

DevGo: Um Modelo para Governança de Desenvolvedores em Ecossistema de Software Móvel a partir de *Developer Relations*

**Awdren Fontão^{1,2}, Rodrigo Pereira dos Santos³ (coorientador),
Arilo Claudio Dias-Neto¹ (orientador)**

¹Instituto de Computação – Universidade Federal do Amazonas (UFAM)

²Faculdade de Computação – Universidade Federal do Mato Grosso do Sul (UFMS)

³Departamento de Informática Aplicada
Universidade Federal do Estado do Rio de Janeiro (UNIRIO)

awdren@facom.ufms.br, rps@uniriotec.br, arilo@icomp.ufam.edu.br

Resumo. *Organizações que mantêm ecossistemas de software móvel (MSECO), como Apple e Google, têm investido em estratégias de governança de desenvolvedores com a formação de um time de Developer Relations (DevRel). Busca-se, com DevRel, estabelecer sinergia entre expectativas das comunidades de desenvolvedores externos e objetivos da organização. Porém, isto não é trivial: se a estratégia for insustentável, pode-se incorrer na “morte” do ecossistema. Por meio de estudos secundários na literatura, estudos primários envolvendo 68 profissionais de DevRel e estudos de mineração de repositórios de software, foi desenvolvido na tese um modelo para a governança de desenvolvedores em MSECO a partir de DevRel – 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.*

1. Visão Geral do Tema e Relevância da Pesquisa

Os relacionamentos entre desenvolvedores, empresas, usuários e organizações que mantêm uma plataforma tecnológica comum, 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].

No domínio de plataformas móveis, as organizações de software, como Apple e Google, têm investido em uma infraestrutura para engajar desenvolvedores externos a fim de contribuir para a expansão da plataforma. Por exemplo, em 2020, espera-se 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

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

aplicações móveis (*apps*) chamado Ecossistema de Software Móvel (MSECO, do inglês *Mobile Software Ecosystem*) [Fontão *et al.* 2015] [Steglich *et al.* 2019].

Do ponto de vista de negócios, MSECO consiste em um tipo de ECOS com estratégias híbridas, 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). Dentre os elementos de um MSECO, uma *app* consiste em um software que é embarcado em um dispositivo móvel (p. ex.: *smartphone*, *tablet*, *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 produz contribuições que podem ser, além das *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). Estima-se que há em torno de 12 milhões de desenvolvedores de aplicações móveis no mundo². Tais desenvolvedores possuem responsabilidades e um desempenho esperado pela organização central, de forma que o MSECO mantenha a capacidade de aumentar ou manter a comunidade ao longo do 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 têm 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.

Neste contexto, 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 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].

A governança de desenvolvedores ainda é um desafio para pesquisadores e profissionais que atuam na área de ecossistemas. No tocante à comunidade científica, há carência de pesquisa, formalização, teorias e modelos, ferramentas de apoio e estudos experimentais no assunto [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.

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

Para a indústria, o processo de expansão de uma plataforma de um ecossistema exige, por meio da governança de desenvolvedores, a 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 as metas do ecossistema [Huber *et al.* 2017].

Se a organização central não possuir estratégias (p. ex.: ferramentas, práticas, processos, modelos) sustentáveis para governar desenvolvedores, ela corre o risco de falhar no atendimento das demandas dos usuários. Uma governança de desenvolvedores insustentável ao longo do 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 podem passar por dificuldades ao tentar monitorar os efeitos das ações de governança de desenvolvedores [Mäenpää *et al.* 2017].

Tendo em vista o contexto, motivação e problemas apresentados anteriormente e a relação desta pesquisa de doutorado com os Grandes Desafios de Pesquisa em sistemas de Informação no Brasil 2016-2026, esta tese se posiciona em: “Sistemas de Informação e Desafios do Mundo Aberto”, mais especificamente no desafio sobre "Ecossistemas". Isto é embasado na parte do texto que trata da necessidade por pesquisa na complexidade que surge em sistemas que detêm características de conexão intra, inter e ócio-organizacional. Outro aspecto relevante está na competitividade entre as organizações e a reestruturação dos seus processos em um cenário aberto.

2. Questão de Pesquisa e Objetivos

A partir do que foi apresentado na seção anterior, esta pesquisa de doutorado buscou 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 um conjunto de oito questões para concepção do modelo: foco na concepção e refinamento do modelo – QP1 a QP7, foco na validação do modelo QP8 (adequação, controle, entendimento, generalização). [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?

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

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

repositórios para governança de desenvolvedores em MSECO e como tais 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.

As outras questões enunciadas foram: **[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 MSECO com base em questões técnicas em um repositório de Perguntas e Respostas (Q&A, do inglês *Questions and Answers*)?

