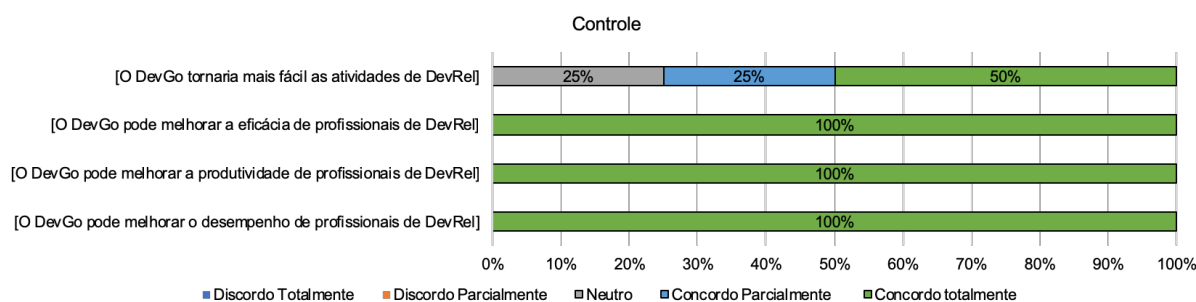


Figura 23. Controle do DevGo.

Dois dos participantes (P1 e P2) comentaram que DevRel e a governança de desenvolvedores é uma área complexa. Por mais que o DevGo os ajude, a área nunca se tornará simples devido a dinâmica do MSECO. Com disse o P2: “... *acredito que sempre responderia parcialmente por isso. Pois esta área sempre será complexa e exigirá muitas habilidades técnicas, sociais e de negociação do profissional*”.

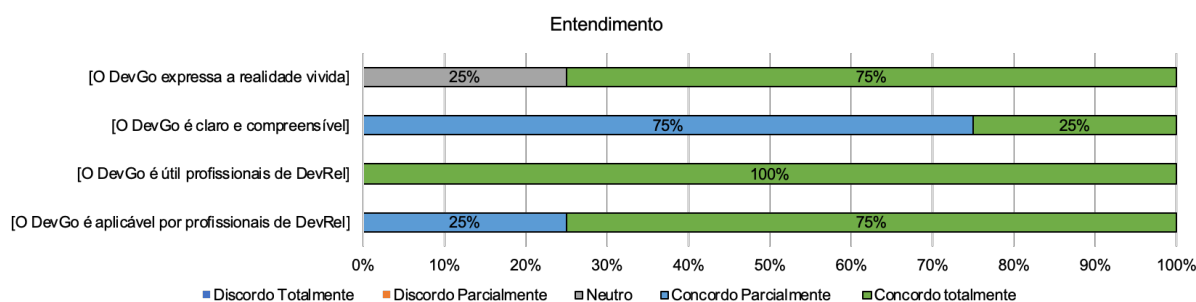


Figura 24. Entendimento do DevGo.

O **entendimento** captura o quanto o profissional de DevRel consegue utilizar o DevGo para a governança de desenvolvedores visando a retenção. Todos os participantes concordaram totalmente que o DevGo é aplicável por profissionais. Três dos participantes (75%) concordaram parcialmente que o modelo é compreensível e claro. Sobre isto, o

participante P3 levantou a necessidade da existência de um guia no uso do DevGo: “você pode criar um guia para ser utilizado sem a sua condução e que leve em consideração o nível de experiência de cada profissional de DevRel”.

Quando a análise era se o DevGo expressava a realidade vivida, somente um participante, o P4, manteve-se neutro. Ele destacou que o estágio de retenção deve criar a dependência do desenvolvedor em relação aos produtos do MSECO. Isto ajuda a reter: “falta no estágio de retenção algo que descreva a necessidade de ‘amarrar’ o desenvolvedor a conhecimento adquirido, investimento financeiro feito e participação em projetos open-source. Quanto mais ‘amarrado’ mais difícil se torna a saída dele.”

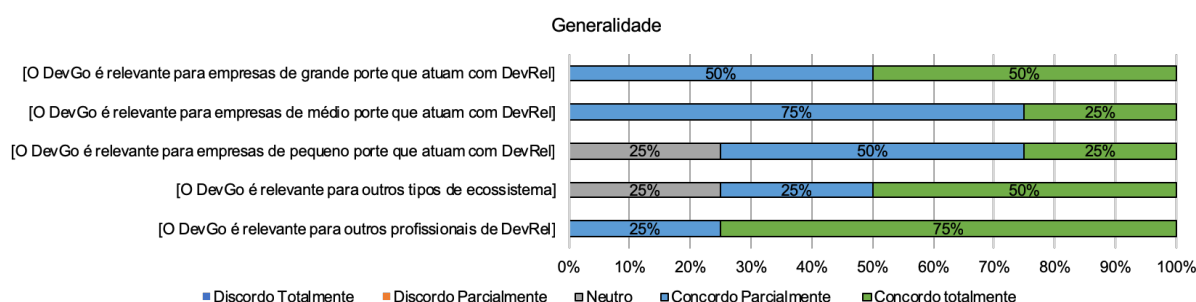


Figura 25. Generalidade do DevGo.

A **Figura 25** cobre o critério de generalidade, ou seja, se o DevGo serve de guia para um conjunto de profissionais de DevRel e outros ECOS sem perder a relevância. Neste sentido, todos os participantes concordam em nível parcial ou total que o DevGo é relevante para: outros profissionais de DevRel, para empresas de médio e grande porte. Quando se trata de empresas de pequeno porte, um foi neutro e dois (50%) participantes concordaram parcialmente.

Um dos participantes P4, comentou que as empresas de pequeno porte focam muito mais na expansão da plataforma, ou seja, quantidade e não na qualidade: “As empresas de pequeno porte focam em quantidade, pra mostrar que têm base de desenvolvedores, gente produzindo, pra até analisar a robustez da plataforma. Isso está mais relacionado a fase de início”.

O participante P2, alertou para o nível de abertura de outros tipos ECOS que pode fazer com que a área de monitoramento seja impactada: “Sobre ser relevante para outros ecossistemas os muito fechados não usam muito de repositórios, então, talvez o monitoramento deveria ser diferente. Acho que pra open source ou mobile, cloud, IOT ... se encaixa mais”.

6.1.8 Análise Qualitativa

A primeira parte do estudo, logo após a contextualização e caracterização do perfil do participante, consiste no entendimento da estratégia utilizada pelo profissional de DevRel.

6.1.8.1 Estratégia dos Participantes

A partir dos comentários dos participantes, foram identificados os seguintes elementos constituintes para a estratégia de retenção de desenvolvedores que eles seguem: (1) objetivo da organização; (2) tipo de contato com o desenvolvedor; (3) uso de embaixadores; (4) adequação de facilitadores; (5) funil de engajamento; (6) recursos para criação de valor; (7) repositórios.

Quanto ao **objetivo da organização**, os participantes destacaram o entendimento a fundo do que a organização precisa dos desenvolvedores como o ponto principal para direcionar a estratégia de governança visando a retenção de desenvolvedores. Enquanto o participante P3 destaca a retenção de desenvolvedores com o objetivo principal de se relacionar com desenvolvedores que possuem contribuições potenciais para o MSECO.

“Depende do objetivo a ser cumprido: quantidade de apps publicados ou qualidade (medido pelo número de downloads ou estrelas na loja).” – P1

“Uma estratégia de relacionamento mais próximos com os desenvolvedores ativados com potências soluções”. – P3

No que está relacionado **ao tipo de contato com os desenvolvedores** é preciso analisar o que é importante no momento, o contato presencial ou a distância. Assim como, as regiões de atuação. A equipe de DevRel de P2 é separada para atuar por regiões.

“Outro fator usado para a estratégia seria a relação com o dev (a distância ou presencial).” – P1

“Peço informação da gerência de qual é o foco das ações para o semestre, inclusive, as possíveis localidades que atuarei. Porque a nossa equipe é separada por regiões.” – P2

O contato presencial pode ser, também, conduzido por um **embaixador** que é membro da equipe do profissional de DevRel. E, este embaixador, um desenvolvedor com nível elevado de colaboração para o MSECO, poderia ser gerenciado a distância. Isto ajudaria a escalar a ação de governança de desenvolvedores.

“Outro ponto que uso muito são meus embaixadores, aí eles reportam as atividades pra mim. Aqui tem ajudado muito. Tenho formado equipes abaixo dos embaixadores. Aí consigo focar numa retenção mais de advocate.” – P2

“...além da aproximação de desenvolvedores com alto grau de colaboração.” – P3

Tendo claro os objetivos da organização e o tipo de contato com os desenvolvedores, a outra parte comum da estratégia dos participantes é **adequar o uso dos facilitadores** a partir das informações de objetivos e tipo de contato. Sempre como forma de retornar valor a organização que fez investimento por meio de orçamento para a área de DevRel. O participante P3 indicou a utilização de facilitadores como *coach* técnico que ajude na orientação do desenvolvedor na seleção por funcionalidades chave e na publicação de contribuições: *“Essa estratégia é amparada basicamente em coach técnico, orientação na seleção de key features, apoio a publicação”*.

O participante P4 assim como P3, comentou sobre a análise dos desenvolvedores a partir de um **funil de engajamento**. Um outro aspecto indicado por P4 como parte da estratégia dele de retenção envolve **recursos para a criação de valor** para os desenvolvedores: conteúdo técnico, novas versões de produtos existentes e novos produtos.

“Estabelecer o funil de engajamento - awareness, experiential, adoption. Para reter, é importante fazer nurture (alimentação) de conteúdo técnico, e ter uma plataforma que tenha recursos que evoluam (novas versões de produtos existentes e novos produtos)” – P4

O que difere dos facilitadores no contato presencial ou remoto é, respectivamente: no presencial, as metas dos profissionais de DevRel já devem ser alcançadas logo após o término do evento; no remoto, os participantes indicam o estabelecimento de desafios/metast mensais para os desenvolvedores que, se alcançadas, os desenvolvedores serão premiados.

“Para treinamentos presenciais, eu separava parte do treinamento para focar nos objetivos, realizando mini-hackathons nos finais de cada aula e premiando os devs com brindes (camisas, bonés, adesivos, etc). Para os devs com contato remoto, eu geralmente estabelecia desafios mensais, premiando os devs que conseguiam atingir os objetivos. As premiações para devs distantes eram pontuações em programas de desenvolvedor.” – P1

“...e entro em contato com esses devs. Aqui é um trabalho que envolve reuniões por Skype, ou quando vou na cidade aproveito, também o Slack ou um github privado.” – P2

Em relação **a repositórios** o participante P2, antes de ir para o contato com os desenvolvedores analisa a situação de quantidade e qualidade de apps pelo painel de controle do desenvolvedor: *“Dentro da minha região faço a análise pelo dashboard de devs como está a situação de quantidade de apps e de qualidade pelos ratings”*. E, ainda utiliza a

loja de apps, para analisar quais delas precisam ser “curadas”, ou seja, adição de funcionalidades e diferenciação no mercado: *“Para escalar minha ação eu analiso a loja as apps que precisam ser ‘curadas’, que precisam de funcionalidades novas pra se diferenciar...”*.

Em resumo, os participantes, em cima dos objetivos da organização central e do tipo de contato com o desenvolvedor, analisam quais os mecanismos permitem a ação com os desenvolvedores. Estes podem ser analisados a partir de um funil de engajamento. Um dos participantes analisa o “estado” dos desenvolvedores antes de ir para a ação. O DevGo poderia ajudar os participantes a organizar melhor a seleção de objetivos, de que tipo de profissional de DevRel deveria atuar dependendo do objetivo e quais facilitadores (com um conjunto maior de opções) poderiam ser utilizados. Informações de outros repositórios poderiam ser utilizados para dar informações mais precisas sobre os desenvolvedores. As estratégias informadas pelos participantes, em sua maioria, exigem o contato do profissional de DevRel de forma presencial. O DevGo já os ajudaria a escalar as estratégias.

6.1.8.2 Estratégia dos Participantes a partir do DevGo

Nesta seção é discutida a seleção de elementos para cada área de foco visando a retenção de desenvolvedores. Os elementos selecionados pelos participantes são apresentados nas **Tabelas 28, 29, 30, 31, 32 e 35**. Os elementos que foram selecionados por dois participantes tem associado o ícone amarelo. Aqueles que obtiveram a indicação de 3 ou 4 profissionais são marcados com o símbolo verde. Os que foram marcados somente uma vez ou nenhuma com o sinal vermelho. De forma geral, considerando 192 elementos possíveis de serem selecionados para retenção, 14 (7%) não foram selecionados pelos participantes. Desta forma, 178 (93%) elementos foram selecionados, pelo menos, uma vez.

