

Os Aspectos Sociais dos Ecossistemas de Software

Renato Ferreira, Müller Miranda
Universidade Federal do Pará
{renpina, mulgsm}@gmail.com

Fernando Figueira Filho, Christoph Treude
Universidade Federal do Rio Grande do Norte
fernando@dimap.ufrn.br, ctreude@gmail.com

Cleudson R. B. de Souza
Universidade Federal do Pará & Instituto Tecnológico Vale
cleudson.desouza@acm.org

Leif Singer
University of Victoria
lsinger@uvic.ca

ABSTRACT

Software ecosystems have become a new paradigm for the development of large-scale software, bringing a complex mixture of technical, business and social aspects. Although previous research has focused on understanding both the technical and business aspects, the social side of software ecosystems is less understood. To fill this gap, this paper presents the results of an empirical study aimed at understanding the influence of social aspects on the adoption and permanence of developers in software ecosystems. We conducted 25 interviews with software developers and we distributed an online questionnaire which included responses from 83 developers of mobile software applications. Using grounded theory techniques, we identified a complex social system based on continuous interaction and mutual support between different actors, including internal and external developers to the ecosystem, community developers, end users and businesses. Our results highlight the importance of social aspects in software ecosystems during the initial phase of adoption, as well as the continued participation of developers.

Categories and Subject Descriptors

H.1.2. [User/Machine Systems]: Human factors.

General Terms

Human Factors, Design.

Keywords

Ecossistemas de Software, Adoção, Permanência, Android, iOS, Aplicações Móveis, Plataformas.

1. INTRODUÇÃO

A natureza colaborativa de desenvolvimento de software tem sido reconhecida e investigada desde o início dos estudos em CSCW. Diferentes ferramentas têm sido propostas [3, 5, 23], estudos empíricos têm sido realizados [12, 4] e abordagens têm sido desenvolvidas [7]. Esta área tem recebido muita atenção tanto da comunidade de CSCW, como de comunidades de pesquisa em engenharia de software.

Ecossistemas de software são definidos por Jansen et al. [17] como “um conjunto de atores que funcionam como uma unidade e interagem em um mercado distribuído entre software e serviços, juntamente com as relações entre essas partes. Estas relações são frequentemente apoiadas por uma plataforma tecnológica comum e operam através da troca de informações, recursos e artefatos”. A área de ecossistemas de software ampliou o foco de estudos empíricos anteriores para abranger as relações entre os diferentes atores que dependem de serviços prestados por uma plataforma de software comum. Ecossistemas de software, portanto, envolvem

um conjunto muito mais amplo de atores, incluindo serviços, infraestrutura, usuários e desenvolvedores.

Em vez de depender exclusivamente de interações intra-organizacionais, a abordagem de ecossistemas de software precisa levar em conta as interações inter-organizacionais, bem como uma “perspectiva da comunidade”, de modo que os atores têm uma maneira centrada na comunidade para colaborar e coordenar o trabalho [6]. Um exemplo de um ecossistema de software é a plataforma iOS. Neste caso, Apple oferece o sistema operacional iOS, bem como um grande conjunto de interfaces de programação de aplicações (APIs) sobre as quais extensões (*apps*) são construídas. Desenvolvedores de software externos usam APIs da Apple para desenvolver seus próprios aplicativos (*apps*) que serão executados no iOS, e, conseqüentemente, em iPhones e/ou iPads. Outros exemplos de ecossistemas bem-sucedidos em diferentes domínios incluem Android, Facebook, Windows Kinect, SAP, e Linux.

Iansiti e Levien [16] cunharam o termo *keystone* para rotular atores centrais em um ecossistema de software. Por exemplo, no ecossistema iOS, a *keystone* é a Apple, a empresa que controla o ecossistema, fornecendo a plataforma para que os aplicativos sejam construídos juntamente com mecanismos para a distribuição desses aplicativos aos usuários finais. As *keystones* fazem um esforço deliberado para atrair e manter desenvolvedores de software externos¹, assim como promovem a disponibilidade de aplicativos para a plataforma com o objetivo de aumentar a sua atratividade para os usuários finais.

Atrair e manter os desenvolvedores para um ecossistema pode fazer uma diferença crucial para empresas concorrentes no mercado, especialmente se a participação de desenvolvedores é sustentada, ou seja, se desenvolvedores adotam uma plataforma e continuam desenvolvendo para ela. A sustentabilidade de um ecossistema é um aspecto tão importante que há um interesse crescente sobre este aspecto na literatura. Dhungana et al. [13] definem um ecossistema de software sustentável como aquele que mantém ou aumenta a sua comunidade de usuários e/ou desenvolvedores durante longos períodos de tempo e pode sobreviver mudanças inerentes, tais como novas tecnologias ou novos produtos. Os trabalhos anteriores focam em aspectos de negócio de ecossistemas sustentáveis, incluindo os aspectos táticos e estratégicos que devem ser levados em conta na concepção de um ecossistema de software [27], como por exemplo, o grau de abertura do mesmo. Abertura é definida como o grau em que a plataforma impõe menos restrições à

¹ Neste trabalho, o termo desenvolvedores *internos* é usado para desenvolvedores que trabalham para a empresa *keystone*, enquanto que o termo desenvolvedores *externos* é usado para todos os outros.

participação, desenvolvimento ou uso, independente do ator, seja desenvolvedor ou usuário final [14]. Entretanto, a maioria dos trabalhos anteriores sobre a sustentabilidade de ecossistemas não leva em conta o ponto de vista dos desenvolvedores externos ao ecossistema.

Este trabalho visa preencher esta lacuna ao compreender as razões dos desenvolvedores para a escolha de um ecossistema de software e os desafios que eles enfrentam ao adotar e permanecer desenvolvendo para o mesmo. Dado o foco deste trabalho, decidiu-se usar a teoria de difusão de inovação de Rogers [26] como um ponto de partida para esta pesquisa. Para Rogers, uma inovação é “qualquer ideia, prática ou projeto que é percebido como novo pela unidade de adoção”. Assim, para um desenvolvedor que nunca desenvolveu uma aplicação móvel, Android e iOS podem ser considerados como inovações.

