

# Um Estudo Exploratório Sobre Contribuições Casuais em Projetos de Software Livre: Caso do Projeto LibreOffice

Felipe V. Ramos<sup>1</sup>, Marco Aurélio Gerosa<sup>2</sup>, Ana Paula Chaves<sup>1</sup>, Igor Steinmacher<sup>1</sup>

<sup>1</sup>Departamento de Computação (DACOM)  
Universidade Tecnológica Federal do Paraná (UTFPR) – Campo Mourão-PR

<sup>2</sup>Instituto de Matemática e Estatística (IME)  
Universidade de São Paulo (USP) – São Paulo-SP

fveigaramos@gmail.com, gerosa@ime.usp.br, {anachaves, igorfs}@utfpr.edu.br

**Abstract.** *Many free software communities leverage contributions from volunteers and need to attract, motivate and engage new developers. A phenomenon that has attracted attention are the casual contributions made by developers who are not interested in becoming members of the projects. The goal of this paper is to conduct an exploratory study of the interaction of newcomers to LibreOffice project to understand the phenomenon of casual contributions. To achieve this, we analyzed historical data from mailing lists, issue tracker and code review management tool. The results showed that 27.1% of the 2013 newcomers were classified as casual contributors and the effort by the community to support them is relatively low.*

**Resumo.** *Diversas comunidades de software livre contam com contribuições de voluntários e precisam atrair, motivar e engajar novos desenvolvedores. Um fenômeno que tem atraído atenção são as contribuições casuais, realizadas por desenvolvedores que não tem interesse em se tornar membros dos projetos. O objetivo deste artigo é conduzir um estudo exploratório da interação dos contribuintes novatos do LibreOffice para compreender o fenômeno das contribuições casuais. Para isso, foram analisados dados históricos da lista de emails, gerenciador de tarefas e gerenciador de revisões de código. Os resultados mostram que 27,1% dos novatos de 2013 foram classificados como contribuintes casuais e o esforço despendido pela comunidade para auxiliá-los é relativamente baixo.*

## 1. Introdução

Uma das aplicações mais bem sucedidas da abordagem de desenvolvimento distribuído de software é o modelo utilizado pelas comunidades de software livre [Carmel and Tjia 2005, German 2003]. Muitos desses projetos são comunidades de contribuição aberta, dependendo de contribuições de voluntários para sua manutenção, evolução e sustentabilidade. Sendo assim é essencial motivar, engajar e reter novos desenvolvedores.

Existem diversos estudos na literatura focando em diferentes aspectos da entrada de novatos, incluindo como se tornar um membro, motivação para entrada e retenção [Steinmacher et al. 2015b]. Entretanto, as pesquisas anteriores se preocupam em entender a dinâmica que leva os novatos a tornarem-se membros, incluindo desenvolvedores do núcleo e desenvolvedores de longo prazo [Fagerholm et al. 2014, Steinmacher et al. 2015a]. Um tópico negligenciado pela literatura diz respeito aos desenvolvedores que não tem interesse em se tornar membros dos projetos, mas contribuir apenas uma vez ou resolver um problema pontual. Esse tipo de comportamento é chamado de “*drive-by commit*” [Pham et al. 2013a], e aqui será chamado de contribuições casuais.

O fenômeno das “contribuições casuais” vem ganhando atenção recentemente. No GitHub<sup>1</sup>, por exemplo, essas contribuições tem se tornado mais comuns [Pham et al. 2013a]. De acordo com Gousios et al. [Gousios et al. 2014], este tipo de contribuição foi responsável por 7% dos *pull requests* no GitHub em 2012. Essas contribuições casuais possibilitam maior colaboração e mais diversidade no projeto [Pham et al. 2013b] e não dependem de engajamento específico com o projeto e, muitas vezes, são feitas por colaboradores que não voltarão.

De acordo com a literatura corrente, mais pesquisas relacionadas ao fenômeno das contribuições casuais precisam ser realizadas para um melhor entendimento dos processos, dos benefícios [Pham et al. 2013a] e das implicações [Gousios et al. 2014, Vasilescu et al. 2015] destas contribuições. Visto isso, o objetivo do presente artigo é conduzir um estudo exploratório preliminar da interação dos contribuintes novatos do LibreOffice para compreender o fenômeno das contribuições casuais nesse projeto. Para alcançar tal objetivo, foram definidas as seguintes questões de pesquisa:

- Q1. É possível encontrar indícios da existência de contribuintes casuais?
- Q2. Qual o esforço que tais contribuições demandam da comunidade?