Vinte e nove dos elementos (15%) foram indicados para retenção por um dos participantes. A metade dos participantes selecionou um conjunto de cinquenta e oito elementos (30%). Todos os participantes, quando analisado o conjunto comum de elementos selecionados, concordaram com trinta dos elementos para a retenção, isto equivale a 16% do conjunto inicial.

O conjunto de lições aprendidas de (LA31 a LA40) foi considerado na contagem acima dentro da área de foco “Fluxo de Avanço do Desenvolvedor”. Na **Tabela 33** é apresentada a contagem para cada lição aprendida relacionada a retenção. Todas as lições aprendidas foram selecionadas, pelo menos uma vez, pelos participantes do estudo. A LA34, que trata da presença do profissional de DevRel no Stack Overflow para uma presença mais ativa e aproximada com os desenvolvedores, foi selecionada por todos os participantes do estudo.

Tabela 28. Seleção de elementos da área de foco Plataforma e Produtos.

		P1	P2	P3	P4		
OBJETIVO	Fornecer informações e recursos que apoiem as metas de uma organização em relação à produtividade;	✓	✓	✗	✗	⚠	
	Fornecer informações e recursos que apoiem as metas de uma organização em relação à criação de nicho;	✓	✓	✗	✗	⚠	
	Fornecer informações e recursos que apoiem as metas de uma organização em relação à qualidade das contribuições.	✓	✓	✓	✓	✓	
COMPONENTE	Plataforma	✓	✗	✓	✓	✓	
	Infraestrutura	✗	✓	✗	✓	⚠	
	Orçamento	✗	✗	✗	✓	✗	
	Produtos (ex: APIs, SDKs, IDEs)	✓	✓	✓	✓	✓	
	Serviços	✗	✓	✗	✗	✗	
Objetos de Transferência de Valor - Consome de	DevRel (Evangelismo e Advocacia)	Credibilidade dos profissionais de DevRel	✓	✓	✓	✗	✓
		Recursos desejados por desenvolvedores	✓	✓	✓	✓	✓
		Conscientização sobre produtos	✗	✗	✓	✗	✗
		Articulador do fluxo de informação	✓	✗	✗	✓	⚠
		Área da estrutura organizacional	✗	✗	✗	✗	✗
		Concentração em custos	✗	✗	✗	✗	✗
		Maturidade da organização	✗	✓	✗	✗	✗
		Serviços	✓	✓	✗	✗	⚠
		Produtos complementares	✗	✓	✗	✗	✗
		Inovações	✓	✓	✓	✓	✓
	Fluxo	Probabilidade de recomendação	✓	✓	✓	✓	✓
		Receita por meio de serviços	✓	✓	✗	✗	⚠
		Reconhecimento da marca	✓	✗	✓	✓	✓
		Investimento financeiro dos desenvolvedores	✗	✗	✗	✗	✗
	Monitoramento	Escala e tamanho das comunidades	✗	✓	✗	✗	✗
		Reconhecimento da marca	✗	✗	✗	✓	✗
		Dados gerados por uso de produtos e serviços	✗	✓	✓	✓	✓
		Dados sobre a interação dos desenvolvedores e contribuições	✓	✓	✗	✗	⚠
Objetos de Transferência de Valor - Provê para	DevRel (Evangelismo e Advocacia)	Programas de acesso	✓	✓	✗	✓	✓
		Informações precisas sobre o roadmap	✓	✓	✓	✓	✓
		Visão e expectativas	✓	✓	✓	✗	✓
		Infraestrutura	✓	✓	✗	✓	✓
		Objetivos da organização central	✓	✓	✓	✗	✓
	Fluxo	Robustez do ecossistema	✓	✗	✗	✗	✗
		Novos produtos	✓	✓	✓	✗	✓
	Monitoramento	Padrões e incentivos	✓	✓	✗	✓	✓
		Produtos com código aberto	✓	✓	✓	✓	✓
		Visão e Expectativas	✓	✓	✓	✗	✓

Tabela 29. Seleção de elementos da área de foco DevRel (Evangelismo e Advocacia) – Parte 1.

		P1	P2	P3	P4	
OBJETIVO	Ajudar a incorporar contribuições com potencial (ou seja, produtos complementares, serviços e inovações) que surgem dos desenvolvedores para a plataforma do ecossistema na área de foco da Plataforma e Produtos;	✓	✓	✓	✗	✓
	Manter o equilíbrio entre as expectativas dos desenvolvedores e as necessidades da organização central;	✓	✗	✗	✗	✗
	Balancear o roadmap interno da organização com as necessidades dos desenvolvedores;	✓	✓	✓	✓	✓
	Construir confiança entre setores da organização e os desenvolvedores.	✓	✗	✗	✓	⚠
COMPONENTE	Evangelismo: divulgação da “palavra” da organização, ou seja, na prospecção de desenvolvedores, na influência de desenvolvedores externos. A parte de evangelismo está associada aos estágios de sensibilização, entrada e ativação do DevGo	✓	✓	✗	✓	✓
	Advocacia: trabalha com desenvolvedores existentes, ou seja, com o ganho de interesse e a fidelização de potenciais desenvolvedores. Isto se dá por meio de conteúdo de qualidade e específico para esses desenvolvedores. A parte de advocacia está relacionada aos estágios de retenção, reconhecimento e referência dentro do DevGo	✓	✓	✓	✓	✓
Objetos de Transferência de Valor - Consome de	Plataforma e Produtos					
	Programas de Acesso	✓	✓	✗	✗	⚠
	Informações precisas sobre o roadmap	✓	✓	✓	✗	✓
	Visão e expectativas	✓	✗	✓	✗	⚠
	Infraestrutura	✓	✓	✗	✓	✓
	Objetivos da organização central	✗	✓	✓	✗	⚠
	Fluxo					
	Comunidade de especialistas	✓	✓	✗	✓	✓
	Interação entre desenvolvedores experientes e parceiros da indústria	✓	✗	✓	✓	✓
	Envolvimento em conjunto	✓	✗	✗	✓	⚠
	Recursos desejados	✓	✓	✗	✓	✓
	Desenvolvedores ativos	✓	✓	✗	✗	⚠
	Combinação de tecnologias	✓	✓	✗	✓	✓
	Novos e competitivos produtos	✓	✓	✗	✗	⚠
	Tecnologias e relacionamentos	✓	✓	✓	✗	✓
	Monitoramento					
	Ação mais ampla	✗	✓	✗	✗	✗
	Recursos desejados	✓	✓	✗	✓	✓
	Produtos que são destaque	✓	✓	✗	✗	⚠
	Dados gerados por uso de produtos e serviços	✓	✓	✓	✓	✓
	Dados sobre a interação dos desenvolvedores e contribuições	✓	✓	✗	✓	✓

Tabela 30. Seleção de elementos da área de foco DevRel (Evangelismo e Advocacia) – Parte 2.

		P1	P2	P3	P4		
Objetos de Transferência de Valor - Provê para	Plataforma e Produtos	Credibilidade dos profissionais de DevRel	✓	✓	✗	✗	⚠
		Recursos desejados por desenvolvedores	✓	✓	✗	✓	✓
		Conscientização sobre produtos	✗	✓	✓	✗	⚠
		Articulador do fluxo de informação	✓	✗	✗	✗	✓
		Área da estrutura organizacional	✗	✗	✗	✗	✗
		Concentração em custos	✗	✓	✗	✗	✗
		Maturidade da organização	✓	✓	✗	✓	✓
		Serviços	✓	✓	✗	✗	⚠
		Produtos complementares	✗	✓	✗	✗	✗
		Inovações	✓	✓	✓	✓	✓
	Fluxo	Bom relacionamento	✓	✓	✗	✓	✓
		Brindes	✓	✓	✗	✓	✓
		Reconhecimento do desenvolvedor	✓	✗	✓	✓	✓
		Programas de acesso	✓	✓	✗	✓	✓
		Contato presencial e por meio da web	✓	✓	✓	✗	✓
		Envolvimento em conjunto	✗	✗	✗	✗	✗
		Capacitação técnica	✓	✓	✗	✓	✓
		Conscientização sobre produtos	✓	✓	✗	✗	⚠
		Visão e expectativas da organização	✗	✗	✗	✗	✗
		Conteúdo gratuito	✓	✓	✗	✗	⚠
		Novas oportunidades	✓	✓	✗	✗	⚠
		Ações diversificadas	✓	✓	✗	✗	⚠
		Conexão escalável a produtos	✓	✓	✓	✓	✓
		Recursos e ferramentas apropriadas	✓	✓	✗	✗	⚠
		Monitoramento	Recursos técnicos e não-técnicos	✓	✓	✗	✓
	Conteúdo acessível		✓	✓	✗	✗	⚠
	Conteúdo gratuito		✓	✓	✗	✗	⚠
	Conscientização sobre produtos		✓	✗	✓	✗	⚠

Tabela 31. Seleção de elementos da área de foco Fluxo de Avanço do Desenvolvedor – Parte 1.

		P1	P2	P3	P4	
Objetivo da Área	Apoiar no acompanhamento e análise do avanço dos desenvolvedores dentro do ecossistema;	✓	✗	✗	✓	⚠
	Entender como os desenvolvedores estão se movimentando e gerando contribuições dentro do ecossistema é importante para uma organização direcionar suas estratégias de atuação para governá-los.	✓	✓	✓	✓	✓
Objetivo do Estágio	Valorizar o desenvolvedor para retê-lo em relação a monetização e oportunidades de benefícios e a própria cultura dentro do MSECO;	✓	✓	✓	✗	✓
	Fornecer impulso de ganho;	✓	✓	✗	✗	⚠
	Gerenciar mudança de plataforma;	✓	✗	✓	✓	✓
	Apoiar contribuições de nicho;	✓	✓	✗	✗	⚠
	Permitir que os desenvolvedores trabalhem com as mais recentes e melhores tecnologias do ecossistema;	✓	✓	✓	✗	✓
	Apoiar o desenvolvimento;	✓	✓	✗	✗	⚠
	Analisar os desenvolvedores periféricos,	✓	✓	✗	✓	✓
	Analisar os desenvolvedores ativos;	✓	✓	✗	✓	✓
	Analisar os desenvolvedores de topo	✗	✗	✗	✓	✗
FACILITADORES	Treinamento	✓	✓	✗	✓	✓
	Portfólio de dispositivos e aplicações móveis;	✗	✓	✗	✗	✗
	Certificação de contribuição;	✗	✗	✓	✗	✗
	Plano de negócios;	✓	✓	✓	✓	✓
	Modelo de geração de renda;	✓	✓	✓	✗	✓
	Acordos de níveis de qualidade e colaboração social.	✗	✗	✗	✗	✗
Objetos de Transferência de Valor - Consome de	Plataforma					
	Novos produtos	✓	✓	✓	✓	✓
	Robustez do ecossistema	✓	✗	✗	✗	✗
	DevRel	Reconhecimento do desenvolvedor	✓	✓	✗	⚠
		Programas de acesso	✓	✓	✓	✓
		Contato presencial	✗	✓	✓	⚠
		Contato por meio da web	✓	✓	✓	✓
		Bom relacionamento	✓	✓	✗	⚠
		Envolvimento em conjunto	✗	✓	✗	✗
		Capacitação técnica	✓	✓	✗	✓
		Conscientização sobre produtos	✓	✗	✗	✗
		Visão e expectativas da organização	✓	✗	✗	✗
		Conteúdo gratuito	✓	✓	✗	⚠
		Novas oportunidades	✓	✓	✗	⚠
		Ações diversificadas	✗	✓	✗	✗
		Brindes	✓	✓	✗	✓
		Conexão escalável a produtos	✗	✓	✗	⚠
		Recursos e ferramentas apropriadas	✓	✓	✓	✓
	Monitoramento	Conteúdo gratuito	✓	✓	✗	⚠
		Dados sobre a interação dos desenvolvedores e contribuições	✗	✓	✗	⚠

Tabela 32. Seleção de elementos da área de foco Fluxo de Avanço do Desenvolvedor – Parte 2.

