Students' Engagement in Open Source Projects: An Analysis of Google Summer of Code

Jefferson O. Silva
PUC-SP, Brazil
silvajo@pucsp.br

Igor S. Wiese
DACOM, UTFPR, Brazil
igor@utfpr.edu.br

Igor Steinmacher
DACOM, UTFPR, Brazil
SICCS, NAU, USA
igorfs@utfpr.edu.br

Marco A. Gerosa
IME, USP, Brazil
SICCS, NAU, USA
marco.gerosa@nau.edu

ABSTRACT

Several open source software (OSS) communities promote and participate in initiatives such as "summer of code" programs to foster contributions and attract new developers. However, little is known about how successful these initiatives are. As a case study, we analyzed Google Summer of Code (GSoC), which is a three-month program to foster students' participation in OSS projects. We found that 82% of the studied OSS projects merged students' at least one commit in codebase. When only newcomers are considered, ~54% of OSS projects merged at least one commit. We also found that ~23% of newcomers started contributing to GSoC projects before knowing they would be accepted. In addition, we did not find statistical difference between newcomers and students with prior participation in the projects regarding retention time after GSoC, with the exception of 2015 edition. Using survival analysis, we found that ~40% of students kept contributing longer than a month, while ~15% of them contributed longer than a year. OSS communities can take advantage of our results to balance the trade-offs involved in entering in this kind of initiative and to set expectations about how much contribution to expect and for how long students will engage.

CCS CONCEPTS

• Software and its engineering~Open source model

PALAVRAS-CHAVES

Google Summer of Code, Software Livre, Novatos, Retenção; Mineração de Repositórios de Software.

ACM Reference format:

J. O. Silva, I. S. Wiese, I. F. Steinmacher. M. A. Gerosa. 2017. Students' Engagement in Open Source Projects: An Analysis of Google Summer of Code. In Proceedings of XXXI Brazilian Symposium on Software Engineering, Fortaleza, Ceará, Brazil, September 2017. *(SBES'17)*, 10 pages.
DOI: 10.1145/123 4

SBES'17, September 2017, Fortaleza, Ceará BRAZIL

DOI: 10.1145/123_4

1 INTRODUÇÃO

A sustentabilidade de diversas comunidades de Software Livre (SL) depende da participação de novos desenvolvedores [1]. Os novatos são necessários não apenas por serem fonte de novas ideias [2], mas também para realizar importantes tarefas de rotina [3]. Entretanto, muitas comunidades enfrentam dificuldades tanto para atrair novatos que possam substituir os desenvolvedores que param de contribuir quanto para reter seus membros, o que faz com que muitas delas fracassem [4].Para atrair contribuições e voluntários, comunidades de SL participam de eventos de engajamento, que são acontecimentos de curta duração que incluem entre seus objetivos engajar novas pessoas no desenvolvimento de software. Esses eventos incluem programas que promovem o desenvolvimento de software por estudantes durante o período de férias (em inglês, esses eventos são conhecidos como Summers of Code) [5], como por exemplo, o Google Summer of Code (GSoC[2]), Rails Girls Summer of Code[3], Julia Summer of Code[4] e Outreachy[5]. Entre as variações de Summers of Code estão os Semesters of Code, que incluem o Facebook Open Academy[6] e o Undergraduate Capstone Open Source Projects[7]. Enquanto os Summers of Code ocorrem durante as férias e muitas vezes incluem remuneração financeira e mentores, os Semesters of Code acontecem em paralelo aos estudos regulares com a obtenção de créditos acadêmicos [6].

Alguns eventos são promovidos por empresas globalmente conhecidas, tais como Facebook, Yahoo! e Google, que são potencialmente atraentes aos estudantes [7], [8]. Embora a participação em eventos de engajamento nas comunidades proporcione recompensas atrativas, tais como valorização do currículo, pagamento e aprendizado, pouco se sabe o quanto a participação influencia na retenção de estudantes como futuros colaboradores, ou ainda o quanto da contribuição é de fato integrada ao repositório.

A literatura atual sobre eventos de engajamento em SL fornece evidência de retenção e contribuição de código para poucos projetos científicos [5], [9]–[11], tendo suas descobertas em grande parte baseadas apenas na percepção de estudantes e mentores. Uma exceção é o trabalho de Schilling *et al.* [12], que mineraram repositórios para quantificar a retenção de estudantes, mas se limitaram ao projeto KDE. Assim, não apenas pouco se sabe sobre

[2] https://developers.google.com/open-source/gsoc/
[3] http://railsgirlssummerofcode.org/
[4] http://julialang.org/blog/2015/05/jsoc-cfp/
[5] https://wiki.gnome.org/Outreachy
[6] https://www.facebook.com/pg/OpenAcademyProgram/about
[7] http://ucosp.ca/

o quanto os eventos de engajamento em SL promovem contribuições em um contexto mais amplo, mas também não existe, até onde sabemos, nenhum estudo empírico que investigue quantitativamente a retenção e contribuição de código de estudantes para mais de um projeto de SL.

Focamos no GSoC porque é um programa bem estabelecido, que tem atraído estudantes de todo o mundo para uma grande variedade de projetos há mais de 10 anos, bem como tem chamado atenção da comunidade científica [5], [9]–[11]. No intuito de entender a dimensão das contribuições (*i.e.*, *churn* de código e commits) que são aceitas no repositório, e os níveis de retenção e analisar a "sobrevivência" dos estudantes do GSoC em projetos de SL, investigamos as questões de pesquisa (QP) apresentadas a seguir.

QP1. Quanto código os participantes do GSoC contribuem?

Motivação: A resposta a essa pergunta visa auxiliar as comunidades de SL a ajustar suas expectativas com relação à quantidade de código que é de fato aceita no repositório.

Abordagem: Focamos na quantidade de commits e *churn* de código para estudar as contribuições. Dividimos os commits feitos aos projetos em três períodos de participação: *antes*, *durante* e *após* o GSoC. Em cada período, contamos quantos commits de autoria dos estudantes foram aceitos no repositório. Avaliamos quanto código o estudante adicionou calculando o *churn* de código (*i.e.*, linhas adicionadas + linhas removidas) em cada commit.

Descobertas: Há commits aceitos no repositório nos três períodos. A maioria dos projetos de SL (~82%) aceitou pelo menos um commit de autoria dos estudantes. Quando apenas novatos são considerados, ~54% dos projetos de SL aceitou pelo menos um commit.

QP2. Os estudantes com participação prévia nos projetos permanecem contribuindo por mais tempo que os estudantes novatos?

Motivação: Tendo em vista que os objetivos dos programas de engajamento têm relação com atração e retenção de novos desenvolvedores, a resposta a esta pesquisa visa avaliar a que ponto tal objetivo é alcançado. Além disso, pode-se apoiar os projetos de SL a gerenciar expectativas em termos de retenção de desenvolvedores.

Abordagem: Classificamos os estudantes em novatos e estudantes com participação prévia nos projetos GSoC utilizando um *limiar*. Estimamos a participação *após* o GSoC estudando: o intervalo entre a data de término do GSoC e o último commit do estudante ao projeto; suas contribuições; e a contagem de dias em que o estudante realizou commits. A participação *antes* do GSoC foi estimada analogamente, mas considerando o intervalo entre o primeiro commit ao projeto e a data de início do GSoC. Com estes fatores, aplicamos a técnica de análise de sobrevivência para responder à questão de pesquisa.