Há uma indicação de que os repositórios podem ser utilizados [Farias *et al.* 2016], para extrair informações sobre a perspectiva socio-técnica e de negócios de um ECOS. Por isto, o repositório analisado para responder a QP7 foi da categoria de Perguntas & Respostas, mais especificamente, o Stack Overflow⁴.

A partir do que foi discutido como questão de pesquisa, o **objetivo desta tese** é:

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.

Para atingir o objetivo geral desta pesquisa, foram alcançados os seguintes resultados intermediários:

- **Definição de um corpo de conhecimento sobre Governança de Desenvolvedores em ECOS:** descoberta de um conjunto de estratégias para governar desenvolvedores identificadas na literatura técnica e análise da sua relevância para os profissionais da indústria;
- **Identificação e análise da estrutura para governança de desenvolvedores:** estabelecimento da estrutura do modelo a partir da perspectiva de DevRel;
- **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 às 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 ciclo de *feedback* entre organização central e desenvolvedores.

3. 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

⁴ <http://www.stackoverflow.com>

abordagens de governança de desenvolvedores em MSECO apesar de alguns dos trabalhos tratar aspectos de governança. A Tabela 1 apresenta os trabalhos relacionados, onde fica clara a necessidade pela identificação, descoberta e modelagem de uma estrutura que considere governança de desenvolvedores do ponto de vista de DevRel.

Tabela 1. Trabalhos relacionados.

Trabalho	Contribuição	Aspecto de Governança	Desenvolvedores? / MSECO?	Necessidade Indicada
Jensen e Scacchi (2010)	Práticas e processos para governança de projetos <i>open source</i>	Controle de desenvolvedores e coordenação de atividades	Não / Não	Modelar a governança de desenvolvedores que envolva a comunidade nos níveis micro e macro de quantidade de desenvolvedores.
Jergensen <i>et al.</i> (2011)	Modelo de migração de desenvolvedores <i>open source</i>	Fluxo de movimentação dos desenvolvedores	Sim / Não	Analisar a variedade de efeitos técnicos e sociais sobre os desenvolvedores
Qiu <i>et al.</i> (2011)	Modelo de governança entre desenvolvedores	Recursos e ambiente dos desenvolvedores	Sim / Não	Examinar ambientes, políticas de governança e recursos de outros ECOS e o impacto disso nos desenvolvedores
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	Garantir o amadurecimento dos desenvolvedores e identificar práticas de apoio a decisão, além de métodos de comunicação
Albert <i>et al.</i> (2013)	Abordagem para localização no mercado e mapeamento de relações	Recursos técnicos e organização consumidora	Não / Não	Analisar o mapeamento de relações em um ecossistema híbrido
Wareham <i>et al.</i> (2014)	Entendimento da governança de um ECOS	Estabilidade e evolução de um ECOS	Não / Não	Estudar as diferentes fases de evolução da governança
Sadi <i>et al.</i> (2015)	Abordagem para identificação de desenvolvedores	Projeto de Colaboração	Sim / Sim	Fornecer orientação específica sobre como o engajamento de desenvolvedores pode ser realizadas
Foerderer <i>et al.</i> (2018)	Análise de limites do conhecimento envolvidos em governança	Conhecimento em torno de recursos do ECOS	Não / Não	Analisar especificamente a governança de desenvolvedores do ponto de vista dos recursos utilizados
Kude <i>et al.</i> (2018)	Estratégias para recrutar, selecionar e treinar gerentes de parcerias	Programas de Parceria	Não / Não	Facilitar a conexão da organização com parceiros e setores internos
Vorraber <i>et al.</i> (2018)	<i>Framework</i> para alinhamento de objetivos entre desenvolvedores e gerentes de negócios	Mecanismos de criação de valor em torno de um projeto	Sim / Não	Criar uma visão holística de necessidades, valores e conexões dentro do ECOS na perspectiva dos desenvolvedores

4. Métodos Científicos Empregados

A metodologia de pesquisa utilizada foi elaborada a partir das questões de pesquisas descritas na Seção 2. Considerou-se na tese [Fontão 2019] um conjunto de métodos visando criar bases mais sólidas para a definição e validação do modelo. O modelo proposto na tese é chamado DevGo (do inglês, *Developer Governance*). A seguir, são descritos os métodos utilizados para ajudar a responder cada questão (entre parênteses), assim como os resultados obtidos.

Mapeamento Sistemático da Literatura (QP1): sobre Governança de Desenvolvedores em ECOS, seguindo os princípios para conduzir 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 verificados no Capítulo 2 da tese [Fontão 2019].

Resultado: extraiu-se, a partir de 91 artigos (2009-2018), um conjunto de definições e de 42 estratégias para governança de desenvolvedores em ECOS *open source*, proprietário e híbrido. Como foram extraídas a partir da literatura técnica, as estratégias e definições passaram por pesquisa de opinião (no estudo seguinte) para avaliação por profissionais de DevRel.