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados; a Seção 3 descreve a forma como os dados foram coletados e analisados; as Seções 4 e 5 discutem os resultados encontrados a partir da análise e as limitações desta pesquisa, respectivamente; as conclusões e possíveis desdobramentos são apresentados na Seção 6.

## 2. Trabalhos Relacionados

Vários trabalhos na literatura analisam aspectos específicos da entrada de novatos em projetos de software livre [Park and Jensen 2009, von Krogh et al. 2003, Zhou and Mockus 2015, Ducheneaut 2005, Bird 2011, Jensen et al. 2011, Stol et al. 2010, Steinmacher et al. 2015b]. Park e Jensen [Park and Jensen 2009] estudaram as necessidades dos novatos por informação. Os autores mostram que ferramentas de visualização de informações apoiam os primeiros passos de novatos na aprendizagem sobre um projeto de software livre, ajudando-os a encontrar informações mais rapidamente. Os autores perceberam que os novatos apoiados por ferramentas de visualização concluem suas atividades mais rapidamente e compreendem melhor o código do projeto. Por sua vez, Von Krogh et al. [von Krogh et al. 2003] conduziram um estudo sobre o projeto FreeNet, por meio de entrevista com desenvolvedores, análise de histórico de emails, repositório de código fonte e documentos do projeto. Os autores propuseram um roteiro de entrada para desenvolvedores. Uma de suas contribuições indica que os novatos frequentemente ficam observando o projeto antes de iniciar sua participação, para, em seguida, interagir. Apesar de estudar o processo de entrada em projetos de software livre, não há uma preocupação com a análise das razões de desistência, verificando apenas o comportamento daqueles que se tornaram membros do projeto.

Alguns trabalhos [Zhou and Mockus 2015, Ducheneaut 2005, Bird 2011] também analisam como a socialização influencia na retenção dos novatos em projetos de software

---

<sup>1</sup><http://www.github.com>

livre. Esses trabalhos analisam as redes sociais de comunicação (listas de email) para verificar com quem os novatos se relacionam, como a rede de contatos evolui de acordo com o tempo e como ocorre o relacionamento com os outros membros do projeto. Esses estudos, entretanto, mais uma vez analisam apenas o comportamento daqueles que permanecem no projeto e se tornam desenvolvedores do núcleo, sem analisar os problemas dos novatos que desistem ou que permanecem na periferia.

Jensen et al. [Jensen et al. 2011] fazem uma análise de quatro listas de emails de projetos de software livre, visando verificar se os emails de novatos são respondidos rapidamente, se o sexo e a nacionalidade dos novatos interferem no tipo de resposta recebida nos emails e na continuidade dos novatos, e, por fim, se o tratamento a novatos é diferente na lista de usuários e na lista de desenvolvedores. Por sua vez, Stol et al. [Stol et al. 2010] conduziram um estudo relacionado à identificação de padrões arquiteturais em projetos de software livre. O estudo de caso foi conduzido por novatos, que, ao final da análise foram entrevistados. Esses novatos relataram um conjunto de problemas e obstáculos enfrentados por eles durante os primeiros contatos com o projeto.

Nakakoji et al. [Nakakoji et al. 2002] estudam quatro projetos de software livre para analisar a evolução da comunidade desses projetos. O estudo apresenta oito papéis encontrados para os membros de um projeto e os apresenta na forma de uma cebola, o chamado *onion patch*. A hipótese desse modelo indica que novatos geralmente iniciam pelas camadas mais externas do modelo e vão em direção ao centro de acordo com seus objetivos. Esses artigos tratam da entrada e evolução da participação de membros em comunidades de software livre, mas nenhum dos estudos foca em novatos que desejam realizar contribuições casuais e não se tornar membros dos projetos.

Com relação às contribuições casuais, diversos autores [Pham et al. 2013a, Gousios et al. 2014, Vasilescu et al. 2015, Pham et al. 2013b] citam em suas pesquisas a existência e crescimento desse fenômeno, mas nenhum dos trabalhos endereça diretamente essa questão. Os autores destacam que é importante observar o impacto destas contribuições para os projetos e os benefícios que trazem para a comunidade, apontando o tópico como um desafio em aberto.