Descobertas: 23,1% dos novatos que começaram a participar antes do GSoC o fizeram sem saber se seriam aceitos. ~40% dos estudantes continuaram contribuindo por mais de um mês após o GSoC, ~25% por mais de seis meses, e ~15% por mais de um ano. Nossos resultados sugerem que embora muitos estudantes tenham longos intervalos de retenção, em muitos casos, a frequência da contribuição não foi proporcional aos intervalos.

2 FUNDAMENTAÇÃO E TRABALHOS RELACIONADOS

Nesta seção, apresentamos trabalhos relacionados à retenção de novatos em eventos de engajamento nas comunidades de SL. Começamos explicando o que é o Google Summer of Code, como funciona e o porquê escolhemos estudá-lo.

2.1 Google Summer of Code

O Google Summer of Code (GSoC) é um programa global da Google que remunera financeiramente os estudantes para escrevem código para projetos de SL, com duração de três meses. Escolhemos estudar o GSoC porque: é o mais conhecido entre os Summers of Code, e provê aos seus estudantes um amplo leque de recompensas de participação [5], incluindo participar do programa de uma empresa global, laços comunitários, desenvolvimento de habilidades; satisfação pessoal; avanço professional; reconhecimento entre os pares; status; e remuneração financeira.

Desde seu lançamento em 2005, o GSoC remunera[8] os estudantes que completarem com sucesso as três fases do programa. O GSoC possui 5 metas[9]. As metas (*ii*) e (*iii*) inspiram nossas QPs: (*ii*) "inspirar jovens desenvolvedores a participar do desenvolvimento de SL" (*iii*) "ajudar projetos SL a identificar e trazer novos desenvolvedores". As outras metas são: (*i*) "criar e publicar código para o benefício de todos", (*iv*) "prover aos estudantes a oportunidade de realizar trabalhos relacionados a seus interesses acadêmicos" e (*v*) "expor o estudante a cenários de desenvolvimento de software de mundo real" (tradução livre).

Candidatos precisam escrever e submeter propostas de projetos para as organizações de SL (previamente aprovadas pela Google), tais como Apache Software Foundation e Debian. Os mentores representantes das organizações – que usualmente são colaboradores regulares – classificam e decidem quais propostas aceitar. Quando os estudantes iniciam efetivamente o desenvolvimento, a Google realiza um pagamento inicial. Após a primeira metade do programa, os mentores avaliam o trabalho e a Google realiza os pagamentos intermediários para os aprovados. Ao final do GSoC, os estudantes submetem seu código e os aprovados recebem o restante do pagamento e são convidados a uma reunião na Califórnia.

2.2 A Retenção de Novatos em SL

Usualmente, os estudos sobre retenção em SL se baseiam na perspectiva do desenvolvedor individual. Logo, motivação intrínseca [13]–[15], entrosamento com membros da equipe [16]–[18], características do projeto [19]–[21], ideologia [22] e incentivos e recompensas [7], [8], [23] têm sido reportados como relevantes para a contribuição contínua.

Zhou e Mockus [24], por exemplo, pesquisaram sobre como identificar novatos com maiores chances de permanecer contribuindo para o projeto, no intuito de oferecer suporte ativo para que

[8] Do GSoC 2013-2015, Google pagou uma quantia de US$ 5.500 aos estudantes
[9] No momento da escrita, o GSoC havia removido a página com as metas. Entretanto, elas podem ser encontradas em sites que suportam as comunidades, ex: http://write.flossmanuals.net/gsocstudentguide/what-is-google-summer-of-code

eles se tornem colaboradores de longo prazo. Os pesquisadores descobriram que a vontade individual e o clima do projeto estão associados com as chances de um indivíduo se tornar colaborador de longo prazo. Por outro lado, Fang e Neufeld [2] usaram a teoria da Participação Periférica Legitimada para entender a motivação dos desenvolvedores para continuarem contribuindo. Os autores descobriram que as condições iniciais de participação não foram efetivas para prever a participação no longo prazo, embora a aprendizagem situada e a construção da identidade estivessem positivamente ligados à continuidade da participação.

2.3 Eventos de Engajamento

Os eventos de engajamento em comunidades de SL estão se tornando comuns. Apesar da relevância prática, poucos trabalhos examinaram como a participação nestes eventos influencia as contribuições voluntárias ou ainda quanto do código escrito é de fato aceito nos repositórios dos projetos. Baseando-se na perspectiva das comunidades de SL, Schilling *et al.* [12] usaram os conceitos de *Person-Job* (a congruência entre o desejo do candidato e as ofertas da profissão) e *Person-Team* (a compatibilidade interpessoal com a equipe existente) para prever a retenção de ex-participantes do GSoC no projeto KDE. Os pesquisadores encontraram que frequências intermediárias de commits (4-94 commits) e altas (>94 commits) estavam fortemente correlacionadas à retenção.

Da mesma forma, Trainer *et al.* [10] analisaram o projeto BioPython, com o objetivo de investigar os resultados do GSoC nesse projeto. Utilizando entrevistas, os pesquisadores listaram três resultados positivos; (*i*) *a adição de novas funcionalidades ao repositório*; (*ii*) *treinamento* – estudantes aprenderam novas habilidades; e (*iii*) *desenvolvimento pessoal* – os estudantes usaram a participação no GSoC para o avanço de suas carreiras. Além disso, os autores também relataram que os mentores enfrentaram muitos desafios relacionados aos processos do GSoC, tais como dificuldades para classificar a grande quantidade de propostas enviadas às comunidades.

Baseando-se na perspectiva dos organizadores dos eventos, Trainer *et al.* [5] conduziram um estudo de caso com 22 projetos GSoC do domínio científico para entender a abrangência dos resultados do programa. Eles relatam que o GSoC facilitou a criação de laços entre mentores e estudantes e que 18% dos estudantes (n=22) se tornaram mentores em edições posteriores. Embora tenham ajudado a esclarecer aspectos dos eventos de engajamento, esses estudos se limitam à investigação de poucos projetos, primordialmente no domínio científico. Apenas Schilling *et al.* [12] utilizaram dados minerados de repositórios de software para quantificar a retenção dos estudantes, embora tenham limitado a análise ao projeto KDE. Nosso estudo realiza uma investigação mais abrangente, ao analisar dados obtidos de múltiplos repositórios de projetos de variados domínios.

3 MÉTODO DE PESQUISA

Nesta seção, apresentamos o método utilizado na coleta e análise dos dados. Para a coleta, pesquisamos pelos projetos dos estudantes, minerando repositórios de software. Para análise, usamos estatística descritiva e testes estatísticos.

3.1 Coleta dos Dados

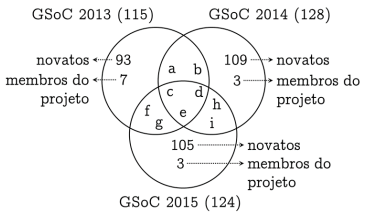
A coleta de dados envolveu diversas etapas, uma vez que a Google publica apenas os nomes das organizações (ex. Apache Software Foundation) e dos candidatos aceitos, sem indicação do projeto específico. Realizamos buscas manuais para encontrar o projeto de cada estudante. Portanto, limitamos nossa análise às edições de 2013 a 2015, que totalizam 3.107 estudantes distintos.

