Analisando as Contribuições da Comunidade Open Source Brasileira em Projetos Distribuídos de Software Um Estudo Inicial

Gustavo Pinto[1], **Fernando K. Kamei**[1,2]

[1]Centro de Informática – Universidade Federal de Pernambuco (UFPE)
50.740-560 – Recife – PE – Brasil

[2]Instituto Federal do Sertão Pernambucano (IF Sertão-PE)
56.200-000 – Ouricuri – PE – Brasil

{ghlp, fkk}@cin.ufpe.br

Abstract. *In the last years, the software development has demanded improvements on techniques and tools in order to facilitate the communication of distributed teams. In this context, social coding environments have gained attention of researchers and mainly of open source communities, due to the fact that its development process is necessarily distributed. This study aimed to understand how the contributions are made by Brazilians users in a distributed software development open source environment. We analyzed the activities of more than 4.000 developers and more the 15.000 projects during one year. The results show that the Brazilian community has weak activity, although of performing various types of contributions.*

Resumo. *Nos últimos anos, o desenvolvimento de software tem exigido a melhoria de técnicas e ferramentas que facilitem a comunicação de equipes distribuídas. Nesse contexto, ambientes de social coding vêm ganhando atenção de pesquisadores e principalmente das comunidades open source, devido ao fato do seu processo de desenvolvimento ser necessariamente distribuído. Este estudo objetivou entender como são realizadas as contribuições de brasileiros em ambientes distribuídos de software open source. Foram analisadas as atividades de mais de 4 mil desenvolvedores, e mais de 15 mil projetos no período de um ano. Os resultados mostram que a comunidade brasileira é pouco ativa, apesar de realizar vários tipos de contribuições.*

1. Introdução

Como reflexo da globalização dos negócios e do crescimento da economia, ocorreu uma migração do mercado local para o mercado global, criando novas formas de competição e colaboração. Esses fatores também atingiram o mercado de software. Neste contexto, surge o Desenvolvimento Distribuído de Software (DDS), onde as equipes de desenvolvimento estão distribuídas em diferentes localizações [Carmel 1999]. Segundo [Prikladnicki 2012], a distribuição do processo de desenvolvimento de software faz ampliar os problemas inerentes ao desenvolvimento tradicional e gera novos desafios ao adicionar distância física, dispersão temporal e diferenças culturais. Algumas maneiras de enfrentar esses desafios é a adoção de processos, técnicas e ferramentas de suporte ao DDS, que objetivem a efetiva comunicação e coordenação de equipes distribuídas.

Neste cenário de DDS destacam-se as comunidades open source, que emergem através do compartilhamento do código fonte dos projetos, desenvolvidos de maneira colaborativa. Ademais, desenvolvedores tem testemunhado o crescimento de plataformas para contribuição de projetos open source, como Github[1] e BitBucket[2]. Estes ambientes chamados de *social coding* oferecem inúmeras funcionalidades, possibilitando que grupos distribuídos possam trabalhar em equipe em um mesmo projeto, facilitando a colaboração. Dessa forma, o desenvolvimento de projetos open source, através do uso dessas ferramentas, tornou-se intrinsecamente distribuído.

No entanto, enquanto que por um lado o acesso ao código é extremamente facilitado, aspectos como a compreensão das comunidades e a análise das contribuições são tópicos que ainda precisam ser melhores explorados, uma vez que grande parte do sucesso da filosofia open source advém das diversas formas de contribuições das comunidades. Dessa forma, as comunidades e suas contribuições desempenham papel importante no desevolvimento distribuído de software open source.

O presente trabalho tem como objetivo investigar as principais formas de contribuições, bem como o perfil das comunidades open source brasileiras que interagem no *social coding* GitHub. Para tanto, este trabalho responderá as seguintes perguntas de pesquisa:

(i) A comunidade brasileira de desenvolvimento distribuído open source é ativa? Este estudo classificou a atuação da comunidade brasileira como pouco ativa. Os dados mostraram que apenas 30% dos usuários brasileiros registrados no GitHub tiveram ao menos uma contribuição durante o período de um ano. No entanto, deste grupo de contribuidores, cerca de 40% realizaram 20 ou mais contribuições ao longo de um ano. Supreendentemente, este estudo identificou que a maioria dos usuários ativos no GitHub são da Região Nordeste (46% do total), além de participarem, em sua maioria, da iniciativa privada (83,65%).