		P1	P2	P3	P4	
Objetos de Transferência de Valor - Provê para	Plataforma	Probabilidade de recomendação	✓	✓	✓	✓
		Receita por meio de serviços	✓	✓	✓	✓
		Reconhecimento da marca	✓	✓	✓	✓
		Investimento financeiro dos desenvolvedores	✓	✓	✗	✓
	DevRel	Comunidade de especialistas	✓	✓	✓	✓
		Interação entre desenvolvedores experientes e parceiros da indústria	✓	✓	✗	✓
		Envolvimento em conjunto	✗	✗	✗	✗
		Recursos desejados	✗	✓	✗	⚠
		Desenvolvedores ativos	✓	✓	✓	✗
		Combinação de tecnologias	✓	✓	✗	✓
		Novos e competitivos produtos	✗	✓	✓	✗
		Embaixadores	✓	✓	✗	✓
		Tecnologias e relacionamentos	✓	✓	✗	✗
	Fluxo		✓	✓	✓	✓
		Networking	✓	✓	✓	✓
	Monitoramento	Novas contribuições	✓	✓	✗	✗
		Contribuições a produtos existentes por meio de extensões	✓	✓	✗	✗
		Expectativas	✗	✓	✓	✗

Tabela 33. Lições aprendidas indicadas para retenção.

Lição Aprendida	Quantidade (%)	Lição Aprendida	Quantidade (%)
LA34	4 (100%)	LA33	2 (50%)
LA35	3 (75%)	LA40	2 (50%)
LA36	3 (75%)	LA31	1 (25%)
LA37	3 (75%)	LA38	1 (25%)
LA32	2 (50%)	LA39	1 (25%)

O conjunto de *insights* (#1 a #6) relacionadas a categoria de repositório “Perguntas e Respostas” foi considerado dentro da contagem relativa a área de foco “Monitoramento”. Assim como, nas lições aprendidas, todas os insights (**Tabela 34**) foram selecionados pelo menos por um participante do estudo.

Tabela 34. Insights para monitoramento indicadas para retenção.

Insight	Quantidade (%)	Insight	Quantidade (%)
Insight #4	4 (100%)	Insight #2	3 (75%)
Insight #5	4 (100%)	Insight #1	2 (50%)
Insight #6	4 (100%)	Insight #3	1 (25%)

Os insights #4, #5 e #6 foram indicados para retenção por todos participantes. Eles tratam, respectivamente, da identificação de comunidade de especialistas para redução de barreiras, das dúvidas em torno de anúncio oficial de produtos e do uso de medalhas (reputação) para a redução de custos operacionais de governança de desenvolvedores.

Tabela 35. Seleção de elementos da área de foco Monitoramento.

		P1	P2	P3	P4	
OBJETIVO	Monitorar o engajamento do desenvolvedor, dando suporte à transparência tanto para a organização quanto para os desenvolvedores;	✓	✓	✗	✓	✓
	Garantir que todos tenham a chance de entender e fornecer feedback.	✓	✗	✓	✓	✓
CATEGORIAS	Perguntas e Respostas	✓	✓	✓	✓	✓
	Lista de E-mails e Fóruns	✗	✗	✗	✗	✗
	Ambiente Social de Codificação	✓	✗	✓	✓	✓
	Website Social de Notícias	✓	✓	✗	✓	✓
	Redes Sociais	✓	✓	✓	✗	✓
	Comunicação do Time	✗	✗	✓	✗	✗
	Loja de Aplicações	✓	✓	✓	✓	✓
	Painel de Controle do Desenvolvedor	✓	✓	✓	✓	✓
Objetos de Transferência de Valor - Consome de	Plataforma e Produtos	Visão e expectativas	✓	✗	✗	✗
		Produtos com código aberto	✗	✓	✗	⚠
		Padrões e incentivos	✗	✓	✓	✗
	DevRel	Conscientização sobre produtos	✓	✓	✗	✗
		Conteúdo acessível	✓	✓	✗	✗
		Conteúdo gratuito	✓	✓	✗	✗
		Recursos técnicos e não-técnicos	✓	✓	✓	✓
	Fluxo	Expectativas	✓	✓	✗	✗
		Contribuições a produtos existentes por meio de extensões	✓	✓	✓	✓
		Novas contribuições	✓	✓	✗	✗
Objetos de Transferência de Valor - Provê para	Plataforma e Produtos	Escala e tamanho das comunidades	✓	✓	✗	✓
		Reconhecimento da marca	✓	✗	✗	✗
		Dados gerados por uso de produtos e serviços	✗	✓	✓	✓
		Dados sobre a interação dos desenvolvedores e contribuições	✓	✓	✗	✗
	DevRel	Ação mais ampla	✓	✗	✗	✗
		Recursos desejados	✓	✓	✗	✓
		Produtos que são destaque	✓	✓	✗	✗
		Dados gerados por uso de produtos e serviços	✗	✓	✓	✓
		Dados sobre a interação dos desenvolvedores e contribuições	✓	✓	✓	✗
		Dados sobre a interação dos desenvolvedores e contribuições	✓	✗	✓	✓
	Fluxo	Dados sobre a interação dos desenvolvedores e contribuições	✗	✓	✗	✗
		Conteúdo gratuito	✗	✓	✗	✗

6.1.8.3 Benefícios e Limitações do DevGo

Para esta seção, além das respostas direcionadas a capturar os benefícios e as limitações, também, foram analisadas as experiências relatadas com o uso do DevGo e os comentários durante a entrevista, quando os participantes selecionavam elementos do DevGo.

Em relação aos **benefícios**, destacados pelos participantes, estão: (1) a estrutura do modelo; (2) a área de foco de monitoramento; (3) o auxílio a profissionais de DevRel no entendimento da estrutura de governança de desenvolvedores. No que cobre as **limitações** identificadas pelos profissionais: (1) a inserção do DevGo em uma organização; (2) mensurar a efetividade dos facilitadores; (3) ausência de papeis na área de foco “DevRel (Evangelismo e Advocacia); (4) ausência de um objetivo de DevRel com foco em retenção; (5) o nível de maturidade de profissionais de DevRel pra utilizar o DevGo.

Em relação ao benefício (1), ou seja a **estrutura do modelo**, os comentários dos participantes descrevem a notação do modelo como simples e de fácil entendimento, a estrutura das interações, os objetos de transferência de valor, que ajudarão a comunicar a setores da organização o valor do trabalho da área, e os elementos indicados no DevGo que passaram a considerar para suas estratégias de retenção. Todas essas informações ajudam no **auxílio ao profissional de DevRel no entendimento das áreas envolvidas** para governança de desenvolvedores.

“...gostei do modelo DevGo, pois este considera as interações e transferência dos valores entre os diversos atores do processo... Auxilia principalmente os profissionais de DevRel a entender as interações entre as áreas envolvidas.” – P1

“... pensar elementos que não levo em consideração. Acho que me ajudaria. O DevGo me ajudou a pensar uns ajustes na minha estratégia, ainda mais na questão do valor. Porque aqui que o negócio pega. Porque quase todo mundo acha que só viajamos ou fazemos slides... A visão sobre a estrutura que preciso pra pensar estratégias. Os objetos de valor que podem me ajudar a comunicar como contribuo para a empresa, seria legal como calcular ROI a partir disso. Isto sempre é tenso. A relação entre as áreas de foco. A notação do modelo é simples e me ajuda a entender.” – P2

“... foi interessante observar diversos aspectos empíricos formalizados em um Modelo. Visão geral do todo e guia para a execução de trabalho de DevRel.” – P3

“... me levou a estruturar o pensamento sobre a minha ação. Ajudou a, enquanto estava selecionando os elementos, pensar sobre minhas estratégias.” – P4

No que envolve o **monitoramento**, os participantes destacaram os outros tipos de categorias existentes, além do que já utilizam, e a percepção de controle por meio da área de foco de monitoramento:

“O monitoramento me chamou a atenção, acho que alguém da minha equipe poderia ficar focado nisso e avançar na discussão de outros insights. A gente fica muito focado na loja de apps e no github.” – P2

“Controle e monitoramento... Considero útil e poderia ter alguém da equipe focado nisso.” – P4

Um dos participantes levantou a questão de **como o DevGo poderia ser inserido na organização dele**. O problema é que, neste caso, a organização não compreende a importância da área. E que, por isto, o DevGo poderia indicar como a organização deve apoiar a área de DevRel. Não ficou claro se é possível customizar o modelo DevGo para além da seleção de elementos, ou seja, a inserção de novos elementos. E no contexto de uso do DevGo, um participante destacou a importância de **mensurar a efetividade dos facilitadores**.

“Talvez o profissional DevRel passe muito tempo monitorando e entendendo quais são as dúvidas e problemas que a comunidade dev esteja enfrentando, mas será que ações de treinamentos, hackathons e palestras não sejam mais efetivas para mensurar os problemas que a comunidade enfrenta?!” – P1

“Fico pensando como poderia inserir isso na minha organização, pois ela ainda não entende completamente a importância da minha área. Vou tentar pela equipe, primeiramente, seria a partir da área de foco de DevRel. Mas dá pra explorar mais... acredito que o modelo futuramente poderia indicar como a organização poderia suportar DevRel... Seria possível customizar de alguma forma? Adicionar mais algo a medida que vai sendo aplicado? Talvez seja isso. Como eu utilizei para ajustar uma estratégia aqui, o que percebi é se posso customizar.” – P2

Quanto a área de foco “DevRel (Evangelismo e Advocacia)” os comentários dos participantes indicaram **a falta de alguns papéis**: um analista de DevRel focado na área de monitoramento e um engenheiro focado na construção de demonstrações dos produtos. E que o nível de detalhamento possa ser uma dificuldade para um profissional de DevRel que esteja iniciando na carreira.

“Seria bom um dos nossos profissionais só focar em monitorar e informar a equipe. Outro profissional que trabalhamos é o focado em demos dos produtos, assim, a gente foca em outras ações. Isso falta no modelo.” – P1

“... acho que tem que ter um profissional focado em monitoramento: DevRel Analyst??” – P2

“Apresenta um nível de detalhamento que pode ser dificultoso para profissionais de início de carreira.” – P3

“Na minha empresa não temos infraestrutura que nos ajude a monitorar o fluxo de avanço do desenvolvedor.” – P4

Uma limitação percebida pelo participante P4, para o estágio de Retenção descrito no DevGo, é que o **estágio precisa, em sua descrição descrever mais um objetivo**, abordar que o profissional de DevRel deve impulsionar o desenvolvedor a investir em conhecimento, financeiramente em algum produto ou serviço e a se dedicar bastante para a plataforma. Desta forma, como dito pelo participante: *“o desenvolvedor irá criar amarras ao ecossistema gerando dependência. E o desenvolvedor vai pensar muito em abandonar o ecossistema”*.

O participante P3 abordou a questão da falta do modelo de abordar a **maturidade de profissionais** que estão dentro da área de foco “DevRel (Evangelismo e Advocacia): *“Pode expandir, uma vez que tenha estabelecido o modelo geral, a área DevRel para considerar a maturidade do profissional”*.

6.1.9 Conclusões do Estudo

Ao retomar o objetivo deste estudo, que era responder a seguinte questão de pesquisa: *“A utilização do modelo DevGo por profissionais de DevRel para governar desenvolvedores, durante o estágio de retenção, no contexto de Ecossistema de Software Móvel (MSECO) é adequada, controlável, entendível e generalizável?”*, apresenta-se nas **Tabelas 36 e 37** a resposta para cada uma das questões de pesquisa que apoiaram a atingir o objetivo, descrevendo para cada métrica os resultados obtidos.

Para cada um dos critérios (adequação, controle, entendimento e generalidade) não houve nível de discordância a partir das respostas dos profissionais de DevRel. Os participantes, antes do uso do DevGo, descreveram suas estratégias para retenção de desenvolvedores em MSECO. Então, verificou-se como o DevGo poderia apoiar as estratégias já utilizadas por eles. No próximo passos, cada participante utilizou o DevGo, por meio de um instrumento, para selecionar os elementos tendo em vista estratégias de retenção.

Tabela 36. Resumo de resultados para cada critério.