Pesquisa de Opinião (QP2): seguindo os procedimentos indicados por Linaker *et al.* (2015) e Molléri *et al.* (2016), com 18 profissionais de DevRel, verificando a relevância das estratégias identificadas anteriormente. As informações referentes a esse estudo estão presentes no Capítulo 3, Seção 3.2, da tese [Fontão 2019].

Resultado: um conjunto de 13 estratégias consideradas relevantes (nível de relevância igual ou maior que a mediana calculada de 74%) pelos profissionais de DevRel. Este conjunto foi utilizado para a construção da primeira versão do modelo DevGo.

Entrevistas (QP3): um conjunto de entrevistas com 15 profissionais de DevRel como forma de análise e refinamento do modelo DevGo, incluindo a definição de governança de desenvolvedores. As informações referentes a esse estudo estão presentes no Capítulo 3, Seção 3.3, da tese [Fontão 2019].

Resultado: foram obtidas 28 sugestões para o refinamento do DevGo. Com este estudo, foram inseridas no DevGo as áreas de foco. Além disso, as fases foram redistribuídas em níveis de avanço (estágios). A troca de valor entre as áreas de foco passou a ser representada na segunda versão do DevGo (Figura 4 na Tese [Fontão 2019]).

Mapeamento da Literatura Cinza (QP4): visando realizar a análise de 100 artigos de profissionais com experiência comprovada em DevRel no *Medium*⁵, portal que contém artigos técnicos. Para isto, utilizou-se de procedimentos de mapeamento sistemático da literatura cinza propostos por Garousi *et al.* (2016). As informações referentes a esse estudo estão presentes no Capítulo 3, Seção 3.4, da tese [Fontão 2019].

Resultado: um conjunto de 63 lições aprendidas classificadas em cada estágio do modelo DevGo: sensibilização (17), entrada (7), ativação (7), retenção (10), reconhecimento (11) e referência (11).

Estudo de Caso Exploratório (QP5): para analisar a estrutura de 19 repositórios (internos e externos) de MSECO seguido de uma revisão por pares com quatro especialistas a fim de caracterizar os repositórios em relação às suas características comuns. As informações referentes a esse estudo estão presentes no Capítulo 3, Seção 3.5, da tese [Fontão 2019].

Resultado: um conjunto de oito categorias agrupando um conjunto de repositórios internos e externos à organização central: (1) Perguntas e Respostas; (2) Lista de E-mails e Fóruns; (3) Ambiente Social de Codificação; (4) Website Social de Notícias; (5)

⁵ <http://medium.com>

Redes Sociais; (6) Comunicação do Time; (7) Loja de Aplicações; e (8) Painel de Controle do Desenvolvedor.

Os resultados obtidos em QP4 e QP5 ajudaram no refinamento do DevGo, gerando assim a terceira versão do DevGo (Figura 8 na Tese [Fontão 2019]). O modelo passou a conter as lições aprendidas por estágios e as categorias de repositórios para monitoramento.

Pesquisa de Opinião (QP6): com 31 profissionais visando obter as 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. As informações referentes a esse estudo estão presentes no Capítulo 3, Seção 3.6, da tese [Fontão 2019].

Resultado: 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; e (4) Complementaridade. Este estudo aprimorou os objetos de transferência de valor do DevGo contribuindo para o refinamento e geração da quarta versão do modelo (Figura 12 na tese [Fontão 2019]).

Mineração de Repositórios de Software – MRS (QP7): a MRS foca em descobrir informação útil sobre os elementos relacionados a software por meio da extração e análise de dados de repositórios. Na tese, foram utilizados mecanismos de MRS para obter informações sobre o desenvolvedor a partir das outras duas dimensões de análise de um ecossistema. Foi gerado um *dataset* do *Stack Overflow* focado em plataformas móveis, envolvendo 1.568.377 postagens de Janeiro de 2008 a Dezembro de 2017 para os MSECO Android, iOS e Windows. Esta etapa da pesquisa buscou explorar as dimensões técnica e social de um ECOS, segundo o *framework* ReuseECOS [Santos e Werner 2012], como discutido anteriormente para a dimensão de negócios. As informações completas do estudo, incluindo os insights a partir de MRS, estão presentes no Capítulo 4 da tese [Fontão 2019].

- **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: um conjunto de seis *insights* obtido a partir da análise dos resultados que aprimoram a área de foco “Monitoramento” do DevGo. Eles também servem para

indicar procedimentos de análise e mineração que podem ser utilizados em outras categorias de repositórios (além de Questões & Respostas). Além disso, foi realizada a identificação e compreensão das emoções primárias envolvidas no contexto de MSECO. Isto contribuiu para aprimorar o modelo DevGo para a quinta versão (final) que é descrita no Capítulo 5 da tese [Fontão 2019].