Dada a recente identificação do fenômeno das contribuições casuais, este trabalho tem por objetivo agregar conhecimento ao estado da arte estudando quantitativamente possíveis casos de contribuições casuais no projeto LibreOffice.

### 3. Método de Pesquisa

Para analisar as contribuições casuais do projeto LibreOffice, foram coletados os dados históricos disponíveis e procedeu-se a análise dos dados de acordo com as questões de pesquisa a serem respondidas. A seguir são apresentados os procedimentos de coleta e análise adotados.

Foram coletados dados do projeto LibreOffice, a partir da lista de email<sup>2</sup>, software gerenciador de tarefas (Bugzilla<sup>3</sup>), sistema de revisão de código (Gerrit<sup>4</sup>) e os registros (logs) do repositório de código-fonte<sup>5</sup>. Para realizar a coleta de dados foram utilizadas

---

<sup>2</sup><http://lists.freedesktop.org/archives/libreoffice/>

<sup>3</sup><https://bugs.documentfoundation.org/>

<sup>4</sup><https://gerrit.libreoffice.org>

<sup>5</sup>[git://anongit.freedesktop.org/libreoffice/core](https://anongit.freedesktop.org/libreoffice/core)

**Tabela 1. Distribuição dos novatos por número de *patches***

<i>patches</i> enviados	# pessoas	% das pessoas	# de <i>patches</i> aceitos	% do total de <i>patches</i>
Até 2	19	27,14%	24	2,36%
Entre 3 e 6	17	24,29%	54	5,32%
Entre 7 e 23	17	24,29%	153	15,07%
Mais de 23	17	24,29%	784	77,24%

ferramentas da suíte MetricsGrimoire<sup>6</sup>, a saber:

- Bicho: utilizada para recuperar e organizar as informações de sistemas gerenciadores de tarefas, neste caso, foi utilizado para o bugzilla e gerrit.
- MailingListStats: recupera os dados de listas de email. Para esta pesquisa, foi utilizada para baixar e armazenar os dados históricos das listas de email que estavam arquivados pelo projeto no formato mbox.

Foram coletados dados das fontes citadas, do período compreendido entre 1 de janeiro de 2012 e 31 de dezembro de 2014. Foram considerados novatos aqueles desenvolvedores que submeteram código pela primeira vez no ano de 2013. Para identificar tais novatos, foi construída uma ferramenta que percorre a lista de desenvolvedores e contribuintes do projeto<sup>7</sup> e analisa os emails referentes à declaração de licença (do inglês, *license statement*), buscando por aqueles datados de 2013. A declaração de licença consiste em mensagem enviada pelo desenvolvedor à lista de emails do projeto declarando que todas as suas contribuições estarão sob licença de software livre. Todos os desenvolvedores que enviaram a declaração de licença no ano de 2013 foram considerados novatos e tiveram suas interações com o projeto analisadas.

Para cada um dos novatos foram coletados os seguintes dados, que foram analisados para responder às questões de pesquisa:

- alterações de códigos (*patches*) enviados pelo novato para revisão a partir da ferramenta Gerrit;
- emails enviados pelo novato para a lista de desenvolvedores e respostas recebidas da comunidade;
- tarefas enviadas e comentários feitos pelo novato, a partir da ferramenta Bugzilla;
- *patches* enviados para revisão e comentários realizados nesses *patches*, a partir da ferramenta Gerrit;

Os dados dos emails, tarefas e revisão de código identificados foram inspecionados manualmente para auxiliar no entendimento do fenômeno no contexto do projeto.

## 4. Resultados

A coleta de dados retornou um total de 121 contribuintes considerados novatos. Foram desconsiderados 51 novatos que nunca enviaram código, resultando em 70 novatos. A Tabela 1 sumariza a distribuição dos novatos por número de *patches* enviados e também a quantidade total de patches aceitos por cada agrupamento.

### 4.1. Questão 1

Para responder à questão “É possível encontrar indícios da existência de contribuintes casuais?”, primeiramente foram comparadas a quantidade de solicitações de mudanças

<sup>6</sup><https://metricsgrimoire.github.io/>

<sup>7</sup><https://wiki.documentfoundation.org/Development/Developers/#Individuals>

submetidas e a quantidade atribuída ao novato. Como essas quantidades eram idênticas, apenas os dados de submissões foram utilizados na análise. Para este estudo, foram considerados contribuintes casuais aqueles que enviaram até dois *patches* em 2013, sem enviar mais contribuições até o final de 2014.