Aleatoriamente, selecionamos uma amostra de 866 estudantes. Manualmente, pesquisamos pelos projetos GSoC dos estudantes nos sistemas de controle de versões (SCV) e na web, usando os nomes dos estudantes e as descrições dos projetos fornecidos pelo GSoC. Julgamos que encontramos os projetos quando havia evidência clara ligando os projetos às informações dos estudantes (*ex.*, quando as descrições dos projetos nos SCVs correspondiam às dos projetos GSoC ou quando as descrições de projetos nos SCVs correspondiam àquelas dos projetos GSoC, ou ainda quando encontramos links web nos blogs dos estudantes criados para os projetos). Encontramos os projetos de 406 estudantes (de 866), todos hospedados no GitHub.

O próximo passo foi identificar os IDs dos estudantes nos registros dos projetos. Em primeiro lugar, usamos a ferramenta *MetricsGrimoire-CVSAnalY*[10] para extrair os dados de repositórios Git e armazená-los em um banco de dados local. Em segundo lugar, pesquisamos por todos os IDs que os estudantes possam ter usado. Para identificar os estudantes, aplicamos heurísticas comuns de desambiguação, tais como aquelas apresentadas por Wiese *et al.* [25]: por exemplo, quando os IDs eram compostos da combinação das iniciais do primeiro nome do estudante com seu sobrenome, ou quando eram compostos pelas iniciais dos estudantes e estas iniciais eram usadas como seus IDs no GitHub. Como resultado, identificamos os IDs de 367 estudantes (de 406) nos projetos, que representa nossa amostra final de estudantes. Essa amostra nos oferece um nível de confiança de 95% e uma margem de erro de 5%. Adicionalmente, verificamos se cada estudante da amostra participou em edições do GSoC anteriores.

Na Figura 1, podemos observar que parte dos estudantes de nossa amostra participaram em mais de uma edição do GSoC: 32 estudantes participaram em 2 edições (8,7%), 15 participaram em 3 edições (4,1%) e os outros 320 (87,2%) participaram em uma das três edições analisadas. Além disso, encontramos 16 estudantes que participaram em edições do GSoC anteriores a 2013. Finalmente, para todos estudantes em nossa amostra, contamos o número de participações como estudante e como mentor usando a lista publicada pelo GSoC, considerando as edições de 2005 (primeira edição) a 2015. Usamos o nome do estudante e do projeto como critério de busca. Se o nome do estudante aparecia como estudante e mentor, analisamos se o projeto do mentor era relacionado ao projeto do estudante e se o ano como estudante era anterior ao ano em que foi mentor.

[10] http://metricsgrimoire.github.io/CVSAnalY



a = 5 estudantes (da amostra de 2013) também participaram do GSoC 2014
b = 5 estudantes (da amostra de 2014) também participaram do GSoC 2013
c = 5 estudantes (da amostra de 2013) também participaram dos GSoC 2014 e 2015
d = 5 estudantes (da amostra de 2014) também participaram dos GSoC 2013 e 2015
e = 5 estudantes (da amostra de 2015) também participaram dos GSoC 2014 e 2015
f = 5 estudantes (da amostra de 2013) também participaram do GSoC 2015
g = 5 estudantes (da amostra de 2015) também participaram do GSoC 2013
h = 6 estudantes (da amostra de 2014) também participaram do GSoC 2015
i = 6 estudantes (da amostra de 2015) também participaram do GSoC 2014

Figura 1. Número de estudantes por ano de participação

3.2 Análise dos Dados

Nesta seção, apresentamos como analisamos os dados coletados por questão de pesquisa. Para atingir nosso objetivo, dividimos a participação dos estudantes em três períodos: *antes*, *durante* e *após* o GSoC. Utilizamos o calendário oficial (*i.e.*, datas de início e fim) para classificar os commits em cada período. Embora os estudantes possam participar de uma comunidade de SL de diversas maneiras, utilizamos o termo **participação dos estudantes** para se referir a seus commits (e *churn* de código). O **intervalo de participação** dos estudantes se refere ao tempo em dias que um estudante contribuiu (*i.e.*, fez commits). Por exemplo, se uma edição do GSoC começou no dia 15 e um commit foi feito no dia 10 de um mesmo mês e ano, então o intervalo dessa contribuição é de 5 dias antes do GSoC.

Além disso, foi necessário diferenciar os **estudantes novatos** dos **estudantes com participação prévia** em seus projetos GSoC. Identificamos dois tipos de estudantes com participação prévia: os que participaram em edições anteriores do GSoC e os que já participavam como membros do projeto de SL. Para identificar os estudantes com participação em edições anteriores do GSoC, contamos o número de edições que os estudantes participaram *antes* da edição considerada em nossa amostra (até a primeira edição de 2005).

Para distinguir membros do projeto dos novatos que começaram a participar recentemente, utilizamos a data de anúncio das organizações de SL aceitas no GSoC como *limiar*. Assim, se um estudante começou a contribuir após esse limiar, classificamos o estudante como novato. Do contrário, classificamos o estudante como membro do projeto. Para o GSoC 2013, o anúncio das organizações de SL aceitas foi feito 70 dias antes do início oficial. Para o GSoC 2014 e 2015, o anúncio foi feito com 84 dias de antecedência.

QP1. Quanto código os participantes do GSoC contribuem?

Para determinar quanto código os estudantes adicionaram, utilizamos a ferramenta *git-log*, que cria um arquivo de log para os projetos clonados do git, contendo o *Secure Hash Algorithm* (SHA), nome do autor e quantidade de linhas adicionadas e removidas por arquivo contido no commit. Em seguida, calculamos o *churn* por commit. Para avaliar se um determinado commit foi aceito, comparamos os SHA dos commits dos estudantes com os SHAs dos commits do repositório.

QP2. Os estudantes com participação prévia nos projetos permanecem contribuindo por mais tempo que os novatos?

No domínio médico, as curvas de sobrevivência descrevem a probabilidade que um sujeito possa viver além de um determinado período: o eixo x representa a duração da sobrevivência (i.e., quanto um indivíduo sobrevive), ao passo que o eixo y representa a probabilidade de um indivíduo sobreviver [26]. Idealmente, para determinar curvas de sobrevivência, deve-se ter todos os dados a respeito da morte de todos os indivíduos da população estudada. Contudo, em muitos casos, os indivíduos ainda estão vivos quando o estudo termina, o que impede os pesquisadores de saberem o tempo real de suas mortes. Esta lacuna de informação sobre os dados é conhecida como *dados censurados à direita*. Por esta razão, Kaplan e Meier [27] propuseram as 'curvas Kaplan-Meier'. Além disso, normalmente utilizam-se essas curvas para comparar grupos ao invés de indivíduos, como por exemplo, para comparar o comportamento de um grupo de controle e outro experimental. Quando aplicados a grupos, cada grupo é representado por uma curva.