Estudo de Caso Confirmatório (QP8): este estudo envolveu entrevistas como forma de coleta de dados. Seguiu-se os procedimentos de Runeson e Host (2009) e de Yin (2009). Participaram deste estudo cinco profissionais de DevRel que atuam na área e representam organizações distintas. Os participantes utilizaram o DevGo durante atividades de DevRel focadas em retenção de desenvolvedores para avaliar com relação aos critérios de adequação, controle, entendimento e generalidade. O planejamento, execução e análise dos resultados são descritos no Capítulo 6 da tese [Fontão 2019].

Resultado: os elementos selecionados pelos participantes cobrem 93% dos disponíveis no DevGo, para o estágio de retenção. Os profissionais de DevRel destacaram como benefícios: a estrutura que facilita o entendimento da governança de desenvolvedores, dos objetos de transferência de valor, inclusive, para comunicar a importância de DevRel para os setores da organização e destacaram a área de foco “Monitoramento” como o diferencial. Como limitações, indicaram: a melhoria da indicação de possíveis papéis na área de foco “DevRel (Evangelismo e Advocacia)”; e como comunicar o DevGo dentro da organização, incluindo a efetividade dos facilitadores.

5. O modelo DevGo (*DEVELOPER GOVERNANCE*)

DevGo consiste em um modelo composto por 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 1**, 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 2**. Exemplos de elementos que compõem o modelo são abordados nas Subseções 5.1-5.3.

O DevGo, como indicado na **Figura 2**, é composto por **Áreas de foco**, que indicam as áreas que uma organização central precisa cuidar para governar desenvolvedores e manter padrões de estratégias de DevRel. 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.

Cada uma das fases é composta por **Estágios**, que compreendem um período de desenvolvimento do desenvolvedor. Um estágio é formado 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 (Evangelismo e Advocacia), Fluxo de Avanço do Desenvolvedor e, por fim, Monitoramento.** Estas áreas de foco dão suporte a uma abordagem estruturada e/ou descentralizada (orgânica) para governar desenvolvedores.

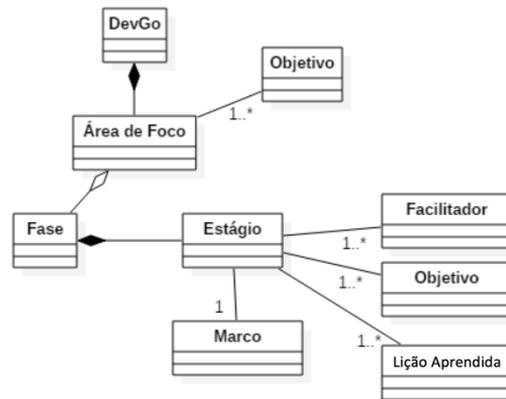


Figura 1. Estrutura do modelo DevGo.