Este artigo apresenta os resultados de um estudo empírico cujos dados foram coletados em duas fases, uma exploratória e uma de confirmação. Na primeira fase, nove entrevistas semiestruturadas foram realizadas com desenvolvedores de software e distribuiu-se um questionário online que contou com respostas de 83 desenvolvedores de software para dispositivos móveis. As entrevistas e as respostas às perguntas do questionário foram codificadas utilizando técnicas da teoria fundamentada em dados [8, 10]. Os resultados sugerem a importância de aspectos sociais na adoção e na permanência de um desenvolvedor em um determinado ecossistema de software, além de aspectos comerciais e técnicos. A fim de confirmar a importância dos aspectos sociais, 16 entrevistas semiestruturadas adicionais centradas exclusivamente nesses aspectos foram realizadas. Os resultados foram novamente codificados e integrados com os resultados iniciais, que apontam para um sistema social complexo baseado na interação contínua e no apoio mútuo entre diferentes atores, revelando assim a importância dos aspectos sociais em ecossistemas de software, tanto durante a fase inicial de adoção quanto para a permanência dos desenvolvedores.

O restante deste artigo está organizado da seguinte forma. Na Seção 2, é apresentada uma visão geral sobre ecossistemas de software. Na Seção 3, é apresentada a metodologia que foi usada para coleta e análise dos dados. Na Seção 4, os resultados são apresentados. Na Seção 5 apresenta-se a discussão dos resultados no contexto da literatura. Por fim, na Seção 6 são apresentadas as considerações finais e ideias para trabalhos futuros.

2. ECOSISTEMAS DE SOFTWARE

A definição de ecossistemas de software apresentada na introdução deste estudo mostra que o ecossistema é apoiado por uma plataforma tecnológica comum para troca de informações, recursos e artefatos. Esta plataforma tecnológica pode ser única, como o Android, do Google, ou basear-se em um conjunto de projetos interdependentes menores, como Apache, que é composta por cerca de 150 projetos. Para este artigo, denomina-se os ecossistemas do primeiro tipo simplesmente de ecossistema de plataforma (“*platform ecosystems*”), enquanto que o segundo tipo será considerado como um ecossistema de projeto (“*project ecosystems*”). Muitas vezes, um ecossistema de projeto é baseado em projetos de código aberto. A distinção entre estes dois tipos de ecossistemas pode ser explicada em termos da dominância da plataforma. A dominância da plataforma refere-se à existência de um grande projeto de software (plataforma) no topo do qual as extensões são construídas. Android do Google é um ecossistema

de plataforma dominante. Em contrapartida, o ecossistema da Apache é centrado no servidor Apache HTTP, mas há cerca de 150 outros projetos relevantes. Neste trabalho, o foco está nos aspectos sociais dos ecossistemas de plataforma.

Barbosa et al. [2] apresentaram uma classificação dos diferentes aspectos dos ecossistemas de software, não importando se são ecossistemas de plataforma ou ecossistemas baseados em projetos. Eles argumentam que os ecossistemas de software precisam ser entendidos em três dimensões: a técnica, de negócio e a social. A dimensão técnica está relacionada com a plataforma de software e a infraestrutura na qual o ecossistema está incorporado; que incide sobre os aspectos arquitetônicos do ecossistema, dentre outros aspectos. A dimensão de negócio está relacionada com modelos de negócios, estratégias de licenciamento e de parceria, entre outros aspectos financeiros. Por fim, a dimensão social se concentra em como os diferentes atores do ecossistema se relacionam entre si, a fim de atingir seus objetivos. Uma classificação semelhante é proposta por Manikas e Hansen [22]. Eles classificam os ecossistemas de engenharia de software, ecossistemas de negócios e gestão, e relacionamentos de ecossistemas, mas acredita-se que esta classificação é equivalente às dimensões técnica, de negócio e sociais de Barbosa et al. [2].

Esta classificação é conceitual, ou seja, há estudos que se concentram em mais de um aspecto ao mesmo tempo. Por exemplo, o trabalho de Koch e Kerschbaum [20] enfatiza as motivações de desenvolvedores de software para participar de um ecossistema de software para dispositivos móveis particular. Eles relatam que a maior motivação para os desenvolvedores é a diversão, o estímulo intelectual e o aprendizado de novas habilidades. O potencial ganho financeiro em participar de um ecossistema de software não é tão relevante. Em outras palavras, este trabalho foca no negócio e nos aspectos pessoais que motivam a participação em ecossistemas de software. Da mesma forma, Karhu et al. [19] sugerem que diferentes plataformas móveis estabelecem diferentes relações com seus *stakeholders*: alguns deles se baseiam mais na competição, enquanto outros são mais colaborativos. Este trabalho foca nos aspectos sociais dos ecossistemas de software. Neste caso, o social é relacionado com as relações entre empresas, e não desenvolvedores individuais.

Estas classificações estão relacionadas aos dois tipos de ecossistemas: plataforma ou baseado em projetos. Mais importante, a literatura que Barbosa et al. [2], Manikas e Hansen [22] identificaram em suas revisões sobre os aspectos sociais de ecossistemas de software é limitada de duas maneiras. Primeiramente, esses estudos focam principalmente sobre os aspectos sociais dos ecossistemas de projeto. Por exemplo, relataram trabalhos tais como o de Jergensen et al. [18] que estavam interessados em compreender a participação progressiva no ecossistema de projeto GNOME (open-source). Outro exemplo identificado nos estudos é o trabalho de Alves e Pessoa [1], que estudou diferentes projetos brasileiros de código aberto. A segunda limitação identificada nestes trabalhos é que o estudo dos aspectos sociais dos ecossistemas de plataforma é realizado a um nível muito alto de abstração. Um exemplo, neste caso, seria o trabalho de Dhungana et al. [13] que contrasta os ecossistemas de software com os ecossistemas naturais usando o Eclipse como seu exemplo, mas apenas a nível conceitual, sem dados empíricos. Yu e Deng [33] estudam as “dependências estratégicas [relações] entre o fornecedor de software, desenvolvedores externos, e os utilizadores finais” com o objetivo de explorar e raciocinar sobre formas alternativas para alcançar os objetivos estratégicos para

cada ator. Isso é feito ao nível conceitual, ou seja, sem dados empíricos. Em suma, de acordo com nosso conhecimento, não existem estudos empíricos sobre os aspectos sociais dos ecossistemas de plataforma.