(ii) Quais são as formas de contribuição mais comumente realizadas? Este estudo analisou as principais contribuições públicas que os desenvolvedores podem fazer no GitHub. Destas, como esperado, as mais utilizadas por brasileiros são os commits (95% do total), seguidos de pull requests (4% do total), e abrir e fechar issues (1% do total), respectivamente. Ademais, foi observado que boa parte dos projetos contribuídos são ativos, ou seja, continuam recebendo contribuições ao longo dos meses.

O restante do trabalho está estruturado da seguinte maneira: na Seção II são apresentados informações preliminares sobre o GitHub; na Seção III é apresentado as questões de pesquisa, e como a pesquisa foi conduzida para realizar a coleta, e análise dos dados; na Seção IV são apresentadas os resultados e discussões para as questões de pesquisa; na Seção V é apresentada uma discussão sobre trabalhos relacionados; por fim, na Seção VI apresentamos as conclusões e os trabalhos futuros.

2. Background

Ambientes de *social coding* têm fornecido oportunidades para que pesquisadores possam responder diversas questões ligadas ao desenvolvimento de software. Durante os últimos

[1]http://www.github.com
[2]http://www.bitbucket.org

anos, o GitHub se tornou uma das mais populares plataformas de colaboração em projetos de desenvolvimento distribuído de software [Gousios and Spinellis 2012], contendo mais de 4 milhões de repositórios de software, e mais de 10 milhões de desenvolvedores.

Criado em cima do sistema de versionamento de código `Git`, adicionou funcionalidades que não estão presentes no software base. No GitHub, um dado desenvolvedor pode participar de vários projetos e cada projeto pode ter mais de um desenvolvedor. Ainda é possível utilizar de recursos como wiki, issues e review de código. Dentro do GitHub existem páginas para desenvolvedores e páginas para usuários[3]. A página de usuário inclui informações sobre as contribuições realizadas, bem como a quantidade seguidores, a quantidade de desenvolvedores que ele segue, o número de projetos forks (projeto que é uma cópia independente de um outro projeto), o número de projetos que ele está acompanhando (watching), dentre outros. Demais tipos de contribuições podem ser consultados através de uma API pública[4]. Esta transparência é um dos pontos fortes de ambientes de *social coding*.

O termo "contribuição" pode ser utilizado para expressar inúmeras atividades do processo de desenvolvimento de software. Dentro da plataforma de *social coding* GitHub, os desenvolvedores podem realizar outros tipos de contribuições, como comentar dentro de uma issue ou em uma específica linha de código. Para este trabalho, porém, apenas as contribuições públicas[5] foram analisadas: (i) Commits, (ii) pull requests, e (iii) abrir e fechar issues. A seguir, é apresentado o conceito de cada contribuição:

- **Commit**: é uma modificação realizada em um artefato do projeto, usualmente o código fonte. Em um commit pode haver inúmeras modificações, mas é recomendado que apenas as modificações relacionadas a uma determinada atividade sejam agrupadas em um mesmo commit.
- **Pull Request**: é a modificação realizada por um usuário (commit) que é enviada para os mantenedores do projeto, para que estes avaliem e, por ventura, aprovem ou não a modificação. Se aprovada, a modificação será incorporada ao repositório oficial do projeto. Um pull request pode agrupar um ou mais commits.
- **Abrir/Fechar Issue**: é o controle de atividades (correções de bug, melhorias, novas funcionaldiades, etc.) que devem ser adicionadas ao projeto. Quando novas, quando uma nova issue é adicionada ao projeto, esta é classificada como *open*, e quando resolvida, e quando resolvida, seu status muda para *closed*.

3. Metodologia

Este trabalho está interessado em descobrir como os desenvolvedores brasileiros contribuem em projetos distribuídos de software open source. Para atingir este objetivo, as seguintes questões e sub-questões de pesquisa serão respondidas:

1. A comunidade brasileira de desenvolvimento distribuído de software open source é ativa?
   (a) Quais são as características dos projetos contribuídos?
   (b) Quais são as comunidades mais ativas?
2. Quais são as formas de contribuição mais comumentes realizadas?

[3]Um exemplo de página de desenvolvedor pode ser em visualizada: https://github.com/gustavopinto
[4]http://api.github.com
[5]https://help.github.com/articles/why-are-my-contributions-not-showing-up-on-my-profile

3.1. Coletando Dados do Github

Este trabalho realizou a coleta dos dados em duas etapas. A primeira, responsável por identificar os usuários brasileiros, e a segunda, para coletar as informações destes usuários. Para identificar os usuários brasileiros, foi realizada uma busca através do campo de usuário *Location*, que pode pode ser preenchido com a sua localidade. Exemplos de localidades brasileiras incluem "Brasil", "Brazil", e "São Paulo".