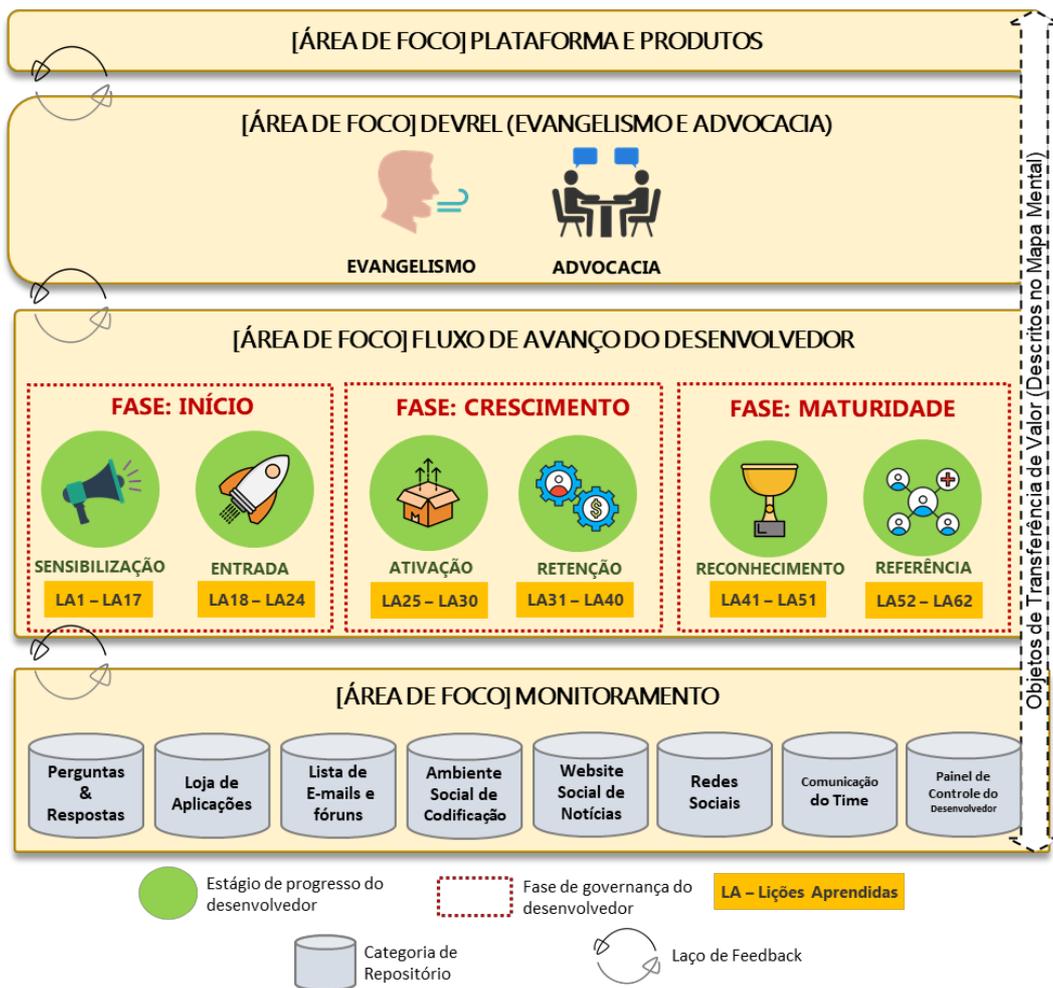


Figura 2. O modelo DevGo.

O conjunto de setas no modelo representam objetos de transferência de valor entre as áreas de foco. Para cada área de foco, são apresentados mapas mentais, onde é possível identificar alguns desses objetos de transferência entre elas. Os objetos de transferência de valor são mecanismos de criação de valor para gerar e distribuir valor para todo o ecossistema com base em inovação, investimentos e compartilhamento de custos. Neste artigo, é apresentada, como exemplo, a área de foco “Fluxo de Avanço do Desenvolvedor”. A descrição do DevGo é parte do Capítulo 5 da tese [Fontão 2019].

5.1. A área de foco “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 as suas estratégias de atuação para governá-los.

De fato, se há muitos desenvolvedores com problemas para entrar no ecossistema, a organização passará por problemas de ativação e retenção, por exemplo (estágios necessários para a geração de contribuições com qualidade). A **Figura 3** apresenta os objetos que ajudam na transferência de valor nesta área de foco (os consumidos e os providos pela área).

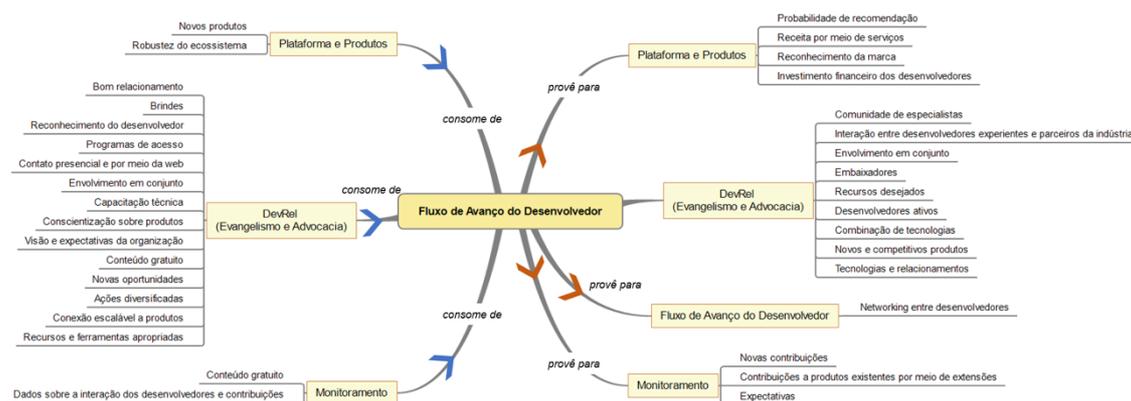


Figura 3. Objetos de transferência de valor.

Especificamente nesta área, há um conjunto de fases, como explicado na estrutura do modelo (**Figura 1**). Há três fases para apoiar o progresso do desenvolvedor e tais fases consideram modelos de negócio e de gerenciamento de parceria. Tem-se, desta forma, as seguintes fases: Início, Crescimento e Maturidade. Neste artigo, é utilizada como exemplo a fase de Maturidade, conforme a subseção a seguir.