As curvas Kaplan-Meier foram utilizadas pela primeira vez no contexto da medicina e então aplicada a outros domínios incluindo a engenharia de software. Por exemplo, Bird et al. [28] utilizou a análise de sobrevivência para estimar a imigração de eventos no Postgres. Samoladas et al. [29] aplicou técnicas de análise de sobrevivência para obter estimativas de desenvolvimento futuro de projetos de SL. Goeminne et al. [30] analisou se diferentes *frameworks* de bancos de dados 'co-ocorrem' em projetos de SL e se alguns *frameworks* bancos de dados são substituídos por outros com o passar do tempo. Neste trabalho de pesquisa, utilizamos a análise de sobrevivência para entender se os estudantes com participação prévia contribuem por mais tempo que os estudantes novatos.

Analisamos os intervalos de participação dos estudantes *antes* e *após* o GSoC utilizando *violin plots* gerados no RStudio. Para enriquecer a análise, investigamos a relação entre os intervalos de participação e a contagem de dias de contribuição, pois longos intervalos de participação podem potencialmente não se traduzir em muitas contribuições.

Além disso, utilizamos análise de sobrevivência para entender quando os novatos e os estudantes com participação prévia param de contribuir após o término do programa. Selecionamos 365 dias como limiar de censura à direita. Baseamos a escolha desse limiar no trabalho de Bin, Robles e Serebrenik [31].

Entretanto, como outros limiares podem ser encontrados na literatura (ex., [32], [33], também analisamos até que ponto a alteração do limiar de censura à direita (*i.e.*, 365 dias) influencia nossos resultados. Assim, repetimos o estudo com limiares de censura à direita diferentes: 30 dias, 90 dias e 180 dias. Em seguida, utilizamos o teste estatístico Log-Rank para analisar se as curvas de sobrevivência dos novatos e dos estudantes com participação são estatisticamente equivalentes.

Tabela I. Caracterização da Amostra

	# de estudantes que participaram como estudantes	# de estudantes que participaram como mentores	$\bar{x}$ de participação - *em dias* - após o GSoC (σ)	$\bar{x}$ de participação - *em dias* - antes do GSoC (σ)	$\bar{x}$ do # de commits ao projeto GSoC (σ)	$\bar{x}$ do # de commits aceitos no repositório
1	307	9	52,0 (135)	36,3 (117)	97,0 (136)	64,3 (92)
2	48	3	77,5 (203)	108,6 (311)	216,9 (489)	115,5 (252)
3	8	0	74,4 (157)	37,2 (99)	115,5 (153)	89,1 (160)
4	3	0	3,5 (305)	0,0 (0)	155,0 (174)	20,0 (17)
5	0	0	0,0 (NA)	0,0 (NA)	0,0 (NA)	0,0 (NA)
6	1	0	476,0 (NA)	1.603,0 (NA)	477,0 (NA)	476,0 (NA)

Assumimos como hipótese nula que as curvas de sobrevivência de novatos e estudantes com participação prévia são estatisticamente equivalentes. Se os valores-p resultantes forem menores que o nível de confiança de 0,05, rejeitamos a hipótese nula.

4 RESULTADOS

A seguir, reportamos os resultados do estudo.

4.1 Análise por período

A Tabela I sumariza as características de nossa amostra em termos de número de participações no programa tanto como estudante quanto como mentores; participação *antes* e *após* o GSoC; o total de commits; e total de commits aceitos. Note que as linhas da tabela referentes a 3-6 participações podem incluir as edições do GSoC 2010 a 2015. Também é relevante mencionar que há poucos estudantes com 3+ participações (antes/após) e commits (total/aceitos). Além disso, os estudantes com participação em apenas uma edição do GSoC não são necessariamente novatos no projeto, e os que possuem 2+ participações não são necessariamente membros do projeto.

Na Figura 3, observamos também que em nossa amostra quase metade dos estudantes tiveram código aceito no repositório *após* o final do GSoC. Vários estudantes (~19%) tiveram código aceito somente *durante* o programa.

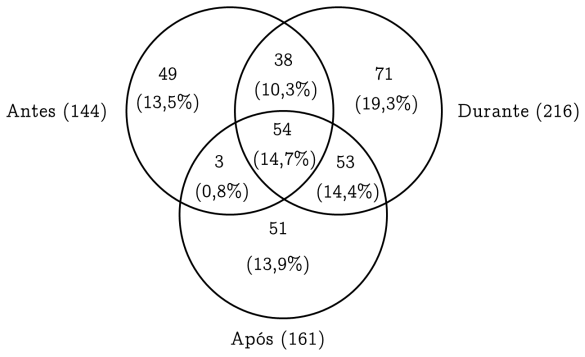


Figura 2. Contagem (%) de estudantes que tiveram commits aceitos no repositório por período de participação.

QP1. Quanto código os participantes do GSoC contribuem?

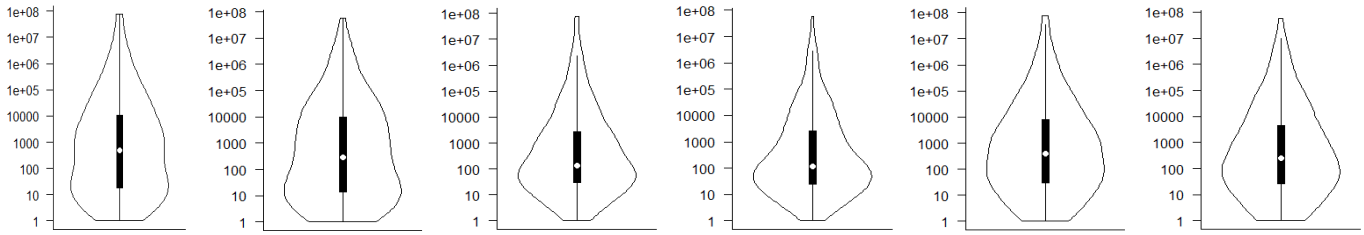
Comparando as Figura 3 (a) e (b), observamos que alguns commits dos estudantes foram aceitos antes do início oficial do GSoC. Esses commits podem ter vindo de pelo menos três fontes distintas: (i) estudantes que já colaboravam como o projeto; (ii) estudantes que participaram em edições anteriores do GSoC; 2 (iii) novatos. Uma possível explicação para a existência de commits de novatos é que alguns candidatos contribuem aos projetos GSoC para aumentar suas chances de serem aceitos no programa. De fato, encontramos alguns blogs de estudantes e mentores (*ex.*, [34]) com dicas de como serem aceitos.

A Figura 4 mostra o número de estudantes distintos que contribuíram para seus projetos GSoC nos 180 dias antes do início oficial. Enquanto os commits dos estudantes com participação prévia no projeto (Figura 4b) permaneceram relativamente constante até a data de divulgação da lista de estudantes aceitos (~30 dias antes do início oficial), alguns novatos (Figura 4a) começaram a participar após a divulgação da lista de organizações de SL aceitas. Isto significa que **alguns novatos (~23%) começaram a fazer commits ao projeto GSoC antes mesmo de saberem que seriam aceitos**, possivelmente tentando demonstrar suas habilidades para a comunidade antes da seleção

Na Figura 3 (c) e (d), observamos que as medianas das distribuições de commits e commits aceitos *durante* o GSoC são um pouco menores que as medianas referentes aos períodos de *antes* e *após* o programa, tipicamente entre 1 commit (Q_1) e 2,6M (Q_3), totalizando ~77MM de commits. Durante este período, os estudantes fizeram 224MM commits. Nos piores casos (~25%), os estudantes não tiveram commits aceitos no repositório, mesmo tendo feito 143MM commits. A Figura 3 (e) mostra a participação de ~55% de estudantes que fizeram commits após o GSoC.