Uma limitação do GitHub é que por mais que a busca retorne um total de, por exemplo, 20.000 desenvolvedores para uma dada localidade, apenas os primeiros 1.000 registros podem ser coletados. Dessa forma, não é possível coletar informações de todos os usuários de uma localidade. Uma forma encontrada para minimizar este problema foi modificando o valor inserido no campo *Location*. Por exemplo, todos os nomes dos Estados e das capitais brasileiras foram utilizados como entrada para busca. Ao fim do processo, os usuários duplicados foram removidos, e o total de 4.481 perfis de usuários foram coletados.

Posteriormente, as contribuições destes desenvolvedores foram coletadas utilizando os seguintes passos: para cada desenvolvedor, é feito um request para a página de perfil deste usuário, então é feito o download do conteúdo da página web e são extraídas informações como o número de seguidores, projetos e contribuições. Os dados deste estudo estão baseados nas contribuições realizadas entre 16/05/2012 e 15/05/2013.

4. Resultados e Discussões

Nesta seção são apresentados os resultados que permitem responder as questões de pesquisa.

4.1. Questão 1: A comunidade brasileira de desenvolvimento distribuído de software open source é ativa?

Foram identificados 4.481 usuários brasileiros, destes, 3.121 (cerca de 70% do total) usuários não possuíam nenhuma contribuição no período analisado, sendo classificados como inativos. Os demais usuários, os ativos, foram então sub-divididos em três novos grupos: pouco ativos (até 19 contribuições em um ano), razoavelmente ativos (até 49 contribuições em um ano) e muito ativos (acima de 49 contribuições em um ano). A escolha destes valores teve como referência o resultado do terceiro quartil (41 contribuições) do total de contribuições. Portanto, o percentual dos grupos de usuários pouco ativos, razoavelmente ativos e muito ativos são, respectivamente, 57,8%, 20,9%, 21,3%, mostrando que a comunidade brasileira de desenvolvimento distribuído de software open source é pouca ativa. As demais análises deste trabalho levam em consideração apenas o grupo de usuários ativos.

4.1.1. Questão 1.1: Quais são as características dos projetos contribuídos?

Foram identificados 15.595 projetos, dos quais 6.290 (40,33%) são forks e 9.305 (59,67%) não são forks. A Figura 1 fornece uma visão geral sobre esses projetos.

Na Figura 1-(a) é possível perceber que, cerca de 75% dos projetos forks analisados foram criados há um pouco mais de 2 anos. Considerando o período de um ano,

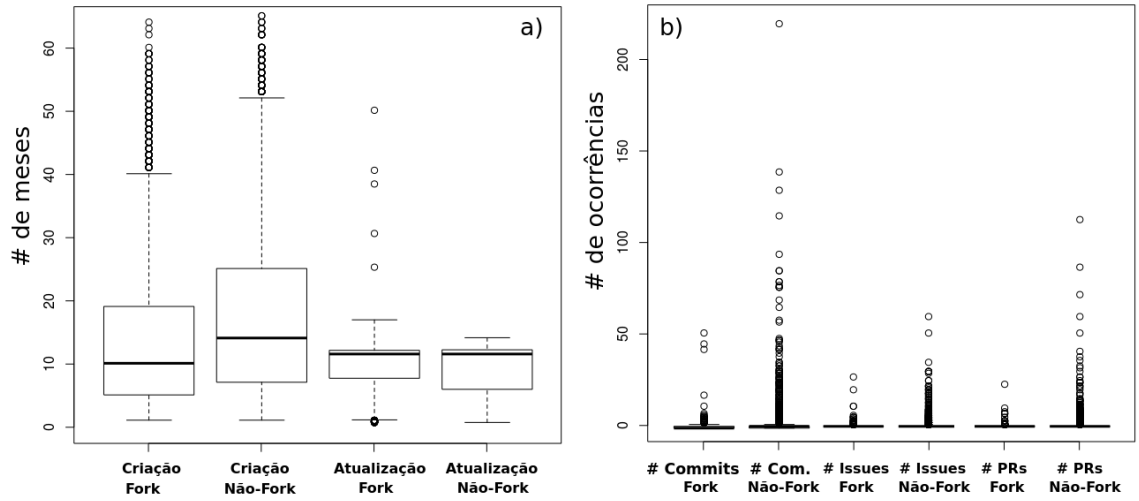

Figura 1. Características gerais sobre os projetos contribuídos ao longo de um ano. A Figura a) tem informações sobre as datas de criação e última atualização dos projetos em meses. A Figura b) contém as informações sobre a distribuição dos commits, issues e pull requests (PRs).