Outro aspecto importante no estudo dos ecossistemas de software refere-se à sua sustentabilidade. Em outras palavras, o ecossistema de software deve ser capaz de atrair desenvolvedores para desenvolver novas aplicações e mantê-los ao apoiar estas aplicações. Este aspecto é fundamental e, de fato, há um interesse crescente sobre o assunto na literatura. Usando a classificação de Barbosa et al. [2], foram identificados trabalhos relacionados que incidem sobre os aspectos de negócio, incluindo aspectos táticos e estratégias que devem ser levadas em conta na concepção de um ecossistema de software [27]. Também identificou-se aspectos técnicos, incluindo a sua flexibilidade, por exemplo, a capacidade de introduzir mudanças em um projeto para construir uma nova funcionalidade [11]. Por outro lado, identificar e compreender a influência dos aspectos sociais relacionados com a sustentabilidade de um ecossistema de software é um tema pouco abordado na literatura. Assim, a questão de pesquisa deste artigo é: *como aspectos sociais influenciam a sustentabilidade de um ecossistema de software?*

3. METODOLOGIA

A fim de examinar a questão de pesquisa descrita na seção anterior um estudo qualitativo de duas fases foi realizado: a primeira fase foi exploratória, e a segunda foi confirmatória. Os resultados do estudo exploratório sugeriram o importante papel dos aspectos sociais na adoção e na permanência de um desenvolvedor em um ecossistema de software em particular, além de outros aspectos técnicos e de negócio. A fase de confirmação explorou o aspecto social em mais detalhes. Os resultados são baseados em entrevistas semiestruturadas e nas respostas para questões abertas de um questionário enviado para desenvolvedores de software.

3.1 Exploratória – Local

Inicialmente, foram realizadas entrevistas semiestruturadas com nove desenvolvedores (oito homens e uma mulher) que ativamente desenvolviam para Android (5), iOS (3), ou ambos (1). Dois eram alunos, um de curso de graduação e outro de pós-graduação, ambos com menos de dois anos de experiência no desenvolvimento de aplicativos para dispositivos móveis. Os outros eram profissionais com mais de dois anos de experiência em desenvolvimento de aplicativos para dispositivos móveis. Os entrevistados tinham entre 19 e 29 anos de idade, e eles foram todos do Brasil. As entrevistas foram realizadas no período de agosto a dezembro de 2013, através de uma amostra não probabilística, com contatos dos pesquisadores.

Foi utilizada a teoria de difusão da inovação [26] como um ponto de partida para a fase de pesquisa exploratória. Uma inovação, de acordo com Rogers, é qualquer ideia, prática ou projeto que seja percebido como novo pela unidade adotante. Assim, para um desenvolvedor que nunca desenvolveu uma aplicação móvel, Android e iOS podem ser considerada como inovação, o que significa que a teoria de difusão da inovação foi apropriada para a pesquisa exploratória.

Rogers explica a difusão da inovação através de quatro aspectos diferentes, ou seja, (i) a própria inovação, (ii) os canais de comunicação utilizados para transmitir informação sobre a

inovação, (iii) o sistema social em que a inovação e os adotantes estão incorporados, e (iv) a escala de tempo em que a inovação é adotada. Com base nesses pilares, criou-se um guia de entrevista. Todas as entrevistas foram gravadas e posteriormente transcritas. Foram utilizadas técnicas de teoria fundamentada de dados para efetuar a análise [10]. A codificação aberta e axial resultou na identificação de 265 diferentes categorias que foram combinados e são apresentados de maneira integrada na seção 4. Os detalhes desta codificação não serão apresentados aqui por limitações de espaço.

3.2 Exploratória - Global

Dado o número pequeno de entrevistas e a pequena diversidade de participantes selecionados na fase exploratória local, a segunda fase do estudo contou com a participação de desenvolvedores cadastrados no serviço GitHub. Os participantes foram selecionados usando um *parser* que extrai informações desse serviço através da análise da documentação de várias plataformas para desenvolvimento de aplicativos para dispositivos móveis e projetos no GitHub. Observou-se que projetos para o Android, Windows Phone e plataformas de FirefoxOS continham arquivos com nomes específicos, enquanto que nos projetos para iOS a palavra `LSRequiresiPhoneOS` ocorria em alguns arquivos. Com base nessas informações, foi escrito um *parser* que acessa o GitHub através de sua API² e identifica projetos com esses indicadores. Dessa forma, identificou-se projetos que incluíam o desenvolvimento de aplicativos para dispositivos móveis. Foram buscadas então informações sobre os desenvolvedores desses projetos. Foram selecionados aleatoriamente 400 informantes para os quais o questionário foi distribuído utilizando o email público disponível em seus perfis.

Foi elaborado um questionário com 15 questões, divididas em três seções. A primeira seção visava a obtenção de informações sobre o participante (experiência de trabalho no desenvolvimento de aplicativos para dispositivos móveis, a(s) plataforma(s) usada(s) etc). A segunda seção coletou informações sobre a aplicação mais importante desenvolvida pelo participante, se o aplicativo estava disponível na loja de aplicativos e quais ferramentas foram utilizadas para desenvolvê-lo. Finalmente, a terceira seção perguntava sobre as percepções dos desenvolvedores em relação ao desenvolvimento de aplicativos móveis, incluindo quais foram os aspectos mais positivos e negativos que tiveram de enfrentar durante o desenvolvimento, uma comparação entre desenvolvimento móvel e Web/desktop, e o problema mais difícil enfrentado. Todas as questões desta última seção eram abertas.

Foram recebidas 83 respostas para a pesquisa (taxa de resposta de 20,75%, adequada para este tipo de pesquisa). Os doze locais mais frequentes dos entrevistados são: 1° - EUA (24); 2- Canadá (4); 3 - Brasil, Índia, Holanda e Reino Unido (3); 4 - Bélgica, Camboja, China, França, Espanha e Taiwan (2). Outros países indicados pelos desenvolvedores incluíram: Bolívia, Dinamarca, Finlândia, Geórgia, Indonésia, Israel, Japão, México, Suécia e Suíça. As informações de localização não são exigidas pelo GitHub, o que explica a quantidade de informação faltando sobre a localização de alguns dos entrevistados (17).