Dos 70 novatos considerados, 11 submeteram um único *patch* (15,7% do total). Se considerarmos contribuintes casuais indivíduos que submeteram até dois *patches*, tem-se o total de 19 pessoas, ou 27,1% do total. É importante destacar que todas as pessoas que enviaram apenas um *patch* tiveram suas contribuições aceitas. Com relação às oito pessoas que enviaram duas contribuições, apenas três tiveram um *patch* aceito e um *patch* negado. Porém, durante a inspeção manual descobriu-se que, em um dos casos, seu *patch* negado estava duplicado em outra solicitação de mudança, sob nome de outro desenvolvedor. Portanto, apenas dois novatos tiveram apenas uma submissão aceita.

Assim, como resposta à Questão 1, foram encontrados indícios da existência de contribuintes casuais já que, para o caso do projeto LibreOffice, com novos contribuintes identificados em 2013, 27,1% do total de novatos enviaram 1 ou 2 propostas de *patch* e não retornaram ao projeto.

#### 4.2. Questão 2

Para responder à questão “*Qual o esforço que tais contribuições demandam da comunidade?*”, foram analisadas a quantidade de interações que o novato realiza com membros da comunidade por meio da lista de emails, gerenciador de tarefas e ferramenta de revisão de códigos (gerrit), além daquelas que fazem parte do processo de contribuição (como declaração de *statement*, revisão de código enviado e confirmação de aceite do *patch*).

Na Tabela 2 são apresentadas a quantidade de interações entre o novato e a comunidade. Para isso, foram analisadas a lista de emails e a ferramenta de revisão de código (gerrit). Não foram representadas as interações por meio do gerenciador de tarefas, pois, analisando essa ferramenta, não foram encontradas tarefas ou defeitos enviados pelos novatos e apenas um novato (C17) possuía tarefa atribuída a ele.

Analisando a quantidade de emails enviados, é possível observar que apenas quatro novatos (C5, C8, C14 e C17) enviaram mais de um email para a lista de desenvolvedores, sendo que três deles enviaram apenas 2, tendo sido respondidos por membros da comunidade (como pode ser visto na coluna “Respostas aos emails”). Foi conduzida uma análise manual, confirmando que todos os novatos enviaram a declaração de licença. Ainda durante a análise manual, identificou-se que: o novato C5 se apresentou para a comunidade e propôs uma nova funcionalidade; C8 pediu ajuda para iniciar, explicitamente solicitando auxílio na escolha de uma tarefa fácil; e C14 solicitou ajuda para compilar o código em seu ambiente local. Nos três casos, os contribuintes receberam duas mensagens de resposta da comunidade. O novato C17 foi o único caso diferenciado. Ele/ela propôs um projeto do *Google Summer of Code* em março de 2013 e trabalhou nesse projeto, tendo enviado emails relacionados ao projeto até outubro de 2013.

As duas últimas colunas da tabela representam a quantidade de mensagens trocadas na ferramenta de revisão de código nos *patches* enviados pelos novatos. Pode-se perceber que cada *patch* recebeu, no mínimo, dois comentários que, de acordo com o processo de revisão do projeto, é o número mínimo de mensagens possíveis. As mensagens correspondem a uma aprovação do revisor e a uma mensagem automática do *commit* para

**Tabela 2. Distribuição dos novatos por número de patches**

Novato	Emails enviados	Respostas aos emails	Patches (gerrit)	Comentários nos patches	
				Outros membros**	Novato
C1	1	0	1	4	0
C2	1	0	1	2	0
C3	1	0	1	4	2
C4	1	0	1	4	1
C5	2	2	1	2	0
C6	1	0	1	2	0
C7	1	0	1	6	0
C8	2	2	1	2	1
C9	1	0	1	2	0
C10	1	0	1	3	0
C11	1	0	1	2	0
C12*	1	0	2	4 (4/2)	2
C13*	1	0	2	4 (2/2)	0
C14*	2	2	2	11 (3/8)	7
C15*	1	0	2	4 (2/2)	0
C16*	1	0	2	4 (2/2)	1
C17*	11	7	2	6 (4/2)	1
C18*	1	0	2	5 (2/3)	1
C19*	1	0	2	8 (4/4)	0

\*contribuintes com 2 patches enviados

\*\*os números entre parênteses representam a quantidade de comentários por patch enviado

o sistema de versão. Dos 27 patches enviados para revisão, 15 (55.56%) receberam o número mínimo de comentários, o que significa que as contribuições estavam em conformidade. Na média, cada submissão recebeu 2,93 comentários. Houve apenas dois casos em que uma submissão rendeu mais de 4 comentários por parte da comunidade.

