Uma Análise Preliminar de Projetos de Software Livre que Migraram para o GitHub

Luiz Felipe Dias[1], Igor Steinmacher[1], Igor Wiese[1], Gustavo Pinto[2], Daniel Alencar da Costa[3], Marco Gerosa[4]

[1]Universidade Tecnológica Federal do Paraná (UTFPR)
Campo Mourão – PR – Brasil

[2]Instituto Federal do Pará (IFPA)
Santarém – PA - Brasil

[3]Universidade Federal do Rio Grande do Norte (UFRN)
Natal – RN – Brasil

[4]Universidade de São Paulo (USP)
São Paulo – SP – Brasil

luizdias@alunos.utfpr.edu.br,{igorfs, igor}@utfpr.edu.br

gustavo.pinto@ifpa.edu.br,danielcosta@ppgsc.ufrn.br,gerosa@ime.usp.br

Abstract. *Social coding environments such as GitHub and Bitbucket are changing the way software is built. Not surprisingly, several mature, active, non-trivial open-source software projects are switching their decades of software history to these environments. There is a belief that these environments have the potential of attracting new contributors to open-source projects. However, there is little empirical evidence to support these claims. In this paper, we quantitatively studied a curated set of open-source projects that migrated to GitHub, aiming at understanding whether this migration fostered collaboration. Our results suggest that although interaction in some projects increased after migrating to GitHub, the rise of newcomers is not straightforward. In this preliminary analysis, we could not assess causality, and there may be factors unrelated to the migration influencing the rise of contributors and contributions.*

Resumo. *Ambientes sociais de codificação tais como GitHub e Bitbucket estão mudando a maneira de construir software. Sem grandes surpresas, vários projetos ativos, não triviais e de código aberto, estão migrando suas décadas de história de software para estes ambientes. Há uma crença de que estes ambientes possuem o potencial de atrair novos contribuidores para projetos de código aberto. No entanto, há pouca evidência empírica para apoiar estas alegações. Neste artigo, nós estudamos um conjunto específico de projetos de software livre que migraram para o GitHub, com o objetivo de entender se esta migração promove a colaboração. Nossos resultados sugerem que, embora a interação em alguns projetos aumente após a migração para o GitHub, o aumento do número de novatos não foi tão relevante. Nesta análise preliminar não foi possível estabelecer relações de causa e efeito dos fenômenos, podendo existir fatores não relacionados à migração que influenciam o aumento na quantidade de contribuidores e contribuições.*

1. Introdução

Projetos de software livre introduziram ao desenvolvimento de software uma nova perspectiva quanto ao conceito de contribuição. Voluntariamente, desenvolvedores ao redor do mundo contribuem a projetos de software livre da maneira que podem, de forma a somar seus esforços para a evolução de uma comunidade, de modo geral, ou de um software, em particular. Com o crescimento desta forma de manifestação, novas plataformas também começaram a surgir para melhor apoiar a atividade de desenvolvimento de software livre. Dentre suas principais mudanças, tais ambientes passam a mudar o modo ao qual desenvolvedores de software se comunicam, colaboram e contribuem com projetos de código aberto [Pham et al. 2013a]. Esses ambientes são chamados de ambientes sociais de codificação [Tsay et al. 2014, Pham et al. 2013a, Thung et al. 2013] oferecem aos desenvolvedores algumas funcionalidades de redes sociais, como menções e perfil e a possibilidade de compartilhar as atividades realizadas, seguir atividades de outros desenvolvedores e/ou projetos em um único ambiente web [Thung et al. 2013].

Um dos exemplos mais conhecidos desse tipo de ambiente é o GitHub, que é um serviço de hospedagem para projetos de software livre (gratuito) e proprietários (pago). Este serviço é disponível para projetos que utilizam o sistema de controle de versão *Git*. Contando com mais de 38 milhões de repositórios hospedados e 15 milhões de usuários cadastrados[1], GitHub é considerado não só o maior *website* para hospedagem de códigos do mundo, mas também um dos mais ricos, quando se refere as suas funcionalidades sociais. Além disso, o GitHub é frequentemente usado em estudos recentes de engenharia de software (por exemplo, [Pinto et al. 2016], [Moura et al. 2015], [Tsay et al. 2014]).