As três plataformas mais citadas foram Android (50 desenvolvedores), Web (40) e iOS (39), e é importante observar que para esta questão os informantes poderiam selecionar mais de

² <https://developer.github.com/v3/search/>

uma plataforma. A experiência dos desenvolvedores variou de menos de um ano a mais de cinco anos. A quantidade de aplicativos publicados em lojas de aplicativos variaram entre um e mais de seis aplicativos em lojas de aplicativos móveis.

Da mesma forma que o conjunto de dados anterior, foram analisadas as respostas das questões abertas do questionário, utilizando técnicas da teoria fundamentada em dados [10]. Além disso, integrou-se os dados de ambos os conjuntos de dados (entrevistas e questionário) adotando um conjunto único de códigos. Com base na análise destes códigos, foi possível observar que há três aspectos que influenciam o desenvolvimento de aplicativos para dispositivos móveis e software: aspectos técnicos, de negócio e sociais.

Chamou a atenção como os aspectos sociais influenciaram os desenvolvedores para escolher e permanecer desenvolvendo para uma plataforma móvel em particular. Por exemplo, nas entrevistas iniciais alguns desenvolvedores mencionaram que eles escolheram Android porque eles tinham amigos desenvolvendo aplicativos para Android, portanto, escolher Android (e ter as pessoas a fazer perguntas, obter *feedback*, etc) foi uma escolha natural para eles. Decidiu-se focar o aspecto social, uma vez que ele é ignorado na maior parte da literatura. Assim, iniciou-se a segunda parte do estudo: a fase de confirmação.

3.3 Confirmatória - Local

O primeiro passo nessa fase foi a construção de um novo guia de entrevista que incidiu principalmente sobre os aspectos sociais do desenvolvimento de software para dispositivos móveis. Perguntas incluindo os benefícios percebidos de ser um membro de uma comunidade online relacionada a uma plataforma móvel, como é importante fazer parte dessa comunidade, como essa comunidade pode ajudar os iniciantes para começar a desenvolver para a plataforma, e se há interação com outros membros da comunidade, por exemplo, *blogs*, respondendo a perguntas em *sites*, etc.

Usando este guia de entrevistas atualizado, foram realizadas nove entrevistas adicionais semiestruturadas com novos desenvolvedores da plataforma Android, todos sendo brasileiros (8 homens e 1 mulher), através de novos contatos dos autores. Os entrevistados tinham entre 20 e 28 anos com experiência de desenvolvimento que variaram de sete meses a seis anos. As entrevistas foram novamente realizadas com base em uma amostra por conveniência. As entrevistas foram gravadas e tiveram duração entre 13 a 28 minutos.

3.4 Confirmação - Global

Uma das perguntas do questionário de pesquisa exploratória perguntava se o informante poderia ser contatado para uma entrevista. Em caso de uma resposta positiva, foi pedido o endereço de e-mail do informante. Foram recebidas 25 respostas positivas dos 83 entrevistados. Enviou-se um e-mail convite para estes 25 informantes perguntando se ainda estariam dispostos a serem entrevistados. Com base em suas respostas, foi possível entrevistar sete informantes, utilizando o mesmo roteiro de entrevista que foi utilizado no conjunto anterior de entrevistas. Os entrevistados eram de diferentes países, incluindo Estados Unidos, Brasil e Venezuela e tinham diferentes níveis de experiência em desenvolvimento de aplicativos para dispositivos móveis, variando de um a quatro anos. Os entrevistados desenvolviam para Android (4), iOS (1), ambos (1) e para

Android, iOS e Windows Phone (1). As entrevistas foram gravadas e tiveram duração de 14 a 45 minutos.

3.5 Análise dos Dados

Na análise de dados, foram transcritas todas as entrevistas e integrados todos os quatro conjuntos de dados em um único conjunto de dados. Depois disso, os dados coletados foram analisados em conjunto utilizando métodos da teoria fundamentada em dados [10].

No primeiro passo de codificação - aberta - os dados foram analisados detalhadamente para identificar categorias. Foram identificadas duas principais categorias relacionadas com os aspectos sociais do desenvolvimento móvel: adoção e permanência, por exemplo, os aspectos sociais que influenciam um desenvolvedor de software para adotar uma plataforma para desenvolvimento de aplicativos para dispositivos móveis em particular, e os aspectos sociais que permitem que um desenvolvedor de software a continuar usando a plataforma escolhida.

No próximo passo - codificação axial - as categorias foram divididas em subcategorias e identificou-se propriedades e dimensões dessas categorias. Por exemplo, uma categoria identificada nos dados é chamada Comunidade; portanto, criou-se propriedades sobre o tamanho (dúzia vs. milhares de pessoas), o escopo desta comunidade (local vs. global), e os canais de comunicação utilizados (redes sociais, listas de discussão, vídeos, *blogs*, etc).

Inicialmente, diferentes autores codificaram conjuntos de dados diferentes, todos focando nos aspectos sociais do desenvolvimento de ecossistemas de software. Como seria de esperar isso resultou em ideias semelhantes que foram codificadas de forma diferente. Por isso, foi realizado um *workshop* de 3 dias em que o conjunto de dados integrado foi analisado novamente e codificado usando um esquema de codificação única. Esse esquema de codificação foi organizado em dois aspectos diferentes: (i) os aspectos sociais da *adoção* e (ii) da *permanência* em um ecossistema. Além disso, o esquema de codificação criou 25 categorias diferentes, incluindo “adoção influenciada pela rede social”, “a relação entre um desenvolvedor e a empresa *keystone*” etc., bem como subcategorias indicadas antes, o aspecto da comunidade e seu tamanho, escopo e canais de comunicação.

4. RESULTADOS

Nesta seção, são relatados os diferentes aspectos que influenciam a adoção dos ecossistemas e a permanência dos desenvolvedores nos mesmos.