5.2. A fase de “Maturidade”

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

Nesta fase, um desenvolvedor é referência para os outros desenvolvedores do ecossistema e precisa de uma ligação direta com a organização central por meio da equipe de DevRel. Cada fase compreende um conjunto de **Estágios de Avanço do**

Desenvolvedor. Cada um destes estágios colabora para um laço de *feedback* que aumenta o conhecimento tanto da organização sobre o MSECO quanto dos desenvolvedores. Este laço de *feedback* apoia a organização central e os desenvolvedores por meio da solução de problemas e da redução de riscos (i.e., situação de cooperação). O laço de *feedback* consiste no exercício contínuo de coletar as percepções e expectativas dos desenvolvedores a fim de facilitar que sejam respondidas. Esta resposta pode ser feita 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 eles e os profissionais de DevRel. Em 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.

Esta fase de “Maturidade” compreende um estágio de reconhecimento do desenvolvedor (Estágio de Reconhecimento) e um estágio que ajuda a identificar e preparar desenvolvedores que são referências para o ecossistema (Estágio de Referência. Na próxima subseção, o Estágio de Referência é apresentado, incluindo um conjunto de lições aprendidas, os facilitadores e o marco.

5.3. O 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 desenvolvedores sobre a plataforma – neste cenário, relacionamentos são compromissos bidirecionais e representam um investimento do desenvolvedor. Como facilitadores para este estágio, tem-se: 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 mais próximo com os profissionais de DevRel.

Para este estágio, há um conjunto de 11 lições aprendidas, das quais apresentamos três neste artigo:

LA56. Defina um manual de eventos, uma vez que, como o influenciador de alguma forma fala do nome da sua empresa usando seus materiais, ele também deve ter metas a serem alcançadas;

LA57. Peça aos influenciadores para submeterem palestras a conferências técnicas para que eles entreguem apresentações técnicas e ajudem na divulgação do ecossistema;

LA58. Envolver os influenciadores na codificação de demonstrações técnicas de produtos do ecossistema. Isto ajuda na capacitação do influenciador;

6. Conclusões

O DevGo descreve como as relações ganha-ganha podem ajudar a tornar o ecossistema sustentável por meio de dois elementos-chave – desenvolvedores e organização central –, a partir do suporte de uma 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 contém a “voz dos desenvolvedores” e os objetivos da organização central. O loop de “feedback” entre as áreas de foco ainda ajuda a tornar o DevGo extensível a futuras mudanças, inclusive, com influência da área de foco “Monitoramento”. O uso de repositórios no DevGo pode formar a base de mecanismos e ferramentas de avaliação maduros para ECOS.

O DevGo foi avaliado em estudos de caso envolvendo 5 profissionais de DevRel de empresas diferentes e que atuaram/atua como: evangelista, *developer advocate*, líder de DevRel, gerente de DevRel. 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?”. O estágio de retenção foi o foco devido, no corpo de conhecimento, ter sido identificado que é o estágio mais desafiador para DevRel. Todo planejamento, execução e análise de resultados é apresentado no Capítulo 6 da Tese (p. 126-147). Foram ainda capturados benefícios e limitações percebidos pelos profissionais durante o uso do DevGo. Os elementos selecionados do 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 nas áreas de foco de DevRel (“Evangelismo” e “Advocacia”) os papéis de DevRel, além de considerar a maturidade do profissional dentro da área de foco.

Os elementos selecionados pelos participantes cobrem 93% daqueles disponíveis no DevGo, para o estágio de Retenção. Os profissionais de DevRel destacaram como benefícios: (1) a estrutura que facilita o entendimento da governança de desenvolvedores; (2) os objetos de transferência de valor, inclusive, para comunicar a importância da área para a organização; (3) a área de monitoramento. Como limitações, indicaram: melhoria da indicação de possíveis papéis na área de foco de DevRel (Evangelismo e Advocacia) e como comunicar o DevGo dentro da organização. Como conclusões obtidas, 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 foco “Monitoramento” para formar uma estrutura sólida de mecanismos de avaliação e ferramentas para MSECO com uma base mais extensa em volume de desenvolvedores; (5) Introdução de estratégias inovadoras para melhor atender às metas da organização: apoio às organizações em evoluir estratégias de governança de desenvolvedores, dimensionar esforços e formar equipes internas.

As limitações desta pesquisa envolvem a amostra de participantes, o tipo de ECOS analisado e o repositório usado para extrair as expectativas dos desenvolvedores: (1) 68 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; (2) O modelo DevGo não é um modelo de referência para ECOS pelo fato de sua definição, refinamento e avaliação considerarem MSECO. Logo, há a necessidade de evoluir o modelo considerando outros tipos de ECOS; (3) Os *insights* monitoramento foram propostos em cima da categoria de “Perguntas e Respostas” da área de foco “Monitoramento”. As outras categorias poderiam ser estudadas para a identificação de outros *insights*; (4) O DevGo precisa ainda de um guia para sua utilização.