Apesar de sua notoriedade, o GitHub não é o único ambiente de codificação utilizado por desenvolvedores de software. Existem diversas opções, como o *CodePlex*, específico para tecnologias Microsoft, e o *BitBucket*, que permite trabalhar com diferentes sistemas de versionamento de código. No entanto, nos últimos meses, tem sido comum a migração de vários sistemas de versionamento para o GitHub, em particular, devido a sua notória popularidade, bem como sua funcionalidades sociais que facilitam a criação e gerenciamento times colaborativos.

O objetivo desse estudo é investigar, quantitativamente, a influência da migração de projetos de software livre para o GitHub, em particular, com relação ao número de contribuintes e de contribuições realizados nestes projetos. Para isso, selecionamos projetos representativos que antes eram hospedados em repositórios tradicionais de codificação, como o *Sourceforge*, e que migraram para o GitHub no decorrer do seu ciclo de vida. Para guiar esta pesquisa, a principal pergunta de pesquisa é:

> ***Q:*** *Quanto o processo de migração para ambientes sociais de codificação impacta na entrada de novos contribuintes e no número de contribuições?*

Para responder à questão de pesquisa, utilizamos dados e meta-dados adquiridos dos repositórios, e buscamos entender se esse processo de migração beneficiou os projetos. Com isso, analisamos número de novos contribuidores, número de contribuidores ativos e contribuições realizadas.

[1] `https://GitHub.com/about/press`

2. Trabalhos Relacionados

Existem trabalhos na literatura que analisam ambientes sociais de codificação. Os trabalhos em questão discutem, sob diferentes perspectivas, os possíveis fatores que possam impactar na migração de projetos. Quanto a este contexto, gostaríamos de ressaltar trabalhos relacionados à: análise de fatores sociais na retenção de novatos, características sociais presentes no GitHub, e o fenômeno dos contribuidores casuais.

`Influência de Fatores Sociais sobre a retenção de novatos:` Alguns estudos na literatura estão focados em analisar a influência de fatores sociais sobre a retenção de novatos em projetos de software livre ([Steinmacher et al. 2015, Zhou and Mockus 2015, Ducheneaut 2005, Bird 2011]). A fim de entender, por meio das redes sociais (e.g., extraídas de listas de emails), com quem os novatos colaboram, e como estas redes evoluíram ao longo dos anos. Jensen et al. (2011) analisaram quatro projetos para entender se novatos costumam ser respondidos rapidamente, se a idade ou nacionalidade dos mesmos impacta no tipo de resposta que recebem e se o tratamento recebido é similar aos dos demais membros do projeto. Apesar dos estudos se concentrarem na relação de aspectos sociais quanto a retenção de novatos, eles não analisam os ambientes sociais de codificação como um meio que possibilite novas contribuições.

`Características Sociais e o GitHub:` Diversos são os estudos voltados a aspectos sociais no GitHub. Marlow et al. (2013) encontraram evidência de que desenvolvedores utilizam de sinais presentes nos perfis do GitHub, tais como habilidades e relacionamentos, para formar primeiras impressões de usuários e projetos. Dabbish et al. (2012) investigaram a influência existente no comportamento de usuários do GitHub, e relataram que o número de observadores em um projeto é um fator que pode atrair novos desenvolvedores. Na sequência, Tsay et al. (2014) evidenciaram que desenvolvedores usam tanto de informações técnicas como sociais para influenciar avaliações em projetos de software livre. O tamanho da comunidade também foi evidenciado como possível indicador de sucesso em projetos de software livre. McDonald e Goggins (2013), ao entrevistar mantenedores de projetos no GitHub, encontraram que as funcionalidades oferecidas pelo GitHub são uma das principais razões do crescimento de contribuições em projetos de software livre. Esses estudos estão focados nos ambientes sociais de codificação como responsáveis pela atração de novos desenvolvedores e pela geração de sinais e impressões entre projetos e desenvolvedores. Entretanto, nenhum destes estudos investiga como a migração para ambientes sociais de codificação influencia a entrada de novatos, e o número de contribuições recebidas.