Os valores típicos de contribuição dos estudantes participantes representados na Figura 3 (e) ficaram entre 31 (Q_1) e 8,5M (Q_3) commits após o programa. Na Figura 3 (f), vemos que ~44% dos estudantes tiveram commits aceitos, tipicamente entre 26 e ~4,5M. Logo, em todos os períodos, houve código aceito no repositório.

A maioria dos projetos de SL de nossa amostra se beneficiou da participação no GSoC, uma vez que em ~87% dos casos as comunidades tiveram pelo menos um commit aceito no repositório. Quando consideramos apenas novatos, em ~54% dos casos, as comunidades de SL aceitaram pelo menos um commit em seus repositórios.



(a) Distribuição dos commits **antes** GSoC [*189/367 (~51%) não fizeram commits*]

(b) Distribuição dos commits aceitos no repositório **antes** do GSoC [*223/367 (~61%) não tiveram commits aceitos*]

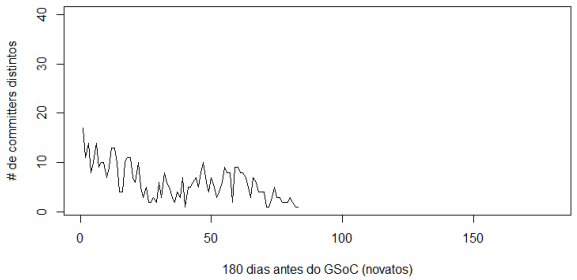
(c) Distribuição dos commits **durante** o GSoC [*99/367 (~27%) não fizeram commits*]

(d) Distribuição de commits aceitos no repositório **durante** GSoC [*151/367(~41%) não tiveram commits aceitos*]

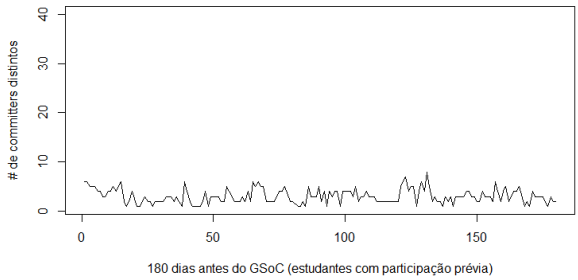
(e) Distribuição dos commits **após** o GSoC [*166/367 (~45%) não fizeram commits*]

(f) Distribuição dos commits aceitos no repositório **após** o GSoC. [*206/367 (~56%) não tiveram commits aceitos*]

Figura 3. Distribuição de commits e commits aceitos no repositório por período de participação (antes, durante e após).



(a) Participação dos estudantes novatos antes do GSoC (180 dias)



(b) Participação dos estudantes com participação prévia antes do GSoC (180 dias)

Figura 4. Participação dos estudantes 180 dias antes do GSoC

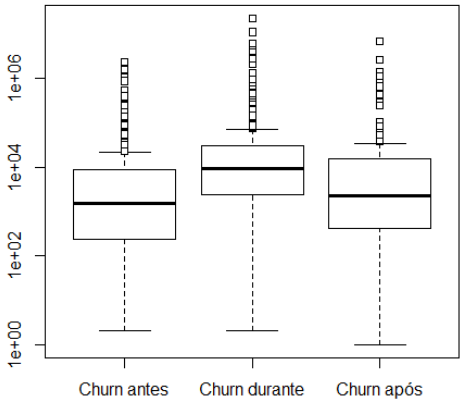


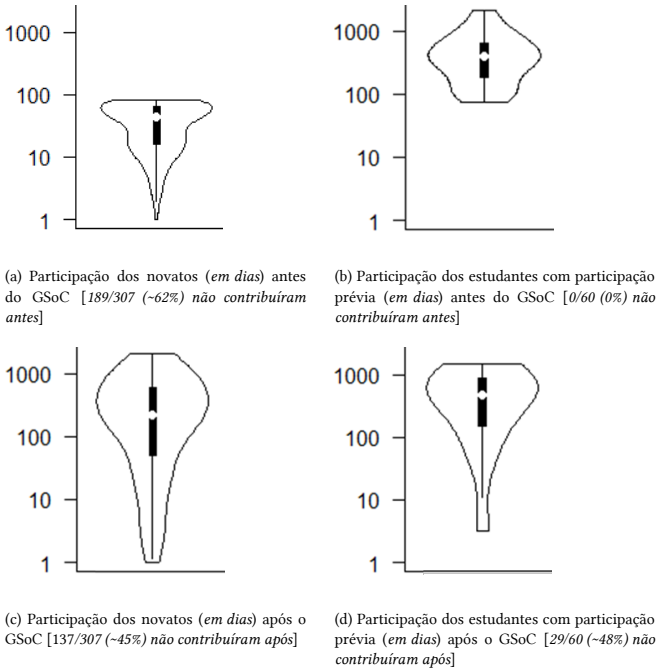
Figura 5. O churn de código dos estudantes por período de participação

Analisamos o *churn* de código para oferecer uma perspectiva adicional. A Figura 5 apresenta a distribuição de *churn* dos estudantes por período de participação. O *boxplot* '*churn* antes' mostra a mediana da distribuição (1,5M), sendo que os ~25% com maior *churn* ficaram entre ~8,9M (Q_3) e ~22M (limite superior). O *boxplot* '*churn* durante' mostra que mediana da distribuição aproximadamente foi aproximadamente seis vezes maior (~8,9M), com os ~25% com maior *churn* ficaram no intervalo entre 30M (Q_3) e 71M (limite superior).

O *boxplot* de '*churn* após' mostra que o *churn* decresce após o programa, com a mediana caindo para 2,4M. Entretanto, os 25% de maior *churn* permaneceram com o *churn* de código alto, no intervalo entre 16M (Q_3) e 33,7M (limite superior). Podemos entender a magnitude da contribuição dos estudantes quando adicionamos o *churn* de código às distribuições. **Desta forma, observamos que o *churn* de código *antes* do GSoC totalizou 11,5MM; *durante*, 81,9MM; e *após*, 19,1MM**.

QP2. Os estudantes com participação prévia nos projetos permanecem contribuindo por mais tempo que os estudantes novatos?

Para entender quanto tempo os estudantes participaram em seus projetos GSoC, a Figura 6 apresenta a distribuição dos intervalos da participação dos estudantes *antes* e *depois* do GSoC. Para facilitar a comparação, dividimos os estudantes em estudantes novatos e experientes. A Figura 6 (a) e (c) mostra a participação de novatos, em dias, *antes* e *depois* do GSoC, e a Figura 6 (b) e (d) de experientes.