63,78% dos projetos forks já haviam sido criados, enquanto que para este mesmo período, os não-forks somam 51,59%. Porém, um dado interessante está relacionado a atualização desses projetos. É possível perceber que 75% dos projetos tiveram a sua última atualização há um pouco mais de 2 meses, indicando que, a maioria dos projetos criados receberam ao menos uma atualização recentemente.

Na Figura 1-(b) percebe-se novamente um comportamento bem similar entre projetos forks e não-forks. No entanto, observa-se que os projetos originais (não-forks) recebem mais contribuições, principalmente commits e pull requests, do que de forks, o que era esperado. Em resumo, os projetos originais que tiveram atualização há menos 3 meses representam 29,48%, e 70,52% de 3 a 6 meses. Em contrapartida, 24,15% dos projetos forks tiveram atualizações há menos de 3 meses e 75,74% de 3 à 6 meses. Esses dados reforçam o fato de que os projetos, ao contrário dos usuários, são ativos, tanto os originais quanto forks.

Também foi possível analisar a popularidade das linguagens de programação, uma vez que o GitHub infere qual a linguagem mais utilizada em um projeto. Levando em consideração os projetos originais e forks, as 10 linguagens mais utilizadas são: JavaScript (22,77%), Ruby (18,78%), Java (14,19%), PHP (13,95%), Python (9,86%), C (3,36%), C++ (2,94%), Shell (2,55%), Objective-C (1,95%) e C# (1,88%).

A partir desses resultados, é possível observar que, apesar dos desenvolvedores contribuirem principalmente com linguagens consolidadas como JavaScript e Java, novas comunidades, como Ruby, vem ganhado espaço. Um dos motivos pelo qual JavaScript apareceu como a mais utilizada é devido ao fato de que muitos projetos são web, e utilizam-a para manipulação de página. No entanto, é comum que estes projetos também utilizem de outras linguagens na camada de negócio do código. Dessa forma, JavaScript pode ser facilmente utilizada com uma sub-linguagem em um projeto.

Finalmente, uma análise de popularidade dos projetos foi realizada. Este trabalho

considerou que, para um projeto ser considerado popular, o projeto deveria possuir ao menos uma issue (aberta ou fechada), um download, mais que 3 watchers e pelo menos um fork. Com isto, 836 projetos foram classificados como populares. Destes, apenas 65 (7,78%) projetos são forks, o que mostra que alguns forks foram capazes de evoluir a ponto de se tornar reconhecido. Ao total, o número de projetos populares é de 8,28% (projetos originais) e 1,03% (forks).

4.1.2. Questão 1.2: Quais são as comunidades mais ativas?

Ao ser analisada a localidade da comunidade brasileira open source, 46% desses estão localizados na Região Nordeste, 25% na Região Sul, 12% no Sudeste, 12% no Centro-Oeste e apenas 5% no Norte. Em uma análise por Estado, o maior número de usuários ativos encontra-se no Paraná (13,65%), seguido do Ceará (12,37%) e Pernambuco (11,25%). A Figura 2 apresenta os dados das contribuições por Estado.

É importante ressaltar que 92% dos usuários ativos preencheram o campo localidade da sua página de perfil, o que torna estes resultados confiáveis. Levando em consideração o total de contribuições por Estado, Pernambuco é o que mais tem contribuído, com 16,46% das contribuições realizadas no período analisado. Seguido pelos Estados do Paraná (10,63%), Bahia (10,43%), Ceará (10,27%) e do Distrito Federal (10,13%). Os demais Estados juntos somaram 42,04% do total.