`Contribuidores Casuais:` Certos trabalhos exploram o fenômeno dos contribuidores casuais no contexto dos ambientes sociais de codificação. Vários autores tem reconhecido a existência e o crescimento deste comportamento [Pham et al. 2013b, Pham et al. 2013a, Gousios et al. 2014, Vasilescu et al. 2015, Pinto et al. 2016]. No entanto, estes trabalhos não analisam o impacto da migração de projetos de software livre para os ambientes sociais de codificação.

3. Método de Pesquisa

Nesta seção, são descritos os projetos selecionados (Seção 3.1), e o processo de coleta e análise dos repositórios (Seção 3.2).

3.1. Projetos

Entende-se por software livre aquele que respeita a liberdade e senso de comunidade dos usuários [Fogel 2013]. De maneira geral, os usuários devem possuir a liberdade de executar, copiar, distribuir, estudar, mudar e melhorar o software [Stallman 1999]. Para representar esta forma de manifestação, foram selecionados três projetos popularmente conhecidos, de domínios diferentes, inicialmente hospedados em um ambiente de desenvolvimento não-colaborativo, mas que migraram para o GitHub em algum momento no seu ciclo de vidas. Ademais, tais projetos são não-triviais, em termos do número de contribuição, contribuintes e tempo de vida. São eles:

- **`Ruby`**, linguagem de programação dinâmica e orientada à objetos. Lançada em 1998, migrou ao GitHub em fevereiro de 2010. Escrita principalmente nas linguagens C e Ruby.
- **`MongoDB`**, banco de dados orientado a documentos. Lançado em outubro de 2007, migrou ao GitHub em janeiro de 2009. Escrito principalmente em C++.
- **`Jenkins`**, serviço de integração contínua. Lançado em novembro de 2006, teve sua data de migração quatro anos após, em novembro de 2010. Escrito principalmente em Java.

Estes projetos foram escolhidos pois: são bem estabelecidos em suas respectivas comunidades, todos com mais de nove anos de existência. Contabilizam um total de contribuições por projeto satisfatório, os três superiores a dez mil. Estão abertos à *pull-requests* de terceiros. Contam com uma média de mais de trezentos contribuidores por projeto, aos quais uma média de 27 contribuidores permanecem ativos mensalmente, e somam em média 280 contribuições mensais. A Tabela 1 apresenta detalhes adicionais sobre os projetos selecionados.

Tabela 1. A diversidade de nossas aplicações alvo. LoC significa Linhas de Código. PR significa Pull Requests. A idade é apresentada em anos.

Projetos	Lançado em	Migrou em	LoC	Contribuidores	Contribuições	PR	Idade
`jenkins`	Nov. 2006	Nov. 2010	191K	556	21K	2K	10
`ruby`	Jan. 1998	Feb. 2010	1,001K	95	40K	1K	18
`mongodb`	Oct. 2007	Jan. 2009	2,104K	324	31K	1K	9

3.2. Coleta e Análise dos Repositórios

Para cada um dos três projetos, foram extraídos dados dos repositórios de código utilizando técnicas de mineração de logs. Entre os dados coletados estão: o número de novatos que ingressaram em cada projeto, o número de contribuições, e o número de contribuidores ativos. Definimos um contribuidor ativo como aquele que teve sucesso ao realizar ao menos uma contribuição ao repositório, sem distinções de tipo de arquivo alterado, seja ela uma contribuição em termos de código, documentação ou tradução. As contribuições são definidas como qualquer alteração de arquivos do projeto, descrita pela

documentação do GitHub como um *commit* ou *pull-request*. Já os novatos são definidos de forma similar aos contribuidores ativos, identificados a partir da data em que fizeram sua primeira contribuição. Por exemplo, um contribuidor é considerado novato na exata data em que o mesmo realizou sua primeira contribuição ao projeto, contribuições posteriores serão descartadas para essa métrica.

Após realizarmos a coleta dos dados, comparamos as distribuições de cada métrica coletada antes e após a migração de cada um dos projetos. Por exemplo, nós comparamos o número de novos contribuidores de um dado projeto antes e após a migração para o ambiente social de codificação. Sendo que, deste modo, nos tornamos capazes de verificar se a migração pode impactar na colaboração de projetos. Para uma melhor visualização do que foi coletado, criamos gráficos apresentando em três informações distintas: número de novatos, de contribuidores e contribuições.