(a) Participação dos novatos (*em dias*) antes do GSoC [*189/307 (~62%) não contribuíram antes*]

(b) Participação dos estudantes com participação prévia (*em dias*) antes do GSoC [*0/60 (0%) não contribuíram antes*]

(c) Participação dos novatos (*em dias*) após o GSoC [*137/307 (~45%) não contribuíram após*]

(d) Participação dos estudantes com participação prévia (*em dias*) após o GSoC [*29/60 (~48%) não contribuíram após*]

Figura 6. Distribuição da participação antes e após para os estudantes novatos e experientes (em dias)

Entretanto, na maioria dos casos os novatos entram para o programa sem ter participado dos projetos, sugerindo que o GSoC atrai potenciais futuros colaboradores. Na Figura 6 (b) observamos que muitos estudantes com participação prévia possuem longos intervalos de participação em seus projetos GSoC (Q_1=187; Q_3=639). Analisando mais profundamente, identificamos que esses casos são na maioria de estudantes com participação em edições do GSoC anteriores (26). Na Figura 6 (c), observamos que os novatos não continuaram fazendo commits após o GSoC (~64%). O longo intervalo de participação dos que continuaram se refere, em parte, a participações em edições futuras do programa, que é um tipo de retenção diferente, embora válida.

Assim, muitos projetos de SL se beneficiaram da participação após o término do GSoC.

Para entender a duração esperada dos estudantes no GSoC, investigamos a técnica de análise de sobrevivência. A Figura 7 mostra as curvas de sobrevivência para os novatos (linha azul) e estudantes com participação prévia (linha vermelha), indicando por meio de linhas pontilhadas os limiares de censura à direita para 30 dias (L1), 90 dias (L2), 180 dias (L3) e 365 dias (L4). A curva dos estudantes novatos possui inicialmente 307 amostras enquanto que a curva dos estudantes com participação prévia possui inicialmente 60 estudantes. Um dia após o término do período de codificação, 137 (~45%) novatos e 29 (49%) dos estudantes com participação prévia param de participar. Por esta razão, as probabilidades iniciais de sobrevivência de ambos os grupos de estudantes começam em torno de 50%.

Nas duas primeiras semanas após a data de término, a curva de sobrevivência dos novatos apresenta uma queda maior comparada a dos estudantes com participação prévia. A partir desse ponto, as duas curvas apresentam níveis de mortalidade semelhantes, com exceção do nível de participação no dia 300 da Figura 7. Analisando mais profundamente esse trecho do gráfico, podemos ver que a curva dos estudantes com participação prévia se mantém constante a partir do L3 até o dia 300. Isso indica que há apenas um estudante em risco durante esse período, o que sugere que essa diferença deva ser interpretada com cuidado, uma vez que ela se baseia na participação de apenas um estudante.

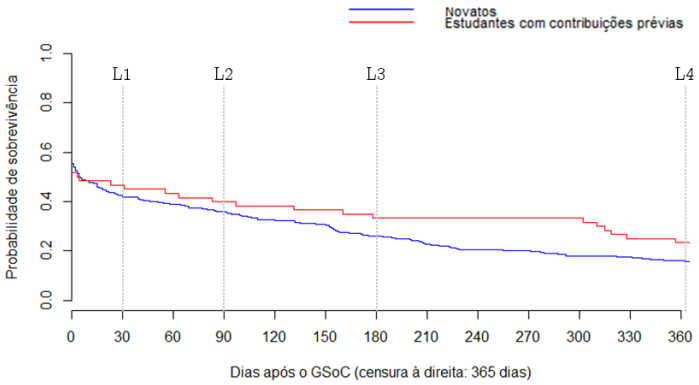
Para verificar se há diferença entre as curvas, aplicamos o teste estatístico Log-Rank para todos os limiares de censura estudados. Os valores-p obtidos foram: valor-$p_{(L1)}$=0,635; valor-$p_{(L2)}$ = 0,599; valor-$p_{(L3)}$ = 0,342; valor-$p_{(L4)}$ = 0,185. Assim, não conseguimos rejeitar a hipótese nula, que estabelece a equivalência estatística entre as curvas de sobrevivência.

De modo complementar, analisamos a duração esperada dos estudantes no GSoC por ano, conforme apresentado na Figura 8. Como anteriormente, apresentamos as curvas de sobrevivência para os novatos e os estudantes com participação prévia separadamente. Assim como na Figura 7, a probabilidade inicial de sobrevivência inicial ficou pouco acima de 50% nas três edições do programa estudadas, seguida de alta mortalidade nos primeiros dias após o término do programa. Nas três edições os estudantes com participação prévia apresentaram uma mortalidade menor após o término, comparada à dos novatos.

Além disso, podemos observar pelos valores-p apresentados na Figura 8 (a) e (b) que não conseguimos rejeitar a hipótese nula. Portanto, essas as curvas de sobrevivência de novatos e de estudantes com participação prévia são estatisticamente equivalentes. Esse resultado não suporta as descobertas em estudos anteriores como os apresentados por Bin, Robles e Serebrenik [31] e Schilling *et al.* [12], que constataram correlação forte entre experiência prévia de desenvolvimento e a retenção de estudantes (ou desenvolvedores de projetos de SL). Entretanto, a curva de sobrevivência apresentada na Figura 8 (c) apresentou diferença significativa estatística. Assim, para a edição do GSoC de 2015, nossos resultados corroboraram os estudos anteriores mencionados. Uma possível explicação para este fenômeno é que houve uma redução na sobrevivência dos estudantes novatos ao passo que houve um pequeno aumento na sobrevivência dos estudantes experientes. No entanto, novos estudos são necessários para entender os motivos dessa (aparente) discrepância, por meio de novas minerações de repositórios e entrevistas.

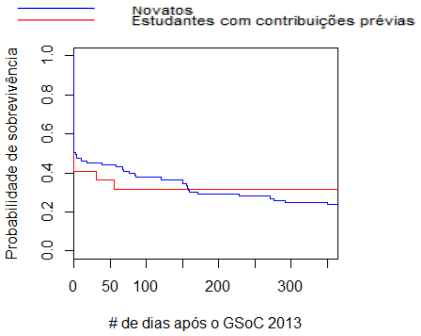
Da mesma forma, a duração esperada dos novatos que participaram da edição de 2015 foi curiosamente mais baixa em L4 que nas outras duas edições. No limiar de censura à direita L4 (i.e., 365 dias), apresentado na Figura 8 (c), teve apenas 4 ao passo que nas edições do GSoC foram censurados mais de 20 estudantes novatos. Assim, a edição do GSoC de 2015 apresentou um índice de mortalidade maior que nas outras edições.

Embora a resposta a estas questões esteja fora do escopo deste estudo, estes resultados indicam a direção para a condução de novos estudos, que possam esclarecer as motivações subjacentes ao comportamento desses estudantes.