4.1 Influências na Adoção de um Ecossistema

Neste estudo, foi perguntado aos desenvolvedores como eles foram introduzidos para o desenvolvimento móvel e se alguém influenciou suas decisões ao adotar uma plataforma particular. Descobriu-se que os desenvolvedores são frequentemente influenciados por outros desenvolvedores próximos socialmente, incluindo amigos, colegas de trabalho e até mesmo a família. A seguir, é apresentada uma citação de um desenvolvedor que foi influenciado por um amigo:

“Na verdade, a ideia veio do nada, mas teve a influência de um amigo nosso, que também trabalhou com o

desenvolvimento para as pessoas com deficiência. E então decidimos: vamos fazer um aplicativo para isso.” [E2]³

Outro exemplo de como a comunidade social em que se está incorporado influencia sua decisão é ilustrado pelo informante E7. Este informante mencionou que ele(a) não escolheu uma plataforma móvel particular, porque nenhuma de seus amigos tinha um smartphone com essa plataforma, portanto, ele(a) viu outras plataformas como mais abertas e acessíveis. Este é um exemplo onde os aspectos do negócio, o custo de uma plataforma e, portanto, sua baixa disponibilidade, estão relacionados aos aspectos sociais, a baixa disponibilidade de smartphones com um determinado plataformas móveis entre os amigos.

A Teoria de Difusão da Inovação de Rogers define compatibilidade como o grau em que uma inovação é percebida como consistente com os valores existentes, experiências passadas, e as necessidades dos potenciais adotantes [26]. Neste estudo, os participantes que aprenderam Java anteriormente tinha uma maior compatibilidade com Android, o que tornou mais fácil para eles adotarem a plataforma, conforme ilustrado abaixo:

“iOS foi mais complicado, devido à sua linguagem [de programação], eu fiz algumas coisas mas realmente não entendia como funcionava. Android era mais simples porque é Java, que eu estudei no meu curso de graduação.” [E4]

Em outras palavras, a citação anterior ilustra como o contexto geral no qual um desenvolvedor está incorporado influencia sua decisão. Neste caso, isso está relacionado com a universidade ele(a) frequentou, o conhecimento anterior que possuía.

As preferências dos desenvolvedores frente aos ecossistemas também são influenciados pela sua identificação pessoal com decisões de governança feitas por empresas *keystone*. Por exemplo, alguns participantes ficaram entusiasmados com o desenvolvimento de código aberto e, portanto, rejeitaram o iOS por considerá-lo uma plataforma fechada. Eles apreciam a capacidade de alterar o código-fonte do Android e adaptá-lo às suas necessidades. Koch e Kerschbaum [20] apresentaram uma conclusão semelhante, apontando a opção para alterar código fonte do sistema operacional da plataforma como um importante critério de decisão para os desenvolvedores para se juntar a um ecossistema.

“Eu participo de algumas comunidades de código aberto. Isso foi algo que me influenciou a escolher Android porque é mais aberto do que a plataforma iOS.” [C3]

A lógica por trás das decisões dos participantes sobre a adoção de plataformas móveis também foi baseada em suas ambições de chegar a um maior número de usuários. O sistema operacional Android estava presente em 89% de todos os smartphones no Brasil até o final do segundo trimestre de 2014, de acordo com um estudo da Kantar Worldpanel ComTech⁴.

“Eu diria que depende da finalidade, se se quer chegar a um maior número de usuários, tem de se desenvolver para o Android, porque há poucas pessoas no Brasil que possuem um dispositivo iOS.” [E5]

³ É utilizado a letra C para designar os participantes da fase de confirmação e E para designar participantes da fase exploratória.

⁴ <http://www.kantarworldpanel.com/global/smartphone-os-market-share/>

Em geral, os resultados sugerem que aspectos sociais, entre outros aspectos técnicos e não técnicos, são fatores importantes que contribuem para atrair desenvolvedores para participar de um ecossistema de software em particular. Desenvolvedores são influenciados por outros desenvolvedores, mas também por suas próprias experiências anteriores, as suas expectativas e percepções do mercado local, e sua identificação pessoal com decisões de governança feitas por empresas *keystone*.

O aumento da adoção de um ecossistema de software por desenvolvedores é um passo importante no processo de construção de um ecossistema de software bem sucedido. No entanto, os aspectos sociais que influenciam a permanência dos desenvolvedores dentro de um ecossistema de software após a adoção inicial são importantes, uma vez que influenciam a sua sustentabilidade.

4.2 Influências Sociais Na Permanência dos Desenvolvedores

Como mencionado anteriormente, a permanência em ecossistema de software foi definida como "a continuação da participação de desenvolvedores em um ecossistema de software". Especialmente na segunda fase do estudo, concentrando nos aspectos sociais que influenciam os desenvolvedores a permanecerem envolvidos com um ecossistema.

De acordo com Rogers [26], adotantes, na fase de confirmação tomara a sua decisão de continuar usando uma inovação. Neste estudo, no entanto, verificou-se que este é apenas o começo de uma série de interações com outros atores sociais que apoiam uns aos outros e tornam possível para os desenvolvedores continuar participando de um ecossistema. É descrito cada um desses atores sociais: desenvolvedores internos, usuários finais e comunidades de desenvolvedores.

4.2.1 O Papel dos Desenvolvedores Internos

Desenvolvedores internos são muitas vezes empregados de grandes empresas *keystone*, sendo envolvidos com o desenvolvimento de SDKs, APIs, bibliotecas, emuladores e outras ferramentas para desenvolvedores externos [6].

Neste estudo, descobriu-se que desenvolvedores externos valorizam a participação e engajamento dos colaboradores internos em *sites* de redes sociais e meios de comunicação social. Esta participação fornece níveis mais elevados de transparência social [31] sobre a identidade de desenvolvedores internos e se os desenvolvedores externos podem interagir com eles.

“Eu não sei quem são os caras que desenvolvem o iOS, eu não sei como eles se comportam em redes sociais, enquanto os desenvolvedores do Android deixam claro como você pode entrar em contato com eles.” [C7]

Desenvolvedores do Android no Google utilizam canais de comunicação específicos para divulgar informações sobre a documentação e lançamentos oficiais [32], mas é aproveitando o seu nível de transparência social na mídia social que conseguem uma melhor eficácia na promoção da comunidade Android para o desenvolvedor externo.

“A comunidade Android se encontra no Google Groups ou Google+. [...] Há mesmo a participação de membros da equipe do Google, o que nos motiva muito.” [C3]

Ao exibir as suas identidades e atividades em redes sociais, desenvolvedores internos estão mostrando seu interesse e

compromisso com a comunidade externa de desenvolvedores. Esta atitude em relação à participação da comunidade aumenta a confiança na plataforma e motiva os desenvolvedores externos para continuar participando do ecossistema.