Para avaliar nossas comparações, utilizamos os testes estatísticos não-paramétricos Mann-Whitney-Wilcoxon (MWW) e o delta de Cliff (tamanho do efeito). Os testes foram escolhidos porque as métricas coletadas antes e após a migração não seguiam uma distribuição normal. O teste MWW foi usado para verificar se duas distribuições de métricas são diferentes para um $\alpha = 0.05$. O delta de Cliff foi utilizado para dimensionar o tamanho da diferença das medidas coletadas antes e depois da migração. Quanto maior o valor do delta de Cliff obtido, maior é a diferença entre as distribuições. Para interpretar os resultados do tamanho da diferença foi utilizado a escala provida por [Romano et al. 2006]: *delta* < 0.147 (diferença insignificante), *delta* < 0.33 (diferença baixa), *delta* < 0.474 (diferença média), *delta* $>= 0.474$ (diferença alta).

4. Resultados

A Figura 1 apresenta uma visão geral dos dados coletados. Estes gráficos trazem uma perspectiva temporal de diferentes características pesquisadas. Por exemplo, a linha pontilhada em verde representa o número de contribuidores novatos que tiveram sucesso ao realizar no mínimo uma contribuição. A linha tracejada em azul representa o número de contribuições realizadas em todo o período de execução do projeto. A linha em vermelho representa o número de contribuidores ativos. E finalmente na vertical, a linha tracejada em preto indica a exata data em que cada projeto migrou para o GitHub. Além disto, trazemos através da Tabela 2, todos os resultados estatísticos obtidos nesta pesquisa (p-valor e de *effect size*).

Tabela 2. Resultados estatísticos. Células em verde indicam um alto tamanho de efeito, enquanto células em amarelo indicam um tamanho de efeito médio.

Projetos	Novatos		Contribuidores		Contribuições	
	p-value	$delta$	p-value	$delta$	p-value	$delta$
`jenkins`	0.001	0.131	2.66^{-06}	0.147	0.138	-0.031
`ruby`	0.489	-0.015	5.16^{-07}	0.106	2.20^{-16}	0.478
`mongodb`	0.178	0.143	1.41^{-07}	0.403	2.20^{-16}	0.710

Percebeu-se um aumento no número de contribuições. Em dois dos três projetos escolhidos foi possível notar um aumento significativo no número de contribuições. O que pode indicar que a migração de fato está relacionada a este crescimento. Estes resultados