Questão 1 – Adequação: Qual o nível de conformidade do DevGo para a governança de desenvolvedores em MSECO?	
Métrica	Resultado
(1) Percentual de concordância e discordância, em nível parcial ou total, para a adequação de áreas de foco, dos estágios, dos objetos de transferência de valor e das lições aprendidas.	- Área de Foco – Plataforma e Produtos: 75% (CT) e 25% (CP); - Área de Foco – DevRel (Evangelismo e Advocacia): 50% (CT) e 50% (CP); - Área de Foco – Fluxo de Avanço do Desenvolvedor: 25% (CT); 50% (CP) e 25% (N); - Área de Foco – Monitoramento: 100% (CT); - Estágios: 50% (CT), 25% (CP) e 25% (N); - Objetos de transferência de valor: 100% (CT); - Lições aprendidas: 50% (CT), 25% (CP) e 25% (N).
(2) Percentual de elementos do modelo e de lições aprendidas que têm sido abordados por profissionais de DevRel (lista consolidada a partir do conjunto inicial).	- Do total de 192 elementos selecionáveis: - Somente 7%(14) não foram selecionados; 178 (93%) foram selecionados, pelo menos, uma vez; 30 (16%) foram selecionados por todos participantes.
Questão 2 – Controle: O DevGo serve como guia para um profissional de DevRel monitorar e intervir durante a governança de desenvolvedores em MSECO	
Métrica	Resultado
(1) Percentual de concordância e discordância, em nível parcial ou total, em relação a percepção de melhoria de desempenho, produtividade e eficácia de profissionais de DevRel.	- Melhoria de desempenho: 100% (CT); - Melhoria de produtividade: 100% (CT); - Melhoria de eficácia: 100% (CT); - DevGo torna mais fácil o DevRel: 50% (CT), 25% (CP) e N (25%).
(2) Conjunto de benefícios e limitações do DevGo identificadas pelos profissionais de DevRel.	- Benefícios: (1) a estrutura do modelo; (2) a área de foco de monitoramento; (3) o auxílio a profissionais de DevRel no entendimento da estrutura de governança de desenvolvedores; - Limitações: (1) a inserção do DevGo em uma organização; (2) mensurar a efetividade dos facilitadores; (3) ausência de papeis na área de foco “DevRel (Evangelismo e Advocacia)”; (4) ausência de outro objetivo de retenção; (5) o nível de maturidade de profissionais de DevRel pra utilizar o DevGo
<i>Legenda: CT: Concorda Totalmente; CP: Concorda Parcialmente; N: Neutro.</i>	

Os elementos selecionados pelos participantes cobrem 93% daqueles disponíveis para seleção, considerando o estágio de retenção do DevGo. Os profissionais de DevRel destacaram como benefícios do modelo: sua estrutura que facilita o entendimento da

governança de desenvolvedores, dos objetos de transferência de valor, inclusive, para comunicar a importância da área para a organização e a área de foco de monitoramento.

Tabela 37. Resumo de resultados para cada critério (cont.).

Questão 3 – Entendimento: O profissional de DevRel consegue utilizar o DevGo para lidar com a governança de desenvolvedores em MSECO?	
Métrica	Resultado
Percentual de concordância e discordância, em nível parcial ou total, para a aplicabilidade, utilidade, clareza, compreensão do DevGo, inclusive, a relação com a realidade vivida por profissionais.	• Aplicabilidade: 75% (CT) e 25% (CP); • Utilidade: 100% (CT); • Clareza e Compreensão: 25% (CT) e 75% (CP); • Relação com a realidade vivida: 75% (CT) e 25% (CP).
Questão 4 – Generalidade: O DevGo serve como guia para um grupo de profissionais de DevRel e outros ECOS sem perder sua relevância?	
Métrica	Resultado
Percentual de concordância e discordância, em nível parcial ou total, em relação a relevância do modelo para: (1) outros ECOS, (2) outros profissionais de DevRel e (3) para empresas de pequeno, médio e grande porte.	• Relevância para outros ECOS: 50% (CT), 25% (CP) e N (25%); • Relevância para outros profissionais: 75% (CT) e 25% (CP); • Relevância para empresas de pequeno porte: 25% (CT), 50% (CP) e N (25%) • Relevância para empresas de médio porte: 25% (CT) e 75% (CP); • Relevância para empresas de grande porte: 50% (CT) e 50% (CP).
<i>Legenda: CT: Concorda Totalmente; CP: Concorda Parcialmente; N: Neutro.</i>	

Entretanto, como limitações, os participantes indicaram que o DevGo poderia indicar papéis dentro da área de foco de DevRel. Outro aspecto de preocupação é relacionado a como comunicar o DevGo dentro da organização, incluindo a efetividade dos facilitadores. O nível de detalhamento do DevGo, a medida que auxilia no entendimento da estrutura e fornece a visão geral, pode ser um ponto de dificuldade dependendo da maturidade do profissional de DevRel que o utilizará.

CAPÍTULO 7 – CONCLUSÕES E PERSPECTIVAS FUTURAS

Neste capítulo são apresentadas as considerações finais sobre a contribuição desta tese. Como perspectivas futuras, além de estudos que podem ser realizados para refinamento do modelo DevGo, são tratadas as áreas de investigação que podem ser exploradas pela comunidade científica.

7.1 Conclusões

A carência por modelos, o apoio para a definição e seleção de estratégias sustentáveis, a necessidade de visão realista da relação cooperativa entre a organização central e a massa crítica de desenvolvedores externos direcionou à investigação de um modelo. Logo, o objetivo desta tese foi estabelecer um modelo envolvendo elementos estruturais e um conjunto de lições aprendidas para a governança de desenvolvedores em MSECO que seja adequado, entendível, controlável e generalizável para apoiar estratégias de profissionais de DevRel.

Um primeiro passo envolveu a identificação do corpo de conhecimento, por meio de um mapeamento sistemático, sobre governança de desenvolvedores em ECOS. Os estudos relacionados aos ECOS do tipo proprietário, *open-source* e híbrido (onde se encaixa o MSECO) foram analisados. Não havia uma definição de governança de desenvolvedores em ECOS. Então, além da definição para cada tipo de ECOS, construiu-se a seguinte definição para governança de desenvolvedores, que foi avaliada por profissionais de DevRel:

“A governança de desenvolvedores consiste em instrumentos para manter relacionamentos sinérgicos entre desenvolvedores internos e externos, usuários do ECOS e organização central determinados a garantir o bem-estar financeiro, social e técnico para os desenvolvedores. Ela mantém a vitalidade do ecossistema e reduz custos e riscos.”

Neste sentido, esta tese definiu, refinou e avaliou um modelo para governança de desenvolvedores externos em MSECO, a partir da perspectiva de profissionais de DevRel. Um total de 68 profissionais da área de DevRel foi envolvido nos estudos. Há representantes de vários países, empresas de pequeno, médio e grande porte, além de, profissionais com diferentes papéis de DevRel. O modelo é chamado DevGo (do inglês, **Developer Governance**) e é composto por:

- Quatro Áreas de Foco: (1) Plataforma e Produtos; (2) DevRel (Evangelismo e Advocacia); (3) Fluxo de Avanço do Desenvolvedor; e (4) Monitoramento;

- Quatro Fases dentro da área “Fluxo de Avanço do Desenvolvedor”: (1) Início; (2) Crescimento; e (3) Maturidade;
- Seis Estágios que estão relacionados a cada fase: (1) Sensibilização; (2) Entrada; (3) Ativação; (4) Retenção; (5) Reconhecimento; e (6) Referência;
- Vinte e três facilitadores distribuídos entre os estágios;
- Sessenta e duas lições aprendidas distribuídas entre os estágios;
- Sessenta e seis objetos de transferência de valor que podem ser consumidos ou providos pelas áreas de foco.

Como forma de analisar a expectativas do desenvolvedor, foram utilizados mecanismos de mineração de repositórios de software para analisar perguntas e respostas no Stack Overflow. Os aspectos analisados envolvem desde tópicos mais frequentes, passando por séries temporais, a emoções primárias. Isto ajudou a refinar a área de foco de monitoramento e na extração de um conjunto de seis insights para apoiar os profissionais de DevRel durante a governança de desenvolvedores em MSECO.

O DevGo foi avaliado em estudos de caso com profissionais de DevRel, tendo em vista os seguintes critérios: adequação, controle, entendimento e generalidade. Além disso, foram capturados os benefícios e limitações percebidos pelos profissionais durante o uso do DevGo. Os elementos selecionados do modelo DevGo, visando a retenção de desenvolvedores, foi outra métrica analisada. Não houve nível de discordância dos profissionais em nenhum dos critérios avaliados. Porém, foi indicada a necessidade de incluir na área de foco de DevRel (Evangelismo e Advocacia) os papéis de DevRel e, também, de considerar a maturidade do profissional de DevRel dentro da área de foco.

O modelo DevGo pode ser útil nos seguintes cenários:

1. **Não há programa de DevRel:** conhecimento das áreas essenciais para planejar e iniciar um programa de governança de desenvolvedores por meio de DevRel;
2. **Estratégias de DevRel em nível básico, documentado e definido):** identificação de onde a organização está e onde ela pode ir;
3. **Estratégias próprias de DevRel:** guia em decisões envolvendo riscos e tendências para manter a competitividade em relação a outras organizações;
4. **Monitoramento e controle:** uso mais focado na área de Monitoramento para formar uma base sólida de mecanismos de avaliação e ferramentas para MSECOS com base mais extensa de desenvolvedores;
5. **Introdução de estratégias inovadoras para melhor atender às metas da organização:** apoio às organizações na evolução das estratégias de governança do desenvolvedor, dimensionar esforços e formar equipes internas.

7.2 Limitações

As limitações desta pesquisa envolvem a amostra de participantes, o tipo de ECOS analisado e o repositório analisado para extração das expectativas dos desenvolvedores:

- Sessenta e oito profissionais de DevRel participaram nos estudos. O que vale destacar aqui é que em todos estudos com profissionais a amostra era diferente. Porém, isto não elimina o fato de que é importante o envolvimento de outros profissionais;
- O modelo DevGo não é um modelo de referência, ainda, pelo fato de sua definição, refinamento e avaliação considerarem MSECO. Logo, há a necessidade de evoluir o modelo considerando outros tipos de ECOS;
- Os insights para monitoramento foram propostos em cima da categoria de “Perguntas e Respostas” da área de monitoramento. As outras categorias poderiam ser estudadas para a identificação de outros *insights*;
- O DevGo precisa ainda de um guia para sua utilização, como foi indicado por alguns dos participantes.

7.3 Perspectivas Futuras

A partir deste ponto, percebe-se que isto aqui gera oportunidades de investigação, de uma forma geral, nas seguintes áreas:

1. *Maturidade de Governança de Desenvolvedores*: a medida que se tenha um estabelecimento do modelo DevGo será possível analisar e diagnosticar a maturidade de cada organização do ponto de vista de DevRel. Com isto, além de ser um meio de reconhecer o nível da organização, poderão ser indicadas ações para o amadurecimento da organização. Assim como em outros modelos de maturidade e/ou capacidade de processos (p.ex.: CMMI e MPS.Br) é necessário a definição de níveis, guias de avaliação e implementação, por exemplo;
2. *Processo de Governança de Desenvolvedores*: criar instâncias do DevGo a partir de processos de negócio e de software. Analisar que notações favorecem o entendimento das instâncias, em forma de processos, pelos profissionais de DevRel;
3. *Estudos específicos para cada área de foco do DevGo*: cada uma das áreas do DevGo pode ser expandida estruturalmente por meio de novos estudos. A área de DevRel (Evangelismo e Advocacia), por exemplo, como indicado em alguns dos estudos, precisa ser aprimorada em relação aos papéis existentes dentro de evangelismo e advocacia.

Especificamente, na área de foco de “Fluxo de Avanço do Desenvolvedor”, as oportunidades de investigação se expandem:

1. *Fases e Estágios*: um exemplo é que há estudos focados na entrada de desenvolvedores novatos em projetos open-source, em testes de apps, na ativação de desenvolvedores e em mecanismos de reputação;
2. *Facilitadores*: cada um dos facilitadores pode se tornar um objeto de pesquisa, existem pesquisas explorando hackathons, treinamentos e até o processo de certificação de apps em MSECO;
3. *Lições Aprendidas*: um conjunto de lições aprendidas, com relevância analisada, pode servir como insumo para empresas que desejam implantar a área de DevRel, assim como, o ajuste da área visando a competitividade em empresas que a possuam.

Uma outra área, com potencial de investigação, é a criação de valor e o retorno sobre o investimento (ROI, do inglês, *Return On Investment*), que tem como base os objetos de transferência de valor identificados no DevGo. Inclusive, esta é uma área que durante a análise das lições aprendidas no *Medium* se destaca como sendo a mais discutida em conferências de profissional de DevRel. É difícil demonstrar aos setores organizacionais o motivo de investir em DevRel.

O monitoramento de governança dos desenvolvedores, a partir de cada categoria de repositórios, é outra área que pode se expandir para pesquisa. Por exemplo, o estudo por meio de séries temporais da alteração de emoções dos desenvolvedores durante o envolvimento no ecossistema. A análise cruzada entre os repositórios para gerar *insights* mais precisos. Isto poderá gerar um painel de controle sobre a governança de desenvolvedores que ajuda na análise global e por subsidiárias das organizações.