Percebeu-se algumas oportunidades de pesquisas futuras, de forma geral, nas seguintes áreas: (1) **Maturidade de Governança de Desenvolvedores:** à medida que se tenha um estabelecimento mais amplo do 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 necessária 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, a partir do ponto de vista de 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 foco “DevRel (Evangelismo e Advocacia)”, por exemplo, como indicado em alguns dos estudos, precisa ser aprimorada em relação a indicação de papéis existentes dentro de evangelismo e advocacia: como, por exemplo, evangelistas, engenheiros de parceria, *advocates* e papéis de gestão de times de DevRel.

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 profissionais de DevRel. Isso reforça o fato de que é um desafio demonstrar aos setores organizacionais o motivo de investir em DevRel.

O monitoramento da governança de desenvolvedores, a partir de cada categoria de repositórios, é outra área que pode ser expandida para pesquisas futuras. Por exemplo, o estudo de séries temporais da alteração de emoções dos desenvolvedores durante o envolvimento no ecossistema e a análise cruzada entre os repositórios para gerar *insights* mais precisos. Isto poderá gerar um painel de controle sobre a governança de desenvolvedores para apoiar na análise global e de subsidiárias das organizações.

Outro aspecto relevante é formação de profissionais para atuar com DevRel dentro dos cursos de Computação oferecidos em universidades. A análise dos currículos atuais para verificar se favorecem à formação de novos profissionais é um trabalho de interesse. É evidente que o profissional de DevRel, precisa desenvolver habilidades que envolvam os aspectos da computação, da engenharia de sistemas e de negócios.

7. Publicações e Premiações

Nesta seção, são descritas as contribuições (Tabela 2) que envolvem artigos científicos em conferências, periódicos e workshops. O autor se envolveu em ações como revisor de periódicos (International Journal of Business Information Systems, iSys, JSERD e IEEE Access) e conferências (p. ex: ICGSE'20, SBSI'20, FIE'19, ICH'19, SBGAMES'19, WAIHCWS'19, WDES'20, SBQS'19, SBSI'19). Por fim, o autor se envolveu em atividades docentes: orientações (5 TCCs) e professor substituto (UFAM).

Tabela 2. Publicações em conferências, periódicos e capítulos de livro.

Artigo em Conferência	<ol style="list-style-type: none"> 1. GoDev-DX: Governança de Desenvolvedores em Ecossistemas de Software Móvel a partir da Experiência do Desenvolvedor (ELA-ES 2016); 2. Investigating factors that influence developers experience in mobile software ecosystem (WDES/ICSE 2017, Qualis B4); 3. Towards a guideline-based approach to govern developers in mobile software ecosystems (ICSR 2017, Qualis B2); 4. Facing up the primary emotions in Mobile Software Ecosystems from Developer Experience Towards a Guideline-Based Approach to Govern Developers in Mobile Software Ecosystems (WASHES 2017); 5. Hearing the Voice of Developers in Mobile Software Ecosystems (SBES 2017, Qualis B2); 6. Challenges in Model Based Testing for Mobile Applications [Colaboração - coautor] (CIbSE 2017, Qualis B4); 7. Tester Experience: Concept, Issues and Definition [Colaboração - coautor] (COMPSAC 2017, Qualis A2); 8. Mobile Application Development Training in Mobile Software Ecosystem: Investigating the Developer eXperience (SBQS 2018, Qualis B3); 9. Which factors affect the evangelist's support during training sessions in Mobile Software Ecosystems? (WDES/ECSA 2018, Qualis B4); 10. Governança de desenvolvedores em Ecossistemas de Software Móvel (WTDSof/CBSoft 2018); 11. Balancing theory and practice in software engineering education – A PBL, toolset based approach (Frontiers In Education 2019, Qualis B1); 12. How are my students going? A tool to analyse students' interactions on capstone courses [Colaboração – coautor] (SBIE 2019, Qualis B1); 13. Investigating Asset Governance Mechanisms in a Proprietary Software Ecosystem [Colaboração – coautor] (SBSI 2020, Qualis B2); 14. Combining Design Thinking and Scrum to enhance Test Suite during innovative software development [Colaboração – coautor] (CIbSE 2020, Qualis B4) 15. Understanding Developer Relations (DevRel) practitioners' motivations for using software repositories (CHASE 2020 – aceito pra publicação, Qualis B1); 16. On value creation in Developer Relations (DevRel): a practitioners' perspective (ICGSE 2020 – aceito pra publicação, Qualis B1)
Artigo em Periódico	<ol style="list-style-type: none"> 1. Exploiting Repositories in Mobile Software Ecosystems from a Governance Perspective (Information Systems Frontiers, 2019, Qualis A1) 2. Supporting governance of mobile application developers from mining and analyzing technical questions in stack overflow (JSERD, 2018, Qualis B2).
Capítulo de Livro	<ol style="list-style-type: none"> 1. Governança de desenvolvedores em Ecossistemas de Software (Minicursos do SBSI 2018); 2. Ecossistemas de Software (Computadores e Sociedade da Série Elsevier – SBC).