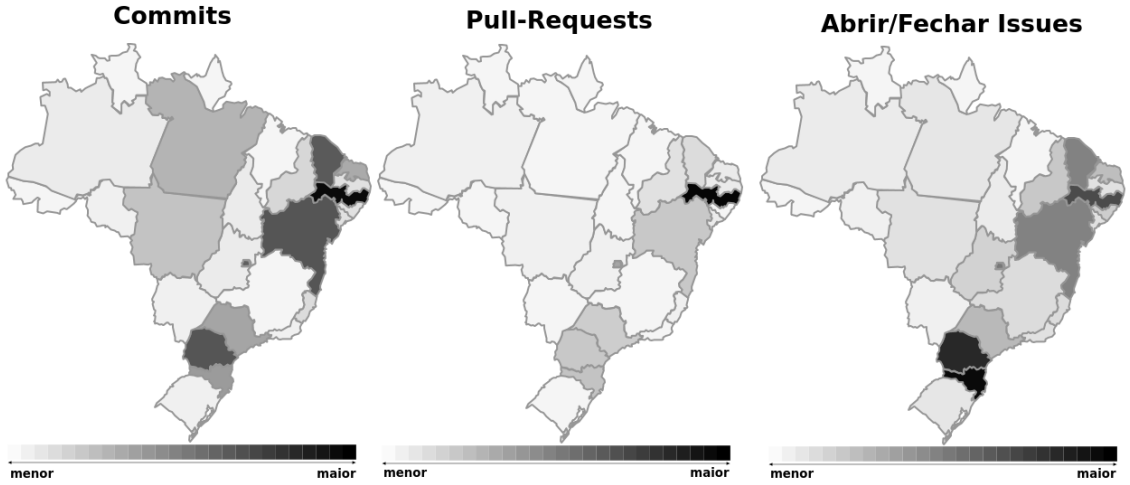

Figura 2. Dados demográficos das contribuições por Estado.

Nesse estudo também foi possível mapear as comunidades com base nas linguagens de programação mais utilizadas pelos usuários, uma vez que o GitHub infere qual a linguagem mais utilizada por um usuário. As 10 linguagens inferidas pelo GitHub como mais utilizadas foram: JavaScript (18,97%), Java (17,42%), Ruby (15,88%), PHP (13,23%), Python (7,72%), C++ (3,01%), C (2,94%), C# (1,98%), Shell (1,83%) e Objective-C (1,32%).

Este resultado é similar ao apresentado para os projetos, com algumas ressalvas. Apesar da linguagem Ruby apresentar maior número de projetos do que a linguagem Java, o número de desenvolvedores é menor, sugerindo que a comunidade Ruby é mais

ativa com relação a comunidade Java. Do total, 10,51% dos usuários não tiveram sua linguagem inferida. Finalmente, foi feita uma análise para identificar o vínculo educacional/empregatício do usuário. Foi observado que, apenas 716 informaram algum tipo de vínculo. Destes, 83,65% estavam vinculados a uma empresa privada, e os demais 16,34% estavam relacionados a algum órgão governamental, destes a maioria ligados a universidades públicas.

4.2. Questão 2: Quais são as formas de contribuição mais comumentes realizadas?

No período analisado, foram identificadas um total de 58.890 contribuições em mais de 15.500 projetos, destas, a sua maioria foram realizadas a partir de commits, representanto 95,50%. As solicitações de pull requests aparecem em seguida, com 3,52% e, por fim, as aberturas e fechamento das issues com 0,98%. A Figura 3 sumariza o total de contribuições por categoria no período.

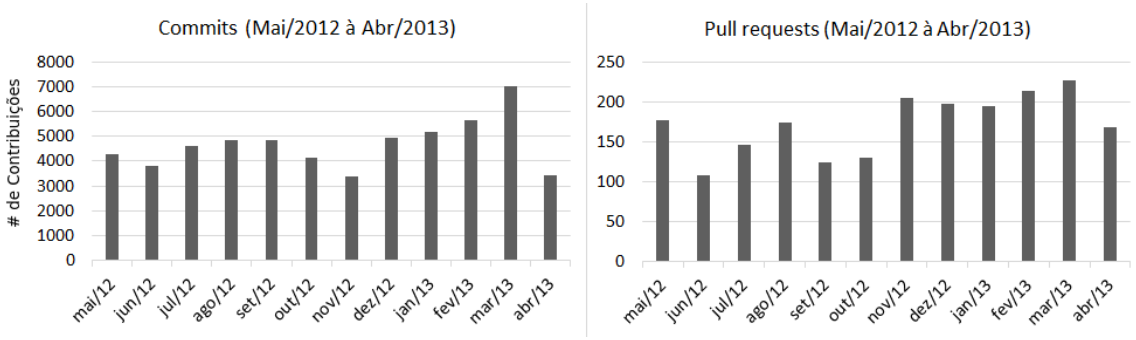

Figura 3. Contribuições por categoria ao longo de um ano.