Um outro aspecto interessante é formação de profissionais para atuar com DevRel ainda dentro dos cursos oferecidos em universidades. A análise se os currículos atuais favorecem a formação de novos profissionais. É evidente que o profissional de DevRel, precisa desenvolver *soft* e *hard skills* quem envolvam os aspectos da computação, engenharia de sistemas e de negócios.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALBERT, B. E., DOS SANTOS, R. P. E WERNER, C. M. (2013). Software ecosystems governance to enable IT architecture based on software asset management. *Proceedings of the IEEE International Conference on Digital Ecosystems and Technologies*, pp. 55-60.
- ALVES, C., OLIVEIRA, J. E JANSEN, S. (2017). Software Ecosystems Governance - A Systematic Literature Review and Research Agenda. *Proceedings of the 19th International Conference on Enterprise Information Systems*, pp. 215-226.
- AMIT, R. E ZOTT, C. (2001). Value creation in e-business. *Strategic management journal*, Vol. 22, pp. 493-520.
- ARANGANAYAGI, S. E THANGAVEL, K. (2007). Clustering categorical data using silhouette coefficient as a relocating measure. *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA)*, pp. 13-17.
- ASCATE, S., MENDES, I., OLIVEIRA, K., FONTÃO, A., VILLANES, I., E DIAS-NETO, A. (2017). Model-Based Testing for Mobile Applications: A Systematic Mapping Study. *Proceedings of the Ibero-American Conference on Software Engineering*, pp. 567-580.
- AXELSSON, J. E SKOGLUND, M. (2016). Quality assurance in software ecosystems: A systematic literature mapping and research agenda. *Journal of Systems and Software*, vol. 114, pp. 69-81.
- BAARS, A. E JANSEN, S. (2012). A Framework for Software Ecosystem Governance. *Proceedings of International conference of software business*, pp. 168-180.
- BAJAJ, K. E MESBAH, A. (2014). Mining Questions Asked by Web Developers. *Proceedings of the 11th Working Conference on Mining Software Repositories*, pp. 112-121.
- BARBOSA, O., DOS SANTOS, R., ALVES, C., WERNER, C. E JANSEN, S. (2013). A Systematic Mapping Study on Software Ecosystems. In: Jansen, S.; Brinkkemper, S.; Cusumano, M. *Software Ecosystems. Analyzing and managing business networks in the software industry*. Edward Elgar Publishing Limited. pp. 59-78.
- BARUA, A., THOMAS, S. W., HASSAN, A. E. (2014). What are developers talking about? An analysis of topics and trends in Stack Overflow. *Empirical Software Engineering*, v. 19(3), pp. 619-654.
- BASILI, V. (1994). GQM approach has evolved to include models. *IEEE Software*, v. 11, n. 1.
- BAYONA, S., BUSTAMANTE, J. E SABOYA, N. (2018). PMBOK as a Reference Model for Academic Research Management. *Proceedings of the World Conference on Information Systems and Technologies*, pp. 863-876.
- BEGON, M., TOWNSEND, C. R. E HARPER, J. L. (2007). *Ecologia - De Indivíduos a Ecossistemas*. 4. ed. Porto Alegre: Artmed.
- BETTENBURG, N., HASSAN, A. E., ADAMS, B. E GERMAN, D. M. (2015). Management of community contributions - A case study on the Android and Linux software ecosystems. *Empirical Software Engineering*, pp. 252-289.
- BOSCH, J. (2009). From Software Product Lines to Software Ecosystems. *Proceedings of the 13th International Software Product Line Conference*, pp. 111-119.
- BOWMAN, C. E AMBROSINI, V. (2000). Value creation versus value capture: towards a coherent definition of value in strategy. *British journal of management*, 11(1), pp.1-15.
- CASALNUOVO, C., VASILESCU, B., DEVANBU, P. E FILKOV, V. (2015). Developer Onboarding in GitHub: The Role of Prior Social Links and Language Experience.

Proceedings of the 10th Joint Meeting on Foundations of Software Engineering, pp. 817-828.

- CHESBROUGH, H. (2003). Open Innovation The New Imperative for Creating and Profiting from Technology Xerox PARC The Achievements and Limits of Closed Innovation. *Harvard Business School Press*, pp. 1–10.
- CHRISTENSEN, H. B., HANSEN, K. M., KYNG, M. E MANIKAS, K. (2014). Analysis and design of software ecosystem architectures - Towards the 4S telemedicine ecosystem. *Information and Software Technology*, v. 56, n. 11, pp. 1476–1492.
- CONSTANTINOU, E. E MENS, T. (2017). Socio-technical evolution of the Ruby ecosystem in GitHub. *Proceedings of the 24th IEEE International Conference on Software Analysis, Evolution, and Reengineering*, pp. 34–44.
- COSENTINO, V., IZQUIERDO, J.L.C. E CABOT, J., (2017). A systematic mapping study of software development with GitHub. *IEEE Access*, vol. 5, pp.7173-7192.
- CRUZES, D. S. E DYBA, T. (2011). Recommended Steps for Thematic Synthesis in Software Engineering. *Proceedings of the International Symposium on Empirical Software Engineering and Measurement*, pp. 275-284.
- DHUNGANA, D., GROHER, I., SCHLUDERMANN, E. E BIFFL, S. (2010). Software Ecosystems vs . Natural Ecosystems: Learning from the Ingenious Mind of Nature. *Proceedings of the 4th European Conference on Software Architecture Companion Volume*, pp. 96–102.
- DIAS NETO, A. C. E TRAVASSOS, G. H. (2008). Surveying model based testing approaches characterization attributes. *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, pp. 324-326.
- DROST, E. (2011). Validity and Reliability in Social Science Research. *Education Research and Perspectives*, v. 38, n. 1, pp. 105–123.
- DUBINSKY, Y. E KRUCHTEN, P. (2009). Software development governance (SDG): report on 2nd workshop. *ACM SIGSOFT Software Engineering Notes*, v. 34, n. 5, pp. 46–47.
- ECKHARDT, E., KAATS, E., JANSEN, S. E ALVES, C. (2014). The merits of a meritocracy in open source software ecosystems. *Proceedings of the European Conference on Software Architecture*, pp. 7.
- EKLUND, U. E BOSCH, J. (2014). Architecture for embedded open software ecosystems. *Journal of Systems and Software*, v. 92, pp. 128–142.
- EKWOGÉ, O. M., FONTÃO, A. E DIAS-NETO, A. C. (2017). Tester Experience: Concept, Issues and Definition. *Proceedings of the International Computer Software and Applications Conference*, pp. 208-213.
- FAGERHOLM, F. E MÜNCH, J. (2012). Developer experience: Concept and definition. In *Proceedings of the International Conference on Software and System Process*, pp. 73-77.
- FARIAS, M. A. DE F., NOVAIS, R., COLAÇO, M.,CARVALHO, L.P., MENDONÇA, M. E SPÍNOLA, R.O. (2016). A Systematic Mapping Study on Mining Software Repositories. *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pp. 1472-1479.
- FOERDERER, J., KUDE, T., SCHUETZ, S. W. E HEINZL, A. (2018). Knowledge boundaries in enterprise software platform development: Antecedents and consequences for platform governance. *Information Systems Journal*, pp. 1–26.
- FONTÃO, A. E DIAS-NETO, A. C. (2016). GoDev-DX : Governança de Desenvolvedores em Ecossistemas de Software Móvel a partir da Experiência do Desenvolvedor. *Anais da*

Escola Latino-Americana de Engenharia de Software.

- FONTÃO, A., DIAS-NETO, A. E SANTOS, R. (2017a). Towards a Guideline-Based Approach to Govern Developers in Mobile Software Ecosystems. *Proceedings of the International Conference on Software Reuse*, pp. 208-213.
- FONTÃO, A., EKWOGE, O. M., DIAS-NETO, A. C. E SANTOS, R. (2017b). Facing up the primary emotions in Mobile Software Ecosystems from Developer Experience. *Proceedings of the 2nd Workshop on Social, Human, and Economic Aspects of Software*, pp. 5-11.
- FONTÃO, A., LIMA, F., ÁBIA, B., DOS SANTOS, R. P. E DIAS-NETO, A. C. (2017c). Hearing the Voice of Developers in Mobile Software Ecosystems. In *Proceedings of the 31st Brazilian Symposium on Software Engineering*, pp. 4-13.
- FONTÃO, A., DIAS-NETO, A. E VIANA, D. (2017d). Investigating Factors That Influence Developers' Experience in Mobile Software Ecosystems. In *Proceedings of the 11th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems*, pp. 55-58.
- FONTÃO, A., BONIFÁCIO, B. E DIAS-NETO, A. C. (2018a). Mobile Application Development Training in Mobile Software Ecosystem: Investigating the Developer eXperience. *Proceedings of the 17th Brazilian Symposium on Software Quality*, pp. 160-169
- FONTÃO, A., ESTÁCIO, B., FERNANDES, J., PEREIRA, R. E DIAS-NETO, A. C. (2018b). Which factors affect the evangelist's support during training sessions in Mobile Software Ecosystems? *Proceedings of the 12th European Conference on Software Architecture*, pp. 22-28.
- FONTÃO, A., DOS SANTOS, R. P. E DIAS-NETO, A. C. (2018c). Exploiting Repositories in Mobile Software Ecosystems from a Governance Perspective. *Information Systems Frontiers*, vol. 21(1), pp. 143–161.
- FONTÃO, A., ÁBIA, B., WIESE, I., ESTÁCIO, B., QUINTA, M., DOS SANTOS, R. P., & DIAS-NETO, A. C. (2018d). Supporting governance of mobile application developers from mining and analyzing technical questions in stack overflow. *Journal of Software Engineering Research and Development*, v. 6, n. 1, pp. 1–34.
- FONTÃO, A., WIESE, I., SANTOS, R. P. e DIAS-NETO, A.C . Governança de Desenvolvedores em Ecossistemas de Software. (2018e). In: R. P. Santos; A. L. Martinotto; S. Ávila e Silva.. (Org.). Tópicos em Sistemas de Informação - Minicursos do XIV Simpósio Brasileiro de Sistemas de Informação. 14ed.Porto Alegre: SBC, 2018, v. 14, p. 76-104.
- FONTÃO, A., SANTOS, R. P. e Dias-Neto, Arilo . (2018f). Towards a Guideline-Based Approach to Govern Developers in Mobile Software Ecosystems. In: Goetz Botterweck and Claudia Werner. (Org.). Mastering Scale and Complexity in Software Reuse. 1ed.: Springer International Publish, 2017, v. 1, p. 208-2015.
- FONTÃO, A. L., DOS SANTOS, R. P., FILHO, J. F. E DIAS-NETO, A. C. (2016). MSECODEV: Application development process in mobile software ecosystems. *Proceedings of the International Conference on Software Engineering and Knowledge Engineering*, pp. 317-322.
- FONTÃO, A., SANTOS, R. P. E DIAS-NETO, A. C. (2015). Mobile Software Ecosystem (MSECO): A Systematic Mapping Study. *Proceedings of the 39th Annual International Computers, Software & Applications Conference*.
- FRANCO-BEDOYA, O., AMELLER, D., COSTAL, D. E FRANCH, X. (2017). Open source software ecosystems: A Systematic mapping. *Information and Software Technology*, v. 91, pp. 160–185.
- GAROUSI, V., FELDERER, M. E MÄNTYLÄ, M.V., (2016). The need for multivocal literature

reviews in software engineering: complementing systematic literature reviews with grey literature. *Proceedings of the 20th international conference on evaluation and assessment in software engineering*, pp. 26.

- HAMBURG, M. (1980). Basic statistics: A modern approach. *Journal of the Royal Statistical Society*, v. 143, n. 1.
- HYRYNSALMI, S., SEPPÄNEN, M. E SUOMINEN, A. (2014). Sources of value in application ecosystems. *Journal of Systems and Software*, vol. 96, pp.61-72.
- HUBER, T.L., KUDE, T. E DIBBERN, J. (2017). Governance practices in platform ecosystems: Navigating tensions between cocreated value and governance costs. *Information Systems Research*, vol. 28(3), pp.563-584.
- JANSEN, S., CUSUMANO, M.A. AND BRINKKEMPER, S. (2013). Software ecosystems: analyzing and managing business networks in the software industry. Edward Elgar Publishing.
- 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 Systems and Software*, v. 85(7), pp. 1495–1510.
- JANSEN, S. E CUSUMANO, M. A. (2012). Defining software ecosystems: A survey of software platforms and business network governance. *Proceedings of the International Workshop on Software Ecosystems*, pp. 13.
- JENSEN, C. E SCACCHI, W. (2010). Governance in open source software development projects: A comparative multi-level analysis. *Open Source Software: New Horizons*, pp. 1–15.
- JERGENSEN, C., SARMA, A. E WAGSTROM, P. (2011). The Onion Patch: Migration in Open Source Ecosystems. *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering*, pp. 70-80.
- KE, W. E ZHANG, P. (2008). Motivations for participating in open source software communities: Roles of psychological needs and altruism. *Proceedings of the Pacific Asia Conference on Information Systems*, pp.76.
- KILAMO, T., HAMMOUDA, I., MIKKONEN, T. E AALTONEN, T. (2012). From proprietary to open source - Growing an open source ecosystem. *Journal of Systems and Software*, v. 85(7), pp. 1467–1478.
- KITCHENHAM, B. E CHARTERS, S. (2007). Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3. *Keele University EBSE Technical Report*.
- KRESTEL, R., FANKHAUSER, P. E NEJDL, W. (2009). Latent dirichlet allocation for tag recommendation. *Proceedings of the third ACM conference on Recommender systems*, pp. 61-68.
- KUDE, T. (2018). Successfully governing software ecosystems: Competence profiles of partnership managers. *IEEE Software*, pp. 1.
- LARSON, R.R. (2010). Information retrieval systems. *Encyclopedia of Library and Information Sciences*, 3rd Edition, CRC Press, New York, IV, pp.2553-2563.
- LIN, B. E SEREBRENIK, A. (2016). Recognizing Gender of Stack Overflow Users. *Proceedings of the 13th International Conference on Mining Software Repositories*, pp. 425-429.
- LINÅKER, J., MUNIR, H., RUNESON, P., REGNELL, B. E SCHREWELIUS, C. (2015). A survey on the perception of innovation in a large product-focused software organization. *Proceedings of the International Conference of Software Business*, pp. 66-80.
- LOPER, E. E BIRD, S. (2002). NLTK: The Natural Language Toolkit.