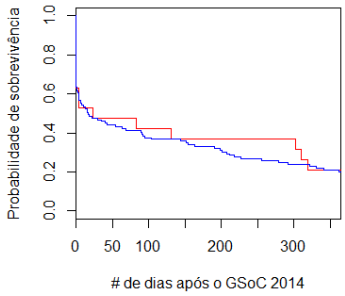
# de novatos:			307
# de estudantes com participação prévia:			60
# de estudantes com participação prévia em risco em L1 →	28	131	**← # de novatos em risco em L1**
# de estudantes com participação prévia em risco em L2 →	24	110	**← # de novatos em risco em L2**
# de estudantes com participação prévia em risco em L3 →	21	80	**← # de novatos em risco em L3**
# de estudantes com participação prévia em risco em L4 →	15	49	**← # de novatos em risco em L4**
# de estudantes com participação prévia censurados após L4 →	5	5	**← # de novatos censurados após L1**

Figura 7. Curva de sobrevivência de novatos e estudantes com participação prévia (em dias após o GSoC)



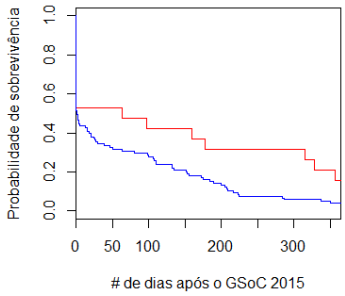
# de novatos:	93
# de estudantes com participação prévia:	22
valor-p:	0,70500
# de novatos censurados:	24
# estudant. com exp. censurados:	8

(a) Curva de sobrevivência de novatos e estudantes com participação prévia participantes do GSoC 2013



# de novatos:	109
# de estudantes com participação prévia:	19
valor-p:	0,0824
# de novatos censurados:	23
# estudant. com exp. censurados:	5

(b) Curva de sobrevivência de novatos e estudantes com participação prévia participantes do GSoC 2014



# de novatos:	105
# de estudantes com participação prévia:	19
valor-p:	0,0334
# de novatos censurados:	4
# estudant. com exp. censurados:	5

(c) Curva de sobrevivência de novatos e estudantes com participação prévia participantes do GSoC 2013

Figura 8. Detalhamento da curva de sobrevivência de estudantes com participação prévia e novatos

Deste modo, de acordo com a amostra analisada, as comunidades de SL podem esperar que cerca de 40% de seus estudantes–estudantes novatos e estudantes com participação prévia–ainda estejam participando 30 dias (L1) após o término; ~35% deles, 90 dias (L2) após; ~25% deles, 180 dias (L3) após; e ~15% deles, 365 dias (L4) após o término oficial do GSoC. No entanto, ressaltamos a necessidade de realizar pesquisas adicionais para entender as reais razões das (aparentes) discrepâncias apresentadas em nossos resultados com relação aos da literatura de retenção atual.

5 DISCUSSÃO

Uma questão que surge para algumas comunidades de SL é sobre o retorno no investimento de mentoria em termos de contribuição de código e novos voluntários. De fato, existem comunidades que almejam reter os estudantes como novos colaboradores, conforme evidenciado pelo trecho a seguir:

> *"(...) Participar no GSoC aumentará a visibilidade dos esforços de projeto da Pharo (...) Esperamos trazer mais pessoas em nossa comunidade [participando do GSoC]"*
> Pharo. Tradução livre. Fonte: http://bit.ly/2mtN0Xr

Entretanto, a participação em eventos de engajamento nas comunidades de SL envolve o custo-benefício entre o esforço e habilidade do mentor de simultaneamente atender às demandas do seu projeto de SL. Esses foi um dos motivos que fizeram a comunidade Debian não participar do GSoC17:

> *"Devido ao baixo nível de motivação geral, e principalmente pela fraqueza de nossa página de projetos durante a revisão da Google (..) a comunidade Debian não fará parte do GSoC neste ano [2017]. Alguns de nossos costumeiros mentores tem mostrado sinais de 'fatiga do GSoC', (...) va-*

mos nos dar o verão para nos recuperar (...) e voltar no próximo ano"
Debian. Tradução livre. Fonte: http://bit.ly/2nT0h99

Embora deixemos como trabalho futuro a tarefa de investigar o real aprendizado dos estudantes ou ainda a natureza das tarefas desempenhadas pelos estudantes, nossos resultados sugerem que algumas comunidades de SL têm suas tarefas realizadas, especialmente as comunidades que trabalharam com estudantes experientes. Isto é compreensível uma vez que é usualmente difícil para os novatos irem do aprendizado da contribuição ao desenvolvimento de contribuições significativas em pouco tempo.

"O GSoC é um programa importante, porque possibilita intensa mentoria de estudantes por um período relativamente longo de tempo. O estudante ganha mais experiência, enquanto o projeto tem suas tarefas realizadas, que [do contrário] seriam difíceis de serem realizadas [pelos] voluntários (...)"
LibreOffice. Tradução livre. Fonte: http://bit.ly/2n1xt1u

Uma possível implicação de nossos resultados é que quando as comunidades selecionam estudantes novatos para participar do GSoC, devem se preparar para investir em mentoria sem expectativas de comprometimento de longo-prazo.

Previsivelmente, o período com maior participação foi durante o GSoC (período patrocinado), como ilustrado, por exemplo, na Figura 3. Nossos resultados mostraram que ~64% dos estudantes de nossa amostra não permaneceram mais que um mês após o programa. Baseado neste resultado, **sugerimos que as comunidades desenvolvam estratégias para lidar com os estudantes que deixam de contribuir**. Pesquisas futuras podem focar em meios de mitigar o desaparecimento de estudantes.

Nossos resultados também sugerem que os eventos de engajamento propiciam às comunidades contribuições de candidatos antes do início do GSoC, possivelmente devido à natureza competitiva dos programas. **As comunidades de SL poderiam oferecer um programa pré-evento (ou pós-evento), no intuito de engajar participantes**. Assim, a comunidade se aproveitaria da participação dos candidatos antes do programa, oferecendo uma oportunidade formal para os candidatos mostrarem suas habilidades e interagir com a comunidade. Como resultado, o projeto receberia mais contribuições e teria mais oportunidades de exibi-las à comunidade. Isso ajudaria a classificar e selecionar candidatos. Esta estratégia poderia funcionar para a comunidade BioPython, que experimentou problemas similares, conforme reportado por Trainer e colegas [10].

As comunidades poderiam oferecer oportunidades de participação voluntária aos não selecionados para os eventos de engajamento. Nesse caso, os participantes poderiam ser premiados com certificados de participação etc. Dessa forma, mesmo os candidatos que não fossem patrocinados teriam chances de adquirir conhecimento, experiência e ter um selo de participação.

Nossos resultados sugerem que encontrar colaboradores recorrentes, embora raro, pode recompensar a comunidade com grandes dividendos, considerando o número de commits (aceitos). As descobertas anteriores – alta visibilidade, contribuição como estratégia de aumentar as chances de aceitação, código aceito durante o programa e encontrar colaboradores recorrentes – podem explicar porque as comunidades interessadas em participar do GSoC têm aumentado ao longo dos anos.

Os eventos de engajamento parecem ser um canal de contribuição para projetos de SL que ajuda a mitigar barreiras para os estudantes novatos (veja Steinmacher *et al.* [16] para um panorama das barreiras que os novatos usualmente enfrentam) e atrai contribuições de quem talvez não teria contribuído de outra forma.

6 AMEAÇAS À VALIDADE

Esta pesquisa possui limitações, e nesta seção, apresentamos como buscamos mitigá-las. Em primeiro lugar, nossa amostra pode não ser representativa de toda a população dos estudantes GSoC, apesar de nossos esforços para coletar uma amostra representativa. Logo, é possível chegar a outras conclusões tendo-se um conjunto diferente de estudantes.

Uma grande ameaça é a identificação errônea de estudantes e seus projetos nos SCVs, assim como a dos IDs dos estudantes na base de dados local. Por exemplo, em alguns casos, os IDs tanto nos SCVs quanto na base de dados local, eram compostos das iniciais dos estudantes (ou combinações). Embora tenhamos excluídos os casos em que não tínhamos certeza, ainda assim é possível que tenhamos incorrido em identificação errônea.