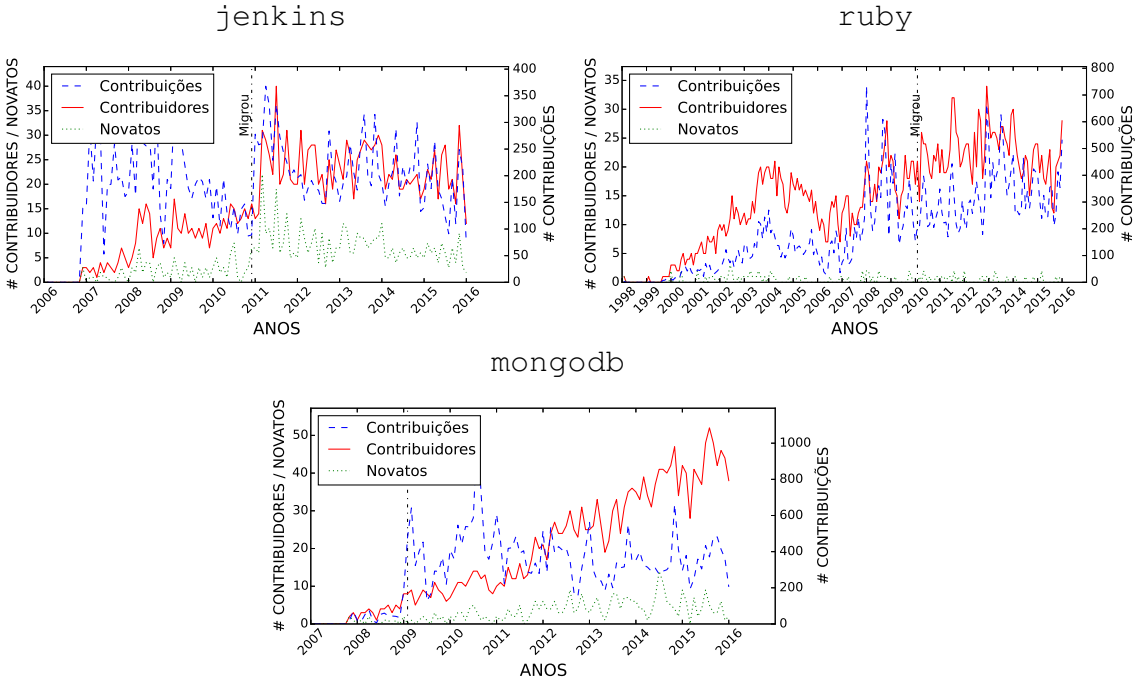

Figura 1. Gráfico de contribuições (linha azul tracejada), contribuidores (linha vermelha) e novatos (linha verde pontilhada), divididos em períodos antes e após a migração (linha tracejada em preto).

foram obtidos pelos projetos `mongodb` (p-valor = 2.20^{-16}, $delta$ = 0.710) e `ruby` (p-valor = 2.20^{-16}, $delta$ = 0.478). Estes resultados indicam que, ao comparar o cenário anterior e posterior a migração, existe uma relevante diferença em número de contribuições após a migração. Uma das hipoteses é que este crescimento esteja relacionado a aspectos sociais presentes no GitHub, que motivem de certa forma os desenvolvedores a contribuir. Entrentanto, não é possível generalizar essa conclusão, uma vez que o projeto `jenkins` apresentou um *effect size* insignificante ($delta$ = -0.031).

Existe uma maior frequência de contribuidores. Ademais, é observado que os gráficos do `jenkins` e do `mongodb` mostram um aparente crescimento em número de contribuidores ativos por período. Quanto mais contribuidores ativos em um projeto, possivelmente mais contribuições existirão. Entretanto, o efeito em termos estatísticos só foi notório no `mongodb` (p-$value$ $= 1.41^{-07}$, $delta$ $= 0.403$), que obteve um efeito considerado médio.

5. Ameaças à validade

Pode-se argumentar que são poucos os projetos de código aberto por nós analisados, o que portanto, limita a generalização de nossos resultados. Entretanto, os projetos de código aberto selecionados são de diferentes domínios, tamanhos e idades. Além disso, a intenção deste trabalho é explorar de forma preliminar o fenômeno.' Para trabalhos futuros pretende-se aumentar o tamanho da amostra analisada.

Outra ameaça para validade de nossa pesquisa está na forma como selecionamos os autores de *commits*. Visto que alguns dos projetos selecionados migraram de ambientes com outros sistemas de controle de versão, tais como `SVN`, que não distinguem autores de

committers. Neste caso, nós usamos o endereço de *email* para diferencia-los. No entanto, em repositórios `SVN` não existe a necessidade de se informar um *email* ao realizar uma contribuição, bem como um contribuidor pode usar diferentes *emails* ao contribuir. Estes fatores tem o potencial de criar falsos-positivos, onde por exemplo, um contribuidor pode ser contado mais de uma vez. Para suavizar esta ameaça, utilizamos técnicas de remoção de ambiguidade, além de testes estatísticos visando mitigar ameaças de generalização, de acordo com nossas hipóteses.

6. Conclusão

Ambientes sociais de codificação estão mudando o modo como *software* é construído. Estes ambientes possuem uma diversidade de aspectos que fazem com que as contribuições se tornem muito mais visíveis. Com toda sua popularidade e melhorias, estes ambientes aparentam ser responsáveis pela solução de diversas barreiras enfrentadas por projetos de código aberto. Neste artigo, nós estudamos se e como estas melhorias realmente podem ser creditadas a tais ambientes.