- MÄENPÄÄ, H., MUNZERO, M., FAGERHOLM, F. E MIKKONEN, T. (2017). The many hats and the broken binoculars. In *Proceedings of the 13th International Symposium on Open Collaboration*, pp. 18.
- MANIKAS, K. (2016). Revisiting software ecosystems Research: A longitudinal literature study. *Journal of Systems and Software*, v. 117, pp. 84–103.
- MANIKAS, K. E HANSEN, K. M. (2013a). Reviewing the Health of Software Ecosystems – A Conceptual Framework Proposal. *Proceedings of the 5th International Workshop on Software Ecosystems*, pp. 33-44.
- MANIKAS, K. E HANSEN, K. M. (2013b). Software ecosystems-A systematic literature review. *Journal of Systems and Software*, v. 86(5), pp. 1294–1306.
- MENDES, E., RODRIGUEZ, P., FREITAS, V., BAKER, S. E ATOUI, M. A. (2018). Towards improving decision making and estimating the value of decisions in value-based software engineering: the VALUE framework. *Software Quality Journal*, v. 26, n. 2, pp. 607–656.
- MELDRUM, S., LICORISH, S.A. E SAVARIMUTHU, B.T.R. (2017). Crowdsourced Knowledge on Stack Overflow: A Systematic Mapping Study. *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, pp. 180-185.
- MOLLÉRI, J.S., PETERSEN, K. E MENDES, E. (2016). Survey guidelines in software engineering: An annotated review. *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* , pp. 58.
- MORISIO, M., EZRAN, M. E TULLY, C. (2002). Success and failure factors in software reuse. *IEEE Transactions on Software Engineering*, v. 28(4), pp. 340–357.
- O'MAHONY, S. E FERRARO, F., 2007. The emergence of governance in an open source community. *Academy of Management Journal*, v. 50(5), pp.1079-1106.
- PETERSEN, K., VAKKALANKA, S. E KUZNIARZ, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, v. 64, pp. 1–18.
- QIU, Y., GOPAL, A. E HANN, I. (2011). Synthesizing Professional and Market Logics: a Study of Independent iOS App Entrepreneurs. *Academy of Management*, pp. 12803.
- RASTOGI, A. E NAGAPPAN, N. (2016). Forking and the Sustainability of the Developer Community Participation - An Empirical Investigation on Outcomes and Reasons. *Proceedings of the International Conference on Software Analysis, Evolution, and Reengineering*, pp. 102–111.
- ROBERTS, J.A., HANN, I.H. E SLAUGHTER, S.A. (2006). Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the Apache projects. *Management science*, v.52(7), pp. 984-999.
- ROSEN, C. E SHIHAB, E. (2016). What are mobile developers asking about? A large scale study using stack overflow. *Empirical Software Engineering*, v. 21(3), pp. 1192–1223.
- SADI, M. H., DAI, J. E YU, E. (2015). Designing Software Ecosystems: How to Develop Sustainable Collaborations? Scenarios from Apple iOS and Google Android. *Proceedings of the Advanced Information Systems Engineering Workshops*, pp. 161–173.
- SANCHEZ, O. P., TERLIZZI, M. A. E DE MORAES, H. R. DE O. C. (2017). Cost and time project management success factors for information systems development projects. *International Journal of Project Management*, v. 35(8), pp. 1608–1626.
- SANTOS, R.P. E WERNER, C.M.L., (2012). Treating social dimension in software ecosystems through reuseecos approach. *Proceedings of the 6th IEEE International*

- Conference on Digital Ecosystems and Technologies (DEST) (pp. 1-6).
- SONG, J., BAKER, J., WANG, Y., CHOI, H. Y. AND BHATTACHERJEE, A. (2018). Platform adoption by mobile application developers: A multimethodological approach. *Decision Support Systems*, v. 107, pp. 26–39.
- STEGLICH, C., MARCZAK, S., GUERRA, L.P. E MOSMANN, L.H. (2019). Revisiting the Mobile Software Ecosystem Literature. *Proceedings of the 13th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems*, to appear.
- STEINMACHER, I., CONTE, T. E GEROSA, M. (2015). Social barriers faced by newcomers placing their first contribution in open source software projects. *Proceedings of the 18th ACM conference on Computer supported cooperative work & social computing*, pp. 1379-1392.
- STRAUSS, A. E CORBIN, J. (1998). Basics of qualitative research techniques. Thousand Oaks, CA: Sage publications.
- TIWANA, A., KONSYNSKI, B. E BUSH, A. A. (2010). Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics. *Information Systems Research*, v. 21(4), pp. 675–687.
- VORRABER, W., MULLER, M., VOESSNER, S. E SLANY, W. (2018). Analyzing and Managing Complex Software Ecosystems - A Framework for Creating a Common Understanding and Aligning Shared Goals for Developers and Business Managers, Applied to a Free Open Source Software Project. *IEEE Software*
- WAREHAM, J., FOX, P. B. E CANO GINER, J. L. (2014). Technology Ecosystem Governance. *Organization Science*, v. 25(4), pp. 1195–1215.
- WOHLIN, C., RUNESON, P., DA MOTA SILVEIRA NETO, P. A. (2013). On the reliability of mapping studies in software engineering. *Journal of Systems and Software*, v. 86(10), pp. 2594–2610.
- YIN, R.K. (2009). How to do better case studies. *The SAGE handbook of applied social research methods*, 2, pp.254-282.

APÊNDICE A – Artigos Selecionados no Mapeamento Sistemático

- S1. Nagy, D., Yassin, A.M. and Bhattacharjee, A., 2010. Organizational adoption of open source software: barriers and remedies. *Communications of the ACM*, 53(3), pp.148-151.
- S2. Tekinerdogan, B. and Scholten, H., 2015. Software Ecosystems for the Life Sciences Application Domains. *ACM SIGSOFT Software Engineering Notes*, 40(6), pp.1-6.
- S3. Brown, A.W. and Booch, G., 2002, April. Reusing open-source software and practices: The impact of open-source on commercial vendors. In *International Conference on Software Reuse* (pp. 123-136). Springer, Berlin, Heidelberg.
- S4. Bitzer, J. and Srhroder, P.J., 2006. Linux vs. Windows: A Comparison of Application and Platform Innovation Incentives for Open Source and Proprietary Software Platforms “published in Elsevier BV.
- S5. Jansen, S. and Brinkkemper, S., 2006, May. Evaluating the release, delivery, and deployment processes of eight large product software vendors applying the customer configuration update model. In *Proceedings of the 2006 international workshop on Workshop on interdisciplinary software engineering research* (pp. 65-68). ACM.
- S6. Scacchi, W., 2007. Free/open source software development: Recent research results and methods. *Advances in Computers*, 69, pp.243-295.
- S7. Cavusoglu, H. and Raghunathan, S., 2007. Efficiency of vulnerability disclosure mechanisms to disseminate vulnerability knowledge. *IEEE Transactions on Software Engineering*, 33(3).
- S8. Jansen, S., Finkelstein, A. and Brinkkemper, S., 2007. Analyzing the Business of Software: A Modelling Technique for Software Supply Networks. In *CAiSE Forum*.
- S9. Jensen, C. and Scacchi, W., 2007. Guiding the discovery of open source software processes with a reference model. *Open Source Development, Adoption and Innovation*, pp.265-270.
- S10. Jansen, S., Brinkkemper, S., Hunink, I. and Demir, C., 2008. Pragmatic and opportunistic reuse in innovative start-up companies. *IEEE software*, 25(6).
- S11. Van Der Schuur, H., Jansen, S. and Brinkkemper, S., 2008, September. Becoming responsive to service usage and performance changes by applying service feedback metrics to software maintenance. In *Proceedings of the 23rd IEEE/ACM International Conference on Automated Software Engineering* (pp. III-53). IEEE Press.
- S12. Fricker, S., 2009. Specification and Analysis of Requirements Negotiation Strategy in Software Ecosystems. In *IWSECO@ ICSR*.
- S13. Boucharas, V., Jansen, S. and Brinkkemper, S., 2009, August. Formalizing software ecosystem modeling. In *Proceedings of the 1st international workshop on Open component ecosystems* (pp. 41-50). ACM.
- S14. Faraz, S.M., Behlim, S.I., Khan, S.M. and Sattar, S.A., 2009, December. Interactive training framework a new approach to eLearning. In *Proceedings of the 7th International Conference on Frontiers of Information Technology* (p. 79). ACM.
- S15. Kabbedijk, J., Brinkkemper, S., Jansen, S. and van der Veldt, B., 2009, August. Customer involvement in requirements management: lessons from mass market software development. In *Requirements Engineering Conference, 2009. RE'09. 17th IEEE International* (pp. 281-286). IEEE.
- S16. van den Berk, I., Jansen, S. and Luinenburg, L., 2010, August. Software ecosystems: a software ecosystem strategy assessment model. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume* (pp. 127-134). ACM.

- S17. Seichter, D., Dhungana, D., Pleuss, A. and Hauptmann, B., 2010, August. Knowledge management in software ecosystems: software artefacts as first-class citizens. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume* (pp. 119-126). ACM.
- S18. van der Schuur, H., Jansen, S. and Brinkkemper, S., 2010, September. A reference framework for utilization of software operation knowledge. In *Software Engineering and Advanced Applications (SEAA), 2010 36th EUROMICRO Conference on* (pp. 245-254). IEEE.
- S19. Grottke, M., Karg, L.M. and Beckhaus, A., 2010, July. Team Factors and Failure Processing Efficiency: An Exploratory Study of Closed and Open Source Software Development. In *Computer Software and Applications Conference (COMPSAC), 2010 IEEE 34th Annual* (pp. 188-197). IEEE.
- S20. Fricker, S., 2010. Requirements value chains: Stakeholder management and requirements engineering in software ecosystems. *Requirements Engineering: Foundation for Software Quality*, pp.60-66.
- S21. Jensen, C. and Scacchi, W., 2010. Governance in open source software development projects: A comparative multi-level analysis. *Open Source Software: New Horizons*, pp.130-142.
- S22. van Ingen, K., van Ommen, J. and Jansen, S., 2011, November. Improving activity in communities of practice through software release management. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems* (pp. 94-98). ACM.
- S23. van Angeren, J., Blijleven, V. and Jansen, S., 2011, November. Relationship intimacy in software ecosystems: a survey of the dutch software industry. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems* (pp. 68-75). ACM.
- S24. Idu, A., van de Zande, T. and Jansen, S., 2011, November. Multi-homing in the apple ecosystem: why and how developers target multiple apple app stores. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems* (pp. 122-128). ACM.
- S25. Draxler, S., Jung, A., Boden, A. and Stevens, G., 2011, May. Workplace warriors: identifying team practices of appropriation in software ecosystems. In *Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering* (pp. 57-60). ACM.
- S26. van der Schuur, H., Jansen, S. and Brinkkemper, S., 2011, November. The power of propagation: on the role of software operation knowledge within software ecosystems. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems* (pp. 76-84). ACM.
- S27. Ververs, E., van Bommel, R. and Jansen, S., 2011, November. Influences on developer participation in the debian software ecosystem. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems* (pp. 89-93). ACM.
- S28. Jergensen, C., Sarma, A. and Wagstrom, P., 2011, September. 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.
- S29. Mizushima, K. and Ikawa, Y., 2011, July. A structure of co-creation in an open source software ecosystem: A case study of the eclipse community. In *Technology Management in the Energy Smart World (PICMET), 2011 Proceedings of PICMET'11:* (pp. 1-8). IEEE.
- S30. Racz, N., Weippl, E. and Seufert, A., 2011, January. Governance, risk & compliance (GRC) software-an exploratory study of software vendor and market research perspectives. In *System Sciences (HICSS), 2011 44th Hawaii International Conference on* (pp. 1-10). IEEE.
- S31. Scholten, U., Fischer, R. and Zirpins, C., 2012, April. The dynamic network notation: harnessing network effects in paas-ecosystems. In *Proceedings of the Fourth Annual Workshop on Simplifying Complex Networks for Practitioners* (pp. 25-30). ACM.