4.2.2 O Papel dos Usuários Finais

Os usuários finais fornecem *feedback* para os desenvolvedores e definem as expectativas sobre novos recursos e correções de *bugs* a serem implementados. Lojas de aplicativos não são apenas um canal para a distribuição de aplicações móveis, mas eles também servem como canal de comunicação entre os usuários finais e desenvolvedores.

“Eu posso ver o número de usuários que estão usando o meu app, quantas pessoas estão desinstalando, quantas pessoas já fizeram o download do mesmo e de qual países [elas são]. Se um usuário classifica meu aplicativo e relata um erro, eu posso responder a ele. Isso permite a interação direta com o usuário.” [E9]

Os desenvolvedores são capazes de testar seus aplicativos através de distribuição direta aos usuários antes de liberar a versão final que vai para a loja de aplicativos. Isso melhora o processo de distribuição, tornando mais fácil para os desenvolvedores obter *feedback* dos usuários finais e clientes.

“Testamos o aplicativo, dando uma amostra do aplicativo para alguns amigos e familiares, então vamos esperar pelo feedback. Nós distribuimos usando e-mail e pen drives.” [E1]

Promover a relação entre desenvolvedores e usuários finais é um aspecto essencial dos ecossistemas de software, pois isso permite que os desenvolvedores melhorem a qualidade de seus aplicativos com base no *feedback* de seus usuários.

Neste estudo, revelou-se que os desenvolvedores usam as lojas oficiais de aplicativos como seu principal canal de comunicação para se comunicar com usuários finais e clientes, mas também dependem de relações sociais para distribuir suas aplicações. Esta flexibilidade no processo de distribuição foi mencionada várias vezes nas entrevistas como um fator que influencia a continuação da participação dos desenvolvedores nos ecossistemas.

4.2.3 O Papel das Comunidades de Desenvolvedores

Desenvolvedores se beneficiam com a ascensão das mídias sociais e o uso ubíquo de *sites* de relacionamento para troca de informações sobre desenvolvimento de software [29, 30].

Este estudo revelou que grupos *on-line* no Google+ e Facebook são espaços importantes para a obtenção de respostas rápidas de desenvolvedores mais experientes e para aprender novas técnicas de desenvolvimento.

“Eu acho que é realmente importante se juntar a um grupo, tirar suas dúvidas e ter pessoas que você pode contar. O tempo de resposta é rápido, às vezes você posta e receber uma resposta em uma hora [...] Eu acho que é mais rápido do que se você fizer perguntas em um fórum.” [E7]

Desenvolvedores apoiam uns aos outros, respondendo a perguntas e postando no Stack Overflow, um popular *site* de perguntas e respostas, um local para os desenvolvedores. Mais de 92% das questões publicadas no Stack Overflow sobre temas especializados são respondidas - em um tempo médio de 11 minutos [21].

“Na semana passada tive um problema [...] que eu postei no Stack Overflow, o cara me ajudou e resolveu o problema. Tud o está funcionando agora, a comunidade é muito útil.” [E8]

A variedade de informações sobre o desenvolvimento de software aumentou drasticamente nos últimos anos. A evolução rápida de técnicas e tecnologias de desenvolvimento fez materiais didáticos, como livros, tornarem-se obsoletos em um ritmo muito mais rápido. Através da participação em comunidades de desenvolvedores e mídias sociais, os desenvolvedores são capazes de manter-se em um ambiente em evolução e eles dependem um do outro para se manter atualizado perante as novidades.

“É difícil encontrar soluções para os problemas cotidianos. Em geral, a comunidade é a principal fonte. No momento em que você compra um livro, é provável que ele já esteja desatualizado.” [E1]

Comunidades de desenvolvedores complementam o papel das empresas *keystone* e seus desenvolvedores internos, apoiando desenvolvedores externos com problemas de programação e perguntas, proporcionando-lhes informações sobre novas técnicas de desenvolvimento. O engajamento de desenvolvedores nessas comunidades é essencial para a sua participação continuada e permanência nos ecossistemas.

5. DISCUSSÃO

Os desenvolvedores de software constantemente tem que tomar decisões sobre quais bibliotecas ou ferramentas usar e para quais plataformas desenvolver. Enquanto algumas decisões possuem um “pequeno impacto”, como a escolha de uma API para um problema específico, outras decisões são de “grande impacto” como tomar a decisão de se tornar um contribuinte ativo para um projeto *open source* ou adotar um ecossistema.

Um exemplo de “pequeno impacto” seria a escolha de uma API, onde aspectos sociais *não* desempenham um papel tão importante. Em pesquisa sobre os obstáculos encontrados pelos desenvolvedores tentando aprender uma API, Robillard [25] identifica os problemas relacionados com recursos inadequados ou ausentes, a estrutura ou o *design* da API, o ambiente técnico em que a API é usada, bem como as questões de processo. Além disso, ele observou os obstáculos causados pelo background dos entrevistados e experiências anteriores - uma categoria que também surgiu neste estudo. Note que os aspectos sociais não são completamente ignorados: por exemplo, Chen et al. [9] identificaram onze temas que desempenham papel importante quando os desenvolvedores escolhem uma API, tais como capacidade de aprendizado e interoperabilidade. Entre outros, eles identificaram uma comunidade ativa como um fator importante na escolha de APIs. Uma comunidade ativa foi particularmente importante quando a documentação oficial é falha pois a mesma poderia responder perguntas específicas e fornecer informações *up-to-date* de uma API - um resultado que é espelhado pelos achados deste estudo.

No outro extremo do espectro estão decisões de “grande impacto” como por exemplo se tornar um contribuinte ativo de um projeto *open source* ou adotar um ecossistema de software. Ao tomar estas decisões, os desenvolvedores consideram uma combinação de fatores sociais, técnicos e de negócios, e os resultados desta pesquisa sugerem que o lado social é muito importante e influencia nesta decisão. Na verdade, Hertel et al. [15]

compararam ingressar em um projeto *open source* a participar de um movimento social, como o movimento dos direitos civis. Em seu estudo sobre a motivação de desenvolvedores de software em projetos *open source*, eles descobriram que o envolvimento dos desenvolvedores foi particularmente determinado pela sua identificação como um desenvolvedor para um projeto particular, bem como por motivos pragmáticos para melhorar o seu próprio trabalho e aumentar as chances de sucesso em sua carreira. Este estudo revelou resultados semelhantes: as preferências dos desenvolvedores frente aos ecossistemas são influenciadas por sua identificação pessoal com as estratégias de governança das empresas *keystone*.