Como pode ser visto na Figura 3, o número de commits de um dado mês chega a ser 40 vezes maior quando comparado ao número de solicitações de pull request do mesmo período. Uma das principais causas para tal, é o fato de que um único pull request pode conter vários commits. Ademais, durante o período em que os mantenedores do projeto analisam o pull request, outros commits podem ser requisitados para que a feature esteja de acordo com o padrão de codificação adotado.

Um outro ponto importante está relacionado a frequência com a qual as contribuições são realizadas. Diferentemente do esperado, o número de contribuições apresenta um crescimento ao longo do ano, independentemente do tipo da contribuição. Por outro lado, meses como Dezembro e Janeiro aparentam ser atípicos de contribuições, quando observado apenas commits. O gráfico de issues foi omitido pela escala ser cerca de 100 vezes menor que a de commits. De acordo com os resultados apresentados, as maiores contribuições estão relacionadas à commits.

5. Trabalhos Relacionados

A análise de sistemas distribuídos open source é um tópico bastante explorado na engenharia de software nos últimos anos [Gousios and Spinellis 2012]. Muitos destes trabalhos estão interessados em avaliar custos de desenvolvimento [Amor et al. 2006], produtividade e performance do desenvolvedor [Koch and Schneider 2000], e até técnicas de desenvolvimento ágil [Warsta and Abrahamsson 2003]. No entanto, nenhum trabalho que

se interesse sobre as principais formas de colaboração de comunidades open-source em ambientes distribuídos foi encontrado.

Na literatura podem ser encontrados outros trabalhos que analisam ambientes de DDS como uma estrutura de redes. O trabalho mais próximo é o de [Surian et al. 2010], que extraí padrões de colaboração no SourceForge em formato de gráfos com alto nível de detalhamento, a partir da análise de contribuições de atributos não relacionados ao código. No entanto, este trabalho não é capaz de identificar as principais comunidades, seja baseada em locais ou por linguagem de programação.

6. Conclusões e Trabalhos Futuros

Neste trabalho foi apresentado um estudo inicial para entender como são realizadas as contribuições de brasileiros em projetos distribuídos de software open source, a partir da análise dos três principais tipos de contribuições públicas do GitHub. Com base nos resultados obtidos é possível afirmar que a comunidade brasileira é pouco ativa: 30% dos usuários são responsáveis por 100% das contribuições. Dentre as comunidades de software, as que mais se destacam ainda estão relacionadas as linguagens mais tradicionais, como JavaScript e Java, além da sua maioria participar da iniciativa privada (84%).

Como trabalho futuro, espera-se analisar outros repositórios de forma a criar relacionamento entre estes dados, a fim de responder perguntas como: (i) Por que determinadas regiões podem estar contribuindo menos?; (ii) Por que determinadas linguagens estão predominando?; (iii) As comunidades também são atuantes em outras plataformas de *social coding*?

Referências

Amor, J. J., Robles, G., and Gonzalez-Barahona, J. M. (2006). Effort estimation by characterizing developer activity. In *Proceedings of the 2006 international workshop on Economics driven software engineering research*, EDSER, pages 3–6.

Carmel, E. (1999). *Global software teams: collaborating across borders and time zones*. Prentice Hall PTR, Upper Saddle River, NJ, USA.

Gousios, G. and Spinellis, D. (2012). Ghtorrent: Github's data from a firehose. In *Proceedings on Mining Software Repositories*, MSR, pages 12–21. IEEE.

Koch, S. and Schneider, G. (2000). Results from software engineering research into open source development projects using public data. In *Informationswirtschaft, H.R. Hansen und W.H. Janko (Hrsg.), Nr. 22, Wirtschaftsuniversitaet*, page 22.

Prikladnicki, R. (2012). Propinquity in global software engineering: examining perceived distance in globally distributed project teams. *Journal of Software Maintenance*, 24(2):119–137.

Surian, D., Lo, D., and Lim, E.-P. (2010). Mining collaboration patterns from a large developer network. In *Reverse Engineering (WCRE)*, pages 269–273.

Warsta, J. and Abrahamsson, P. (2003). Is open source software development essentially an agile method? In *Proceedings of the 3rd Workshop on Open Source Software Engineering*, pages 143–147.