- S32. Nguyen, T.T., Nguyen, T.N., Duesterwald, E., Klinger, T. and Santhanam, P., 2012, June. Inferring developer expertise through defect analysis. In Software Engineering (ICSE), 2012 34th International Conference on (pp. 1297-1300). IEEE.
- S33. Kilamo, T., Hammouda, I., Mikkonen, T. and Aaltonen, T., 2012. From proprietary to open source—Growing an open source ecosystem. *Journal of Systems and Software*, 85(7), pp.1467-1478.
- S34. Krogstie, J., 2012. Modeling of digital ecosystems: Challenges and opportunities. *Collaborative Networks in the Internet of Services*, pp.137-145.
- S35. Qiu, Y., Hann, I.H. and Gopal, A., 2014, January. From Invisible Hand to Visible Hand: Platform Governance and Institutional Logics of Independent Mac Application Developers, 2001-2012. In *Academy of Management Proceedings* (Vol. 2014, No. 1, p. 14385). Academy of Management.
- S36. Bettenburg, N., Hassan, A.E., Adams, B. and German, D.M., 2015. Management of community contributions. *Empirical Software Engineering*, 20(1), pp.252-289.
- S37. Handoyo, E., Jansen, S. and Brinkkemper, S., 2013, October. Software ecosystem modeling: the value chains. In *Proceedings of the Fifth International Conference on Management of Emergent Digital Ecosystems* (pp. 17-24). ACM.
- S38. Taylor, R.N., 2013, August. The role of architectural styles in successful software ecosystems. In *Proceedings of the 17th International Software Product Line Conference* (pp. 2-4). ACM.
- S39. Musil, J., Musil, A. and Biffl, S., 2013, August. 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.
- S40. Schultis, K.B., Elsner, C. and Lohmann, D., 2013, May. Moving towards industrial software ecosystems: Are our software architectures fit for the future?. In *Product Line Approaches in Software Engineering (PLEASE), 2013 4th International Workshop on* (pp. 9-12). IEEE.
- S41. Hoving, R., Slot, G. and Jansen, S., 2013, July. Python: Characteristics identification of a free open source software ecosystem. In *Digital Ecosystems and Technologies (DEST), 2013 7th IEEE International Conference on* (pp. 13-18). IEEE.
- S42. Cardoso Jr, J.L., Barbin, S.E. and Andres, F., 2013, October. The public software ecosystem: exploratory survey. In *Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems* (pp. 289-296). ACM.
- S43. Exploring the relationship between partnership model participation and interfirm network structure: An analysis of the Office365 ecosystem
- S44. van Angeren, J., Jansen, S. and Brinkkemper, S., 2014, June. Exploring the relationship between partnership model participation and interfirm network structure: An analysis of the office365 ecosystem. In *International Conference of Software Business* (pp. 1-15). Springer, Cham.
- S45. Eckhardt, E., Kaats, E., Jansen, S. and Alves, C., 2014, August. The merits of a meritocracy in open source software ecosystems. In *Proceedings of the 2014 European Conference on Software Architecture Workshops* (p. 7). ACM.
- S46. Santos, R.P.D., 2014, May. ReuseSEEM: an approach to support the definition, modeling, and analysis of software ecosystems. In *Companion Proceedings of the 36th International Conference on Software Engineering* (pp. 650-653). ACM.
- S47. Miranda, M., Ferreira, R., de Souza, C.R., Figueira Filho, F. and Singer, L., 2014, June. An exploratory study of the adoption of mobile development platforms by software engineers. In *Proceedings of the 1st International Conference on Mobile Software Engineering and Systems* (pp. 50-53). ACM.
- S48. Teixeira, J. and Lin, T., 2014, May. Collaboration in the open-source arena: the webkit case. In *Proceedings of the 52nd ACM conference on Computers and people research* (pp. 121-129). ACM.

- S49. Schultis, K.B., Elsner, C. and Lohmann, D., 2014, November. 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* (pp. 542-552). ACM.
- S50. Valença, G., Alves, C., Heimann, V., Jansen, S. and Brinkkemper, S., 2014, August. Competition and collaboration in requirements engineering: A case study of an emerging software ecosystem. In *Requirements Engineering Conference (RE), 2014 IEEE 22nd International* (pp. 384-393). IEEE.
- S51. Knauss, E., Damian, D., Knauss, A. and Borici, A., 2014, August. Openness and requirements: opportunities and tradeoffs in software ecosystems. In *Requirements Engineering Conference (RE), 2014 IEEE 22nd International* (pp. 213-222). IEEE.
- S52. Rickmann, T., Wenzel, S. and Fischbach, K., 2014. Software ecosystem orchestration: the perspective of complementors.
- S53. Che, M. and Perry, D.E., 2014, April. Architectural design decisions in open software development: A transition to software ecosystems. In *Software Engineering Conference (ASWEC), 2014 23rd Australian* (pp. 58-61). IEEE.
- S54. Beckhaus, A., Neumann, D. and Karg, L., 2014. The Emergence of Intra-Organizational Communities of Operations: Evidence from the Software Industry.
- S55. Sadi, M.H., Dai, J. and Yu, E., 2015, June. Designing Software Ecosystems: How to Develop Sustainable Collaborations?. In *International Conference on Advanced Information Systems Engineering* (pp. 161-173). Springer International Publishing.
- S56. Goldbach, T. and Benlian, A., 2015. Understanding Informal Control Modes on Software Platforms—The Mediating Role of Third-Party Developers' Intrinsic Motivation.
- S57. Manikas, K. and Kontogiorgos, D., 2015, September. Characterizing software activity: The influence of software to ecosystem health. In *Proceedings of the 2015 European Conference on Software Architecture Workshops* (p. 46). ACM.
- S58. Lima, T., dos Santos, R.P. and Werner, C., 2015, October. A survey on socio-technical resources for software ecosystems. In *Proceedings of the 7th International Conference on Management of computational and collective intelligence in Digital EcoSystems* (pp. 72-79). ACM.
- S59. Alami, D., Rodríguez, M. and Jansen, S., 2015, September. Relating health to platform success: Exploring three e-commerce ecosystems. In *Proceedings of the 2015 European Conference on Software Architecture Workshops* (p. 43). ACM.
- S60. Kilamo, T., Rahikkala, J. and Mikkonen, T., 2015, August. Spicing Up Open Source Development with a Touch of Crowdsourcing. In *Software Engineering and Advanced Applications (SEAA), 2015 41st Euromicro Conference on* (pp. 390-397). IEEE.
- S61. Mens, T. and Grosjean, P., 2015. The ecology of software ecosystems. *Computer*, 48(10), pp.85-87.
- S62. Gharehyazie, M., Posnett, D., Vasilescu, B. and Filkov, V., 2015. Developer initiation and social interactions in OSS: A case study of the Apache Software Foundation. *Empirical Software Engineering*, 20(5), pp.1318-1353.
- S63. de Souza, C.R., Figueira Filho, F., Miranda, M., Ferreira, R.P., Treude, C. and Singer, L., 2016, May. The social side of software platform ecosystems. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (pp. 3204-3214). ACM.
- S64. Hyrynsalmi, S., Suominen, A. and Mäntymäki, M., 2016. The influence of developer multi-homing on competition between software ecosystems. *Journal of Systems and Software*, 111, pp.119-127.
- S65. Haile, N. and Altmann, J., 2016. Structural analysis of value creation in software service platforms. *Electronic Markets*, 26(2), pp.129-142.

- S66. van Angeren, J., Alves, C. and Jansen, S., 2016. Can we ask you to collaborate? Analyzing app developer relationships in commercial platform ecosystems. *Journal of Systems and Software*, 113, pp.430-445.
- S67. Goldbach, T., Benlian, A. and Buxmann, P., 2018. Differential effects of formal and self-control in mobile platform ecosystems: Multi-method findings on third-party developers' continuance intentions and application quality. *Information & Management*, 55(3), pp.271-284.
- S68. Fontão, A., Ekwoje, O.M., Santos, R. and Dias-Neto, A.C., 2017, May. Facing up the primary emotions in Mobile Software Ecosystems from Developer Experience. In *Proceedings of the 2nd Workshop on Social, Human, and Economic Aspects of Software* (pp. 5-11).
- S69. Fontão, A., Lima, F., Ábia, B., dos Santos, R.P. and Dias-Neto, A.C., 2017, September. Hearing the Voice of Developers in Mobile Software Ecosystems. In *Proceedings of the 31st Brazilian Symposium on Software Engineering* (pp. 4-13).
- S70. Poo-Caamaño, G., Knauss, E., Singer, L. and German, D.M., 2017. Herding cats in a FOSS ecosystem: a tale of communication and coordination for release management. *Journal of Internet Services and Applications*, 8(1), p.12.
- S71. Bogart, C., Kästner, C., Herbsleb, J. and Thung, F., 2016, November. How to break an API: cost negotiation and community values in three software ecosystems. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering* (pp. 109-120).
- S72. Avila, A. and Terzidis, O., 2016. Management of Partner Ecosystems in the Enterprise Software Industry. In *IWSECO@ ICIS* (pp. 39-55).
- S73. 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*, 135, pp.17-36.
- S74. Hyrynsalmi, S., Suominen, A., Jansen, S. and Yrjökoski, K., 2016. Multi-homing in ecosystems and firm performance: Does it improve software companies' ROA?. In *IWSECO@ ICIS* (pp. 56-69).
- S75. Schall, D., 2016. Skill-based Team Formation in Software Ecosystems. In *Proceedings of DECOSYS 2016*.
- S76. Constantinou, E. and Mens, T., 2017, February. Socio-technical evolution of the Ruby ecosystem in GitHub. In *24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pp. 34-44.
- S77. Duc, A.N., Cruzes, D.S., Hanssen, G.K., Snarby, T. and Abrahamsson, P., 2017, June. Coopetition of software firms in Open source software ecosystems. In *International Conference of Software Business* (pp. 146-160). Springer, Cham.
- S78. Lima, T., Dos Santos, R., Oliveira, J., Werner, C., 2016. The importance of socio-technical resources for software ecosystems management. In *Journal of Innovation in Digital Ecosystems*, Volume 3, Issue 2, pp. 98-113.
- S79. Valença, G. and Alves, C., 2017. A theory of power in emerging software ecosystems formed by small-to-medium enterprises. *Journal of Systems and Software*, 134, pp.76-104.
- S80. Vorraber, W., Müller, M., Voessner, S. and Slany, W., 2018. Analyzing and Managing Complex Software Ecosystems-A Framework for Creating a Common Understanding and Aligning Shared Goals for Developers and Business Managers, Applied to a Free Open Source Software Project. *IEEE software*.
- S81. Than, E.P.P., Nolte, A., Filippova, A., Bird, C., Scallen, S. and Herbsleb, J., 2018. Designing Corporate Hackathons With a Purpose. *IEEE Software*.
- S82. da Silva Amorim, S., McGregor, J.D., de Almeida, E.S. and von Flach G Chavez, C., 2018, September. Educating to achieve healthy open source ecosystems. In *Proceedings of the 12th European Conference on Software Architecture: Companion Proceedings* (p. 28).
- S83. Alves, C., Valença, G. and Franch, X., 2018. Exercising Power in Software Ecosystems. *IEEE Software*.