Por meio desta análise preliminar, percebeu-se que o esforço que as contribuições casuais demandam da comunidade é relativamente pequeno. Foram poucos os novatos que solicitaram ajuda pela lista de email e, dentre os que solicitaram, as respostas foram simples, direcionando-os a recursos existentes no wiki do projeto. Com relação à revisão dos códigos submetidos, apenas um patch submetido, dentre os 27 analisados, teve uma quantidade maior de interações (C14, com 8 comentários de membros da comunidade).

## 5. Limitações

Como em qualquer pesquisa empírica, este trabalho tem algumas limitações. Primeiramente, foi escolhido apenas um projeto e um período de tempo para conduzir essa pesquisa preliminar. As conclusões e discussões apresentadas são específicas para o projeto. Sendo assim, não foi possível analisar a influência de variáveis como tamanho do projeto, quantidade de participantes, maturidade do projeto e domínio da aplicação. Para um resultado com conclusões para uma população mais ampla, é necessário analisar uma amostra significativa de projetos e períodos de verificação. Como trabalho futuro, pretende-se realizar uma pesquisa com outros projetos, incluindo projetos hospedados pelo Github.

Em segundo lugar, foram analisados apenas os dados dos projetos disponíveis *online* para realizar uma análise quantitativa. Para complementar a análise quantitativa, os emails trocados pelos novatos e as interações pelas outras ferramentas foram analisados manualmente. Uma pesquisa qualitativa utilizando dados coletados por meio de entrevistas com os contribuintes casuais pode revelar mais detalhes sobre esse comportamento.

As medidas utilizadas neste artigo podem não ser a melhor maneira de mostrar os resultados, podendo ser interpretada de maneiras diferentes. Não foram encontrados trabalhos que ofereçam outros meios para medição ou que possibilitem comparação ou confirmação dos resultados obtidos. Além disso, foram considerados contribuintes casuais aqueles desenvolvedores que submeteram até 2 patches para o projeto, já que não

existem valores de referência na literatura que definam contribuintes casuais de acordo com a quantidade de *patches* submetidos por período de tempo.

As janelas de tempo escolhidas podem ter afetado as observações. Alterar o tamanho dos períodos de tempo ou alterar o início e fim dos intervalos de avaliação podem gerar observações diferentes. Usuários podem ter dois *logins* no gerenciador de tarefas ou participar da lista de desenvolvedores com dois emails diferentes. Pode, ainda, haver casos em que o mesmo usuário está cadastrado no gerenciador de tarefas e na lista de desenvolvedores com emails diferentes.

## 6. Conclusões

Neste artigo foi apresentado um estudo preliminar sobre o fenômeno das contribuições casuais, no escopo do projeto LibreOffice. Para este estudo buscou-se verificar a frequência desse tipo de contribuinte, bem como o esforço que tais contribuições demandam da comunidade. Foram analisados quantitativamente dados históricos extraídos da lista de emails de desenvolvedores, gerenciador de tarefas e gerenciador de revisões de código.

Considerando como contribuintes casuais aqueles desenvolvedores que enviaram até duas submissões de mudanças de código (*patches*) para o projeto LibreOffice no ano de 2013. Foram identificados 19 contribuintes que se encaixam nessa classificação, o que representa 27,14% dos 70 novatos identificados em 2013. Com relação ao esforço que essas contribuições demandam da comunidade, percebeu-se que poucos desses 19 novatos trocaram mensagens pela lista de emails, solicitando ajuda dos membros e também que, na média, a quantidade de mensagens solicitando alterações dos códigos submetidos ao sistema de revisão de códigos foi baixa.

Entende-se que as comunidades de software livre precisam utilizar o fenômeno das contribuições casuais a seu favor, beneficiando-se das possibilidades oferecidas. Visto que este fenômeno tem se tornado comum em projetos de software livre, é importante estudar o custo/benefício deste tipo de contribuição a fim de propor mudanças nos processos de contribuição de novatos de forma a facilitar e fomentar as contribuições.