Minicursos	<p>1. Certificação de <i>Apps</i> em Ecossistemas de Software Móvel (CBSOFT 2017);</p> <p>2. Governança de Desenvolvedores em Ecossistemas de Software (SBSI 2018).</p>
-------------------	---

As premiações obtidas durante o doutorado também fazem parte desta seção: (1) Melhor artigo (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 (SBES); (3) Reconhecimento pela produtividade no biênio 2017-2018, Instituto de Computação/UFAM; e (4) Reconhecimento e escolha como a quinquagésima tese de doutorado do PPGI/UFAM.

Referências

- Albert, B. E., Dos Santos, R. P. e Werner, C. M. (2013). Software ecosystems governance to enable IT architecture based on software asset management. In *IEEE International Conference on Digital Ecosystems and Technologies*, p. 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*, n. January, p. 215-226.
- 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. A. [Orgs.]. *Software Ecosystems. Analyzing and managing business networks in the software industry*. Edward Elgar Publishing Limited. p. 59-78.
- Bosch, J. (2009). From Software Product Lines to Software Ecosystems. In *SPLC '09 Proceedings of the 13th International Software Product Line Conference*, p. 111-119.
- Ekwoje, O. M., Fontao, A. e Dias-Neto, A. C. (2017). Tester Experience: Concept, Issues and Definition. In *Proceedings - International Computer Software and Applications Conference*, p. 208-213.
- Farias, M. A. de F., Novais, R., Colaço, M., *et al.* (2016). A Systematic Mapping Study on Mining Software Repositories. In *31st ACM/SIGAPP Symposium on Applied Computing*, p. 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*, n. October 2015, p. 1-26.
- Fontão, A., Santos, R. P. e Dias-neto, A. C. (2015). Mobile Software Ecosystem (MSECO): A Systematic Mapping Study. In *The 39th Annual International Computers, Software & Applications Conference*, p. 653-658.
- Fontão, Awdren de Lima. DevGo: um modelo para governança de desenvolvedores em ecossistema de software móvel a partir de developer relations. 2019. 182 f. Tese (Doutorado em Informática) - Universidade Federal do Amazonas, Manaus, 2019.
- Franco-Bedoya, O., Ameller, D., Costal, D. e Franch, X. (2017). Open source software ecosystems: A Systematic mapping. *Information and Software Technology*, v. 91, p. 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*, p. 26.
- Jansen, S. e Cusumano, M. A. (2012). Defining software ecosystems: A survey of software platforms and business network governance. In *Proceedings of the international Workshop on*

Software Ecosystems 2012.

Jensen, C. e Scacchi, W. (2010). Governance in open source software development projects: A comparative multi-level analysis. *Open Source Software: New Horizons*, p. 1–15.

Jergensen, C., Sarma, A. e Wagstrom, P. (2011). The Onion Patch: Migration in Open Source Ecosystems. In *Foundations on Software Engineering*, p. 70-80.

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, n. 7, p. 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*.

Kude, T. (2018). Successfully governing software ecosystems : Competence profiles of partnership managers. *IEEE Software*, v. 1, p. 1.

Mäenpää, H., Munezero, M., Fagerholm, F. e Mikkonen, T. (2017). The many hats and the broken binoculars. In *Proceedings of the 13th International Symposium on Open Collaboration - OpenSym '17*, p. 1-9.

Manikas, K. (2016). Revisiting software ecosystems Research: A longitudinal literature study. *Journal of Systems and Software*, v. 117, p. 84–103.

Manikas, K. e Hansen, K. M. (2013a). Reviewing the Health of Software Ecosystems – A Conceptual Framework Proposal. In *Proceedings of the 5th International Workshop on Software Ecosystems*, p. 33-44.

Manikas, K. e Hansen, K. M. (2013b). Software ecosystems-A systematic literature review. *Journal of Systems and Software*, v. 86, n. 5, p. 1294–1306.

Morisio, M., Ezran, M. e Tully, C. (abr 2002). Success and failure factors in software reuse. *Software Engineering, IEEE Transactions on*, v. 28, n. 4, p. 340–357.

Qiu, Y., Gopal, A. e Hann, I. (2011). Synthesizing Professional and Market Logics: a Study of Independent Ios App Entrepreneurs. In *ICIS*, p. 10.

Sadi, M. H., Dai, J. e Yu, E. (2015). Designing Software Ecosystems: How to Develop Sustainable Collaborations? - Scenarios from Apple iOS and Google Android. *Advanced Information Systems Engineering Workshops - CAiSE 2015 International Workshops, Stockholm, Sweden, June 8-9, 2015, Proceedings*, v. 215, p. 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, n. 8, p. 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).

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, n. 4, p. 1195–1215.

Yin, R.K. (2009). How to do better case studies. *The SAGE handbook of applied social research methods*, 2, p.254-282.