Em resposta à nossa questão de pesquisa (*Quanto o processo de migração para ambientes sociais de codificação impacta na colaboração em projetos de software livre em termos de novos contribuintes e número de contribuições?*), podemos dizer que foi possível observar que o crescimento da quantidade de contribuições e contribuidores após a migração para o GitHub não é algo verdadeiro para todos os casos. Não pudemos evidenciar aumento nas contribuições ou nos contribuintes em um dos projetos avaliados (`ruby`). Portanto, existem indícios de que a crença de que o próprio GitHub será eficaz na captação de novos contribuintes para projetos de OSS não é de todo verdadeiro. Encontramos, entretanto, em dois casos foi possível evidenciar aumento na quantidade de contribuição após a migração para o GitHub, mas o crescimento em número de novatos não é algo garantido. Acreditamos que existam fatores não relacionados à migração que influenciam o aumento na quantidade de contribuidores e contribuições (por exemplo, receptividade da comunidade, complexidade do projeto, interesse da comunidade).

Para o futuro, nós planejamos expandir o escopo deste estudo conduzindo um estudo em mais larga escala e conduzir análise qualitativas que possam indicar as razões para o crescimento e quais fatores podem influenciar o crescimento das contribuições e contribuidores.

References

[Bird 2011] Bird, C. (2011). Sociotechnical coordination and collaboration in open source software. In *ICSM*, pages 568–573, Washington, DC, USA. IEEE Computer Society.

[Dabbish et al. 2012] Dabbish, L., Stuart, C., Tsay, J., and Herbsleb, J. (2012). Social coding in github: Transparency and collaboration in an open software repository. In *CSCW*, pages 1277–1286, New York, NY, USA. ACM.

[Ducheneaut 2005] Ducheneaut, N. (2005). Socialization in an open source software community: A socio-technical analysis. *CSCW*, 14(4):323–368.

[Fogel 2013] Fogel, K. (2013). *Producing Open Source Software: How to Run a Successful Free Software Project*. O'Reilly Media, first edition.

[Gousios et al. 2014] Gousios, G., Pinzger, M., and Deursen, A. v. (2014). An exploratory study of the pull-based software development model. In *ICSE*, pages 345–355.

[Jensen et al. 2011] 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*, HICSS '10, pages 1–10. IEEE.

[Marlow et al. 2013] Marlow, J., Dabbish, L., and Herbsleb, J. (2013). Impression formation in online peer production: Activity traces and personal profiles in github. In *CSCW*, pages 117–128.

[McDonald and Goggins 2013] McDonald, N. and Goggins, S. (2013). Performance and participation in open source software on github. In *CHI*, pages 139–144.

[Moura et al. 2015] Moura, I., Pinto, G., Ebert, F., and Castor, F. (2015). Mining energy-aware commits. In *MSR*, pages 56–67.

[Pham et al. 2013a] 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 *ICSE*, pages 112–121.

[Pham et al. 2013b] Pham, R., Singer, L., and Schneider, K. (2013b). Building test suites in social coding sites by leveraging drive-by commits. In *ICSE*, pages 1209–1212.

[Pinto et al. 2016] Pinto, G., Steinmacher, I., and Gerosa, M. (2016). More common than you think: An in-depth study of casual contributors. In *SANER*, pages 112–123.

[Romano et al. 2006] Romano, J., Kromrey, J. D., Coraggio, J., and Skowronek, J. (2006). Should we really be using t-test and cohen's d for evaluating group differences on the nsse and other surveys? In *Annual meeting of the Florida Association of Institutional Research*.

[Stallman 1999] Stallman, R. (1999). The gnu operating system and the free software movement.

[Steinmacher et al. 2015] Steinmacher, I., Conte, T., Gerosa, M. A., and Redmiles, D. F. (2015). 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*, CSCW '15, pages 1–13, New York, NY, USA. ACM.

[Thung et al. 2013] Thung, F., Bissyande, T. F., Lo, D., and Jiang, L. (2013). Network structure of social coding in github. In *Software maintenance and reengineering (csmr), 2013 17th european conference on*, pages 323–326. IEEE.

[Tsay et al. 2014] Tsay, J., Dabbish, L., and Herbsleb, J. (2014). Influence of social and technical factors for evaluating contribution in github. In *ICSE*, pages 356–366.

[Vasilescu et al. 2015] Vasilescu, B., Filkov, V., and Serebrenik, A. (2015). Perceptions of diversity on github: A user survey. In *CHASE*.

[Zhou and Mockus 2015] 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.