- S84. Fontão, A., dos Santos, R.P. and Dias-Neto, A.C., 2018. Exploiting Repositories in Mobile Software Ecosystems from a Governance Perspective. *Information Systems Frontiers*, pp.1-19.
- S85. Steinmacher, I., Treude, C. and Gerosa, M., 2018. Let me in: Guidelines for the Successful Onboarding of Newcomers to Open Source Projects. *IEEE Software*.
- S86. Fontão, A., Bonifácio, B., Santos, R.P.D. and Dias-Neto, A.C., 2018, October. Mobile Application Development Training in Mobile Software Ecosystem: Investigating the Developer eXperience. In *Proceedings of the 17th Brazilian Symposium on Software Quality* (pp. 160-169).
- S87. Valença, G., Alves, C. and Jansen, S., 2018. Strategies for managing power relationships in software ecosystems. *Journal of Systems and Software*, 144, pp.478-500.
- S88. Kude, T., Huber, T. and Dibbern, J., 2018. Successfully Governing Software Ecosystems: Competence profiles of partnership managers. *IEEE Software*.
- S89. Marsan, J., Templier, M., Marois, P., Adams, B., Carillo, K. and Mopenza, G.L., 2018. Towards solving social and technical problems in open source software ecosystems: Using cause-and-effect analysis to disentangle the causes of complex problems. *IEEE Software*.
- S90. Farias, V., Wiese, I. and Santos, R., 2018. What Characterizes an Influencer in Software Ecosystems?. *IEEE Software*.
- S91. Fontão, A., Estácio, B., Fernandes, J., dos Santos, R.P. and Dias-Neto, A.C., 2018, September. 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).

APÊNDICE B – Pesquisa de Opinião sobre Relevância de Estratégias

What is the relevance of strategies to govern mobile application developers?

Hello, we are researchers from Brazil investigating about Mobile Software Ecosystems (e.g Apple, Google, and Microsoft). We aim to identify and analyze which strategies could be relevant in the support provided by evangelists, managers, and partner engineers during the governing of mobile application developers.

Mobile Application developers' governance is defined by a set of mechanisms to attract external developers to create products and new capabilities to react to fast changing markets and customers needs, thereby fostering the developer's innovation and expanding the demand by user of software products. The developer's governance affects business and economic features of the software industry and also the quality of the ecosystem platform.

This is an academic research with no commercial interests. We will openly publish the results so everyone can benefit from them, making anonymous everything before doing so. Once the data is collected and analyzed, your email will be removed from the data and it will not be used at any time during the analysis or when we present the results. If at some point during this survey you want to leave, you're free to do so without any negative consequences.

Thanks a lot!

Arilo Claudio Dias Neto - Federal University of Amazonas (UFAM) - arilo@icomp.ufam.edu.br

Awdren Fontão - Federal University of Amazonas (UFAM) - awdren@icomp.ufam.edu.br

Bernardo Estácio - Pontifical Catholic University of Rio Grande (PUCRS) -

bernardo.estacio@acad.pucrs.br

Rodrigo Pereira dos Santos - Federal University of the State of Rio de Janeiro (UNIRIO) -

rps@uniriotec.br

Your profile

Platforms on which you have supported/managed developers: *

- ☐ Android
- ☐ iOS
- ☐ Windows
- ☐ Other: _____

How long (in months or years) have you been supporting/managing developers for that mobile platform? *

Your answer _____

In which country do you live? *

Your answer _____

Which of the following best describes your primary role in supporting/managing mobile application developers? *

- ☐ Developer Evangelist, Developer Advocate, Partner Engineer
- ☐ Manager
- ☐ Community Leader
- ☐ Researcher
- ☐ Other: _____

Strategies to govern developers

Please, select a number within 1-5 indicating the relevance of a strategy in governing mobile application developers, where: 0 - No relevance, 1 - Low Relevance, 3 - Medium Relevance and 5 - High Relevance

Analyze barriers for participation of developers and potential remedies to them *

0 1 2 3 4 5

☐ ☐ ☐ ☐ ☐ ☐

Meet the developers' respective needs to maintain absorptive capacity and avoid discouraging them *

0 1 2 3 4 5

☐ ☐ ☐ ☐ ☐ ☐

Identify the market needs and construct a well-defined development process *

0 1 2 3 4 5

☐ ☐ ☐ ☐ ☐ ☐

Create and maintain a reference model to coordinate developers *

0 1 2 3 4 5

☐ ☐ ☐ ☐ ☐ ☐

APÊNDICE C – Dados sobre Entrevistas

Tabela 38. Perfil dos participantes das entrevistas.

ID	Anos	País	Perfil
P1	8	Canadá	Ajuda as empresas a obter uma compreensão mais profunda das forças que moldam os mercados e tem como objetivo delinear as oportunidades.
P2	7	Brasil	Dirige Tecnologia e Pesquisa incentivando a criação de soluções e startups de tecnologia.
P3	7	México	Coordenar relações com os desenvolvedores de mais de 120 universidades na América Latina para treinar estudantes, empresários e comunidade local.
P4	6	Alemanha	Direciona e localiza os programas de desenvolvedores globais para a LATAM, refletindo as necessidades locais, garantindo a comunicação oportuna entre equipes globais e locais. Coordena iniciativas de marketing da comunidade para criar uma forte comunidade de desenvolvedores.
P5	5	EUA	Trabalha com as principais marcas e desenvolvedores para trazer aplicativos da Web ou móveis novos ou já existentes para as plataformas Microsoft.
P6	5	Canadá	Trabalha com equipes remotas de DevRel e indivíduos de diferentes nacionalidades e habilidades.
P7	6	Brasil	Coordena as comunidades de desenvolvedores para direcionar o uso de recursos técnicos e o desenvolvimento de contribuições para o ecossistema.
P8	6	Brasil	Atua no planejamento, na promoção e na execução do engajamento de parceiros por meio de suporte técnico, hackathons, palestras, treinamentos, palestras e coaching de projetos.
P9	9	EUA	Funciona em escala como a interface entre as ofertas de desenvolvedores da organização e os ecossistemas de desenvolvedores de terceiros. Encorajamos novos desenvolvedores a nossas ofertas, garantindo um DX avançado.
P10	13	EUA	Tem formação técnica para construir soluções e interagir com comunidades de desenvolvimento, consegue comunicar ideias e estratégias de forma eficaz.
P11	8	EUA	Responsável pela liderança e gestão geral da empresa, incluindo vendas, marketing, suporte e engenharia que dão suporte às atividades de DevRel.
P12	5	EUA	Responsável por estratégias que promovem a comunidade de desenvolvedores e ajudam a direcionar as ações em torno das relações com o desenvolvedor da empresa.
P13	3	Inglaterra	Trabalha para ajudar as empresas a criar equipes de relações com desenvolvedores.
P14	16	EUA	Com ampla experiência global em estratégia de marketing, execução e operações, trabalhando com diversas equipes para ajudar as empresas a desenvolver seus negócios através de atividades de construção e parcerias.
P15	7	EUA	Chefe de Marketing de Desenvolvedores e gerenciamento de equipes DevRel.

APÊNDICE D – Pesquisa de Opinião sobre Percepção de Valor

Hello, we are researchers investigating about Developer Relations (DevRel). We aim to identify how the value is perceived in developer relations activities.

This is an academic research with no commercial interests. We will openly publish the results so everyone can benefit from them, making anonymous everything before doing so. Once the data is collected and analyzed, your email will be removed from the data and it will not be used at any time during the analysis or when we present the results. If at some point during this survey you want to leave, you are free to do so without any negative consequences.

Thanks a lot!

Arilo Claudio Dias Neto - Federal University of Amazonas (UFAM) - arilo@icomp.ufam.edu.br
Awdren Fontão - Federal University of Amazonas (UFAM) - awdren@icomp.ufam.edu.br
Igor Wiese - Federal University of Technology - Paraná (UTFPR) - igor@utfpr.edu.br
Rodrigo Pereira dos Santos - Federal University of the State of Rio de Janeiro (UNIRIO) - rps@uniriotec.br

OK

* 1. Platforms on which you have supported developers:

- | | |
|--|--|
| ☐ Android | ☐ Microsoft Azure |
| ☐ iOS | ☐ Unity |
| ☐ Windows | ☐ Facebook |
| ☐ Symbian | ☐ Twitter |
| ☐ Amazon Web Services | |
| ☐ Other (please specify the platform) | |

* 2. How long (in months) have you been supporting developers for that platform?

* 3. How many trainings have you performed?

* 4. How many events (e.g., hackathons, developers conference, meetups) have you performed?

* 5. What is the total number of employees in your company?

- ☐ 0 - 99 employees
☐ 100 - 999 employees
☐ Over 1000 employees

* 6. What is your title at your company?

Value Perception - Repositories

To select the repositories in the question below consider the repositories with the following examples:

1. Questions & Answers (e.g. Stack Overflow);
2. Mailing lists/forums (e.g. Discourse, Android Developers, Windows Developers, Apple Developers);
3. Social Coding Environment (e.g., Github, Bitbucket, Code Plex);
4. Social News web site (e.g. Reddit, Hacker News);
5. Social Network (e.g. Facebook, Twitter);
6. Team Collaboration (e.g. Telegram, WhatsApp, Slack, Confluence);
7. App Store (e.g. Google Play, App Store, Windows Store);
8. Developer Dashboard (e.g. Android Developers, Apple Developers).

* 7. Check the motivations that lead you to contribute/use the repositories during your Developer Relations activities.

	Questions & Answers	Mailing Lists/forums	Social Coding Environment	Social News Web Site	Social Network	Team Collaboration	App Store	Developer Dashboard
Long-term strategy	☐	☐	☐	☐	☐	☐	☐	☐
Community Growth	☐	☐	☐	☐	☐	☐	☐	☐
Financial Value	☐	☐	☐	☐	☐	☐	☐	☐
ROI	☐	☐	☐	☐	☐	☐	☐	☐
Delivery Cost	☐	☐	☐	☐	☐	☐	☐	☐
Maintenance Cost	☐	☐	☐	☐	☐	☐	☐	☐
Operational Cost	☐	☐	☐	☐	☐	☐	☐	☐
Product's Development Effort	☐	☐	☐	☐	☐	☐	☐	☐
Time-to-market	☐	☐	☐	☐	☐	☐	☐	☐
Technical efficiency enabler	☐	☐	☐	☐	☐	☐	☐	☐
Product's competitiveness	☐	☐	☐	☐	☐	☐	☐	☐
Developer Satisfaction	☐	☐	☐	☐	☐	☐	☐	☐

APÊNDICE E – Estudo de Caso e Entrevistas

1. Preparação

- a. Enviar e-mail marcando a data e o horário do estudo com o participante;
- b. No dia do estudo, o primeiro instrumento a ser aberto está no link:
<https://forms.gle/FydnkGAtCkxeDQyE7>

2. Contextualização

- a. Apresentação da pesquisa, incluindo os pesquisadores;
- b. Leitura do Termo de Consentimento Livre e Esclarecido (TCLE). Caso o participante concorde com o TCLE, o pesquisador poderá seguir para a próxima parte do estudo.

3. Caracterização de Perfil

- a. O pesquisador informará que tudo será gravado mas dentro dos termos do TCLE;
- b. O pesquisador direciona o participante a responder o formulário de caracterização de perfil.

4. Entendimento da Estratégia

- a. Como primeira parte do estudo, o participante é convidado a descrever como planeja, executa e analisa atividades de DevRel visando a retenção de desenvolvedores.

5. Utilização do modelo DevGo

- a. O pesquisador pedirá ao participante que abra em uma outra janela o link que contém o DevGo;
- b. O pesquisador apresenta o DevGo somente a partir da parte estrutural;
- c. O pesquisador pede, para fins de verificação do entendimento da estrutura do DevGo, que o participante localize e informe as lições aprendidas (a numeração delas) dentro do DevGo;
- d. O pesquisador pergunta ao participante se há alguma dúvida em relação à estrutura do DevGo;
- e. O participante segue a leitura;
- f. O participante deverá ir registrando como utilizou cada parte do modelo visando a retenção;
- g. O pesquisador informará ao participante que este utilizará o instrumento com o DevGo visando a retenção de desenvolvedores.
- h. O participante precisará abrir o link disponível no instrumento. Este link conterá o documento para preenchimento dos elementos;

- i. Para isto, o pesquisador informará ao participante o caso que está sendo analisado e ao final do uso o participante deverá estar com os seguintes elementos definidos:
 - i. Objetivos por área de foco;
 - ii. Objetivo de estágio;
 - iii. Objetos de transferência de valor;
 - iv. Facilitadores;
 - v. Componentes de elementos do DevGo;
 - vi. Repositórios para monitoramento;
 - vii. Quais áreas de foco monitorar e por qual motivo;
 - viii. Conjunto de insights para monitoramento;
 - ix. Conjunto de lições aprendidas prioritárias.

6. Entrevista com o participante

- a. O pesquisador se guiará pelas perguntas descritas no seguinte link:
<https://forms.gle/FydnkGAtCkxeDQyE7>.

7. Finalização

- a. O pesquisador pergunta se o participante tem ainda algum comentário;
- b. O pesquisador agradece ao participante e informa que os resultados serão comunicados.