Como trabalhos futuros, pretende-se entender que benefícios as contribuições casuais trazem para as comunidades. Pretende-se ainda conduzir estudos qualitativos, utilizando dados provenientes de entrevistas com contribuintes casuais e com membros da comunidade para melhor explorar o fenômeno. Além disso, estão sendo conduzidas pesquisas com projetos hospedados no Github, para aumentar a abrangência da pesquisa.

## Agradecimentos

Os autores agradecem à Fundação Araucária, UTFPR, CNPq (processos universais 461101/2014-9 e 477831/2013-3), NAPSoL-PRP-USP e FAPESP pelo apoio financeiro.

## Referências

- Bird, C. (2011). Sociotechnical coordination and collaboration in open source software. In *Proceedings of the 2011 27th IEEE International Conference on Software Maintenance*, pp 568–573. IEEE Computer Society.
- Carmel, E. and Tjia, P. (2005). *Offshoring information technology: Sourcing and outsourcing to a global workforce*. Cambridge University Press.

- Ducheneaut, N. (2005). Socialization in an open source software community: A socio-technical analysis. *Computer Supported Cooperative Work*, 14(4):323–368.
- Fagerholm, F., Johnson, P., Guinea, A. S., Borenstein, J., and Münch, J. (2014). Onboarding in open source projects. *IEEE Software*, 31(6):54–61.
- German, D. M. (2003). The gnome project: a case study of open source, global software development. *Software Process: Improvement and Practice*, 8(4):201–215.
- Gousios, G., Pinzger, M., and Deursen, A. v. (2014). An exploratory study of the pull-based software development model. In *Proceedings of the 36th International Conference on Software Engineering*, pp 345–355. ACM.
- Jensen, C., King, S., and Kuechler, V. (2011). Joining free/open source software communities: An analysis of newbies’ first interactions on project mailing lists. In *Proceedings of the 44th Hawaii International Conference on System Sciences*, pp 1–10. IEEE.
- Nakakoji, K., Yamamoto, Y., Nishinaka, Y., Kishida, K., and Ye, Y. (2002). Evolution patterns of open-source software systems and communities. In *Proceedings of the International Workshop on Principles of Software Evolution*, pp 76–85. ACM.
- Park, Y. and Jensen, C. (2009). Beyond pretty pictures: Examining the benefits of code visualization for open source newcomers. In *Proceedings of the 5th IEEE International Workshop on Visualizing Software for Understanding and Analysis*, pp 3–10. IEEE.
- Pham, R., Singer, L., Liskin, O., Figueira Filho, F., and Schneider, K. (2013a). Creating a shared understanding of testing culture on a social coding site. In *Proceedings of the 2013 International Conference on Software Engineering*, pp 112–121. IEEE Press.
- Pham, R., Singer, L., and Schneider, K. (2013b). Building test suites in social coding sites by leveraging drive-by commits. In *Proceedings of the 2013 International Conference on Software Engineering*, pp 1209–1212. IEEE Press.
- Steinmacher, I., Conte, T., Gerosa, M. A., and Redmiles, D. F. (2015a). Social barriers faced by newcomers placing their first contribution in open source software projects. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pp 1–13. ACM.
- Steinmacher, I., Silva, M. A. G., Gerosa, M. A., and Redmiles, D. F. (2015b). A systematic literature review on the barriers faced by newcomers to open source software projects. *Information and Software Technology*, 59:67–85.
- Stol, K.-J., Avgeriou, P., and Babar, M. A. (2010). Identifying architectural patterns used in open source software: approaches and challenges. In *Proceedings of the 14th International conference on Evaluation and Assessment in Software Engineering*, pp 91–100. British Computer Society.
- Vasilescu, B., Filkov, V., and Serebrenik, A. (2015). Perceptions of diversity on github: A user survey. In *Proceedings of the 2015 8th International Workshop on Cooperative and Human Aspects of Software Engineering*. IEEE.
- von Krogh, G., Spaeth, S., and Lakhani, K. R. (2003). Community, joining, and specialization in open source software innovation: A case study. *Research Policy*, 32(7):1217–1241.
- Zhou, M. and Mockus, A. (2015). Who will stay in the floss community? modelling participant’s initial behaviour. *IEEE Transactions on Software Engineering*, 41(1):82–99.