Semelhante aos resultados deste estudo, a investigação sobre as motivações dos desenvolvedores para participar de projetos de código aberto tem sido dividida entre a adoção inicial e a fase de permanência. Shah [28] observou que, para alguns desenvolvedores, motivação evolui ao longo do tempo e participação torna-se um *hobby*. Desenvolvedores que veem a sua participação na comunidade *open source* como um *hobby* são fundamentais para a sustentabilidade em longo prazo do software à medida que assumem tarefas que poderiam não ser realizadas por outros desenvolvedores. A seção 2, revisão da literatura, relata que ecossistemas podem ser baseados em plataforma (como iOS, Android, etc) ou baseados em projetos (como Eclipse, Apache, etc). Embora os resultados de Shah [28] sejam relacionados aos ecossistemas baseados em projetos, os resultados do estudo aqui apresentado indicam resultados similares para os ecossistemas baseados em plataformas.

O trabalho de Koch e Kerschbaum [20] é um dos poucos que se concentra nas razões que levam os desenvolvedores de software a adotarem um ecossistema de software específico. Estes autores relatam que a maior motivação para os desenvolvedores é a experiência de diversão e estímulo intelectual e o processo de aprendizagem de novas habilidades. O ganho financeiro de participar de um ecossistema de software é fator o que menos motiva estes desenvolvedores. Além disso, para escolher entre dois ecossistemas, a maioria dos critérios importantes são o grau de abertura⁵ [24], barreiras / critérios de inscrição, tamanho da rede (número de usuários no mercado de *apps* e *market share* da plataforma) e a qualidade das ferramentas de desenvolvimento. Em outras palavras, os aspectos sociais dos ecossistemas de software são apenas implicitamente mencionados no trabalho de Koch e Kerschbaum [20]. Por outro lado, os resultados apresentados neste estudo sugerem que os aspectos sociais, como ter um amigo que desenvolve para uma determinada plataforma, são importantes pois reduzem as barreiras à entrada no ecossistema. De fato, diversos aspectos sociais dos ecossistemas de software são relatados neste trabalho.

6. CONCLUSÃO

Ecossistemas de software estão mudando a maneira como o software é desenvolvido. O modelo simples em que uma única empresa desenvolve um produto e seus clientes compram este produto foi substituído por um modelo complexo composto por várias empresas, desenvolvedores internos e externos, uma comunidade de desenvolvedores e usuários finais. Enquanto os

⁵ O grau de abertura da plataforma é definido como o grau em que uma plataforma coloca menos restrições à participação, desenvolvimento, ou uso em seus papéis distintos, seja ao desenvolvedor ou ao usuário final [14].

aspectos técnicos e de negócio de tais ecossistemas de software têm sido investigados em pesquisas anteriores, o lado social dos ecossistemas de software não é tão bem compreendido.

Para preencher esta lacuna, foi realizado um estudo empírico qualitativo com base em 25 entrevistas com desenvolvedores de software móvel e uma pesquisa online com 83 respondentes da comunidade de desenvolvimento móvel. A análise de dados usando técnicas de teoria fundamentada em dados revelou um complexo sistema social baseado na interação contínua e apoio mútuo entre os diferentes atores. Este sistema social não só desempenha um papel importante quando os desenvolvedores de software inicialmente optaram por *adotar* um ecossistema de software em particular, mas também é crucial para *sustentar* os laços de um desenvolvedor com esse ecossistema.

Como em qualquer estudo empírico, este trabalho possui diversas limitações. Por exemplo, a maior parte dos entrevistados desenvolviam aplicativos para as plataformas Android ou iOS. Assim, os resultados encontrados neste trabalho podem não ser aplicáveis às plataformas Windows Phone, Blackberry, ou outras. De maneira similar, os resultados não foram classificados de acordo com as experiências dos entrevistados, idades ou regiões dos mesmos. Sendo que estes fatores podem influenciar no resultados obtidos. Por exemplo, o entrevistado E5 relata que para se atingir um maior número de usuários finais, o melhor seria desenvolver aplicações para a plataforma Android (ver Seção 4.1), pois ele considera o contexto (país) no qual ele está inserido. Desta forma, dependendo do contexto do informante este resultado ou opinião pode ser diferente.

Em trabalhos futuros, é interessante continuar a investigar os aspectos sociais dos ecossistemas de software a partir da perspectiva dos diferentes atores envolvidos. Em particular, pretende-se estudar as consequências das diferentes estratégias das empresas *keystone* em um ecossistema com mais detalhes. Pretende-se também avaliar com os entrevistados aspectos negativos em relação aos ecossistemas, ou seja, aspectos que afastaram o desenvolvedor de um determinado ecossistema.

7. AGRADECIMENTOS

Os autores gostariam de agradecer ao apoio financeiro recebido através da Chamada 59/2013 MCTI/CT-Info/CNPq, processo 440880/2013-0 e através do processo 310468/2014-0.

8. REFERÊNCIAS

- [1] Alves, A. M. and M. Pessôa (2010): 'Brazilian Public Software: Beyond Sharing'. In: Proc. of the Intl. Conf. on Management of Emergent Digital EcoSystems. pp. 73–80.
- [2] Barbosa, O., R. Santos, C. Alves, C. Werner, and S. Jansen (2013): Software Ecosystems: Analyzing and Managing Business Networks in Software Industry, Chapt. A Systematic Mapping Study on Software Ecosystems through a Three-dimensional Perspective. Edward Elgar.
- [3] Biehl, J. T., M. Czerwinski, G. Smith, and G. G. Robertson (2007): 'FASTDash: A Visual Dashboard for Fostering Awareness in Software Teams'. In: Proc. of the Conf. on Human Factors in Computing Systems. pp. 1313–1322.
- [4] Bjorn, P. and L. Christensen (2011): 'Relation work: Creating socio-technical connections in global engineering'. In: S. Bdker, N. O. Bouvin, V. Wulf, L. Cioffi, and W. Lutters (eds.): ECSCW 2011: Proceedings of the 12th

- European Conference on Computer Supported Cooperative Work, 24-28 September 2011, Aarhus Denmark. Springer London, pp. 133–152.
- [5] Boden, A., F. Rosswog, G. Stevens, and V. Wulf (2014): ‘Articulation Spaces: Bridging the Gap Between Formal and Informal Coordination’. In: Proc. of the 17th Conf. on Computer Supported Cooperative Work & Social Computing. pp. 1120–1130.
- [6] Bosch, J. (2009): ‘From Software Product Lines to Software Ecosystems’. In: Proc. of the 13th Intl. Software Product Line Conf. pp. 111–119
- [7] Cataldo, M., P. A. Wagstrom, J. D. Herbsleb, and K. M. Carley (2006): ‘Identification of coordination Requirements: Implications for the Design of Collaboration and Awareness Tools’. In: Proc. of the 20th Conf. on Computer Supported Cooperative Work. pp. 353–362
- [8] Charmaz, K. (2006): *Constructing Grounded Theory: A Practical Guide through Qualitative Analysis*. SAGE Publications
- [9] Chen, X., M. Dunlap, R. Fung, and T. Souza, ‘How Do Developers Choose APIs?’. University of Calgary, Canada
- [10] Corbin, J. and A. Strauss (2008): *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage Publications, 3rd edition
- [11] da Silva Amorim, S., J. D. McGregor, E. S. de Almeida, and C. von Flach G. Chavez (2014): ‘Flexibility in Ecosystem Architectures’. In: Proc. of the European Conf. on Software Architecture Workshops. pp. 14:1–14:6.
- [12] de Souza, C. and D. Redmiles (2009): ‘On The Roles of APIs in the Coordination of Collaborative Software Development’. *Computer Supported Cooperative Work (CSCW)*, vol. 18, no. 5-6, pp.445–475
- [13] Dhungana, D., I. Groher, E. Schludermann, and S. Biffi (2010): ‘Software Ecosystems vs. Natural Ecosystems: Learning from the Ingenious Mind of Nature’. In: Proc. of the 4th European Conf. on Software Architecture: Companion Volume. pp. 96–102.
- [14] Eisenmann, T. R., G. Parker, and M. W. Van Alstyne (2009): ‘Opening Platforms: How, When and Why?’. In: A. Gawer (ed.): *Platforms, Markets and Innovation*. Edward Elgar Publishing, pp. 131–162.
- [15] Hertel, G., S. Niedner, and S. Herrmann (2003): ‘Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel’. *Research Policy*, vol. 32, pp. 1159–1177
- [16] Iansiti, M. and R. Levien (2004): *The keystone advantage: what the new dynamics of business ecosystems mean for strategy, innovation, and sustainability*. Harvard Business Press
- [17] Jansen, S., A. Finkelstein, and S. Brinkkemper (2009): ‘A sense of community: A research agenda for software ecosystems’. In: Proc. of the 31st Intl. Conf. on Software Engineering - Companion Volume. pp. 187–190.
- [18] Jergensen, C., A. Sarma, and P. Wagstrom (2011): ‘The Onion Patch: Migration in Open Source Ecosystems’. In: Proc. of the 19th SIGSOFT Symp. and the 13th European Conf. on Foundations of Software Engineering. pp. 70–80.
- [19] Karhu, K., T. Tang, and M. Hämäläinen (2014): ‘Analyzing Competitive and Collaborative Differences Among Mobile Ecosystems Using Abstracted Strategy Networks’. *Telemat. Inf.*, vol. 31, no. 2, pp. 319–333.
- [20] Koch, S. and M. Kerschbaum (2014): ‘Joining a smartphone ecosystem: Application developers motivations and decision criteria’. *Information and Software Technology*, vol. 56, no. 11, pp. 1423–1435.
- [21] Mamykina, L., B. Manoim, M. Mittal, G. Hripscak, and B. Hartmann (2011): ‘Design Lessons from the Fastest Q&A Site in the West’. In: Proc. of the Conf. on Human Factors in Computing Systems. pp. 2857–2866.
- [22] Manikas, K. and K. M. Hansen (2013): ‘Software ecosystems A systematic literature review’. *Journal of Systems and Software*, vol. 86, no. 5, pp. 1294–1306.
- [23] Miranda, G., de Souza, J., Braganholo, V. and de Oliveira D. (2014): ‘CollabCumulus: Uma Ferramenta de Apoio à Análise Colaborativa de Proveniência em Workflows Científicos’. in Proc. of the XI Brazilian Symposium in Collaborative Systems, SBSC ’14, (Curitiba, PR, Brazil).
- [24] Parker, G. and M. Van Alstyne (2010): ‘Innovation, Openness & Platform Control’. In: Proc. of the 11th Conf. on Electronic Commerce. pp. 95–96.
- [25] Robillard, M. P. (2009): ‘What Makes APIs Hard to Learn? Answers from Developers’. *IEEE Software.*, vol. 26, no. 6, pp. 27–34.
- [26] Rogers, E. M. (2003): *Diffusion of innovations*. Free Press, 5th edition.
- [27] Ron, A. (2012): *The Wide Lens: A New Strategy for Innovation*. Portfolio Hardcover.
- [28] Shah, S. K. (2006): ‘Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development’. *Manage. Sci.*, vol. 52, no. 7, pp. 1000–1014.
- [29] Storey, M.-A., L. Singer, B. Cleary, F. Figueira Filho, and A. Zagalsky (2014): ‘The (R) Evolution of Social Media in Software Engineering’. In: Proc. of Future of Software Engineering. pp. 100–116.
- [30] Storey, M.-A., C. Treude, A. van Deursen, and L.-T. Cheng (2010): ‘The Impact of Social Media on Software Engineering Practices and Tools’. In: Proc. of the Workshop on Future of Software Engineering Research. pp. 359–364.
- [31] Stuart, H. C., L. Dabbish, S. Kiesler, P. Kinnaird, and R. Kang (2012): ‘Social Transparency in Networked Information Exchange: A Theoretical Framework’. In: Proc. of the CSCW. pp. 451–460.
- [32] Treude, C. and M.-A. Storey (2011): ‘Effective Communication of Software Development Knowledge Through Community Portals’. In: Proc. of the 19th SIGSOFT Symp. and the 13th European Conf. on Foundations of Software Engineering. pp. 91–101.
- [33] Yu, E. S. K. and S. Deng (2011): ‘Understanding Software Ecosystems: A Strategic Modeling Approach’. In: Proc. of the 3rd Intl. Workshop on Software Ecosystems. pp. 65–76.