Em alguns casos, o mesmo estudante utilizou diversas IDs para fazer os commits. Nesse caso, as ameaças são que poderíamos ter agrupado incorretamente os IDs de diferentes estudantes; não ter identificado todos os IDs; e/ou ter identificado os IDs utilizados em uma edição do GSoC diferente daquelas que estamos considerando. Embora tenhamos feito uma análise aprofundada de cada estudante de nossa amostra, ainda é possível que estas ameaças tenham enfraquecido nossos resultados.

Adicionalmente, utilizamos os nomes de estudantes e seus nomes de projetos como critério de busca para determinar se os estudantes participaram como mentores em outras edições. Para o caso de estudantes homônimos trabalhando para o mesmo projeto, podemos ter erroneamente contabilizado como sendo o mesmo estudante. Reduzimos essa ameaça inspecionando se o ano de participação como estudante era anterior ao ano de participação como mentor para o mesmo projeto. Adicionalmente, como não contatamos mentores, é possível que os estudantes tenham entregue seu código após a data final oficial do GSoC, que por nosso método seria erroneamente contado como retenção.

Finalmente, nossas conclusões podem estar enviesadas com relação ao número de commits aceitos. Não controlamos variáveis como linguagem de programação, complexidade de código ou o quão importante os commits aceitos foram para as comunidades. Pode ser o caso de que estudantes que tenham um único commit aceito no repositório tenham contribuído mais – em termos de valor agregado – que aqueles que contribuíram mais numericamente.

7 CONCLUSÃO

As comunidades de SL esperam que eventos de engajamento tais como os Summers of Code sejam um canal para a atração e

retenção de novatos [5], [9]–[11]. Neste artigo, investigamos o Google Summer of Code (GSoC), fornecendo evidências de diversos aspectos da participação dos estudantes, tais como o quanto o GSoC fomentou contribuições (*i.e.*, commits, commits aceitos ao repositório e *churn* de código) e a curva de sobrevivência dos estudantes nos projetos de SL após o término do programa.

Com relação à QP1 (Quanto código os participantes do GSoC contribuem?), descobrimos que 82% dos projetos de nossa amostra aceitaram pelo menos um commit de autoria dos estudantes. Quando consideramos apenas novatos, ~54% dos projetos de SL aceitaram pelo menos um commit. Também reportamos a magnitude das contribuições dos estudantes analisando as medianas do *churn* de código: 1,5M (*antes*); ~8,9M (*durante*); e 30M (*após*) o GSoC.

Já com relação à QP2 (Os estudantes com participação prévia nos projetos permanecem contribuindo por mais tempo que os estudantes novatos?), descobrimos que ~23% dos novatos começaram a contribuir aos projetos antes de saberem que seriam aceitos pelo programa. Não encontramos diferenças estatísticas com relação aos níveis de retenção de novatos e estudantes com participação prévia, com exceção para a edição do GSoC 2015. Por meio de análise de sobrevivência, encontramos que ~40% dos estudantes contribuem por mais de um mês enquanto ~15%, mais de um ano.

Concluímos ressaltando que muitas comunidades online, incluindo as comunidades de SL, enfrentam problemas para atrair e reter colaboradores. Nossos resultados indicam que muitos projetos de SL conseguiram reter novos colaboradores e incorporar ao repositório códigos de autoria dos estudantes. Futuras pesquisas podem estender os resultados investigando mais profundamente o tipo de contribuição dos estudantes. Sugerimos às comunidades que utilizem estratégias para lidar com os estudantes que deixam de contribuir logo nos primeiros dias após o programa. Adicionalmente, as comunidades de SL podem estabelecer um período recomendado antes dos eventos para que os candidatos possam começar a contribuir e interagir com a comunidade. Assim, candidatos que comecem mais cedo tenham mais chances de aceitação.

REFERÊNCIAS

[1] P. Resnick and R. E. Kraut, *Building Successful Online Communities: Evidence-Based Social Design*, no. May. The MIT Press, 2009.

[2] Y. Fang and D. Neufeld, "Understanding Sustained Participation in Open Source Software Projects," *J. Manag. Inf. Syst.*, vol. 25, no. 4, pp. 9–50, 2009.

[3] G. Pinto, I. Steinmacher, and M. A. Gerosa, "More Common Than You Think: An In-depth Study of Casual Contributors," *2016 IEEE 23rd Int. Conf. Softw. Anal. Evol. Reengineering*, vol. 1, no. 1, pp. 112–123, 2016.

[4] K. Crowston, H. Annabi, and J. Howison, "Defining Open Source Software Project Success," in *Proceedings of the 24th International Conference on Information Systems (ICIS)*, 2003, pp. 1–14.

[5] E. H. Trainer, C. Chaihirunkarn, A. Kalyanasundaram, and J. D. Herbsleb, "Community code engagements: Summer of Code & hackathons for community building in scientific software," in *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work*, 2014, pp. 111–121.

[6] F. J. García-Peñalvo *et al.*, "Developing Win-win Solutions for Virtual Placements in Informatics: The VALS Case," in *Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality*, 2014, pp. 733–738.

[7] I.-H. Hann, J. Roberts, S. Slaughter, and R. Fielding, "Economic Incentives for Participating Open Source Software Projects," *Twenty-Third Int. Conf. Inf. Syst.*, vol. ICIS 2002, 2002.

[8] J. Tirole and J. Lerner, "Some Simple Economics of Open Source," *J. Ind. Econ.*, vol. 50, no. 2, pp. 197–234, 2002.

[9] L. Christopherson, R. Idaszak, and S. Ahalt, "Developing Scientific Software through the Open Community Engagement Process," in *First Workshop on Sustainable Software Science: Practice and Experiences*, 2013.

[10] E. H. Trainer, C. Chaihirunkarn, and J. D. Herbsleb, "The Big Effects of Short-term Efforts: Mentorship and Code Integration in Open Source Scientific Software," *J. Open Res. Softw.*, vol. 2, no. 1, p. Art. e18, 2014.

[11] E. H. Trainer, A. Kalyanasundaram, C. Chaihirunkarn, and J. D. Herbsleb, "How to Hackathon: Socio-technical Tradeoffs in Brief, Intensive Collocation," in *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing - CSCW '16*, 2016, pp. 1116–1128.

[12] A. Schilling, S. Laumer, and T. Weitzel, "Who will remain? - An evaluation of actual Person-Job and Person-Team fit to predict developer retention in FLOSS projects," *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, pp. 3446–3455, 2011.

[13] K. R. Lakhani and R. G. Wolf, "Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects," in *Perspectives on Free and Open Source Software*, Cambridge: MIT Press, 2005.

[14] A. Hars and S. Shaosong Ou, "Working for free? Motivations of participating in open source projects," in *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, 2001, vol. 7, p. 9.

[15] J. a. Roberts, I.-H. Hann, and S. a. Slaughter, "Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects," *Manage. Sci.*, vol. 52, no. 7, pp. 984–999, 2006.

[16] I. Steinmacher, M. A. Gerosa, D. F. Redmiles, T. Conte, M. A. Gerosa, and D. F. Redmiles, "Social Barriers Faced by Newcomers Placing Their First Contribution in Open Source Software Projects," *Proc. 18th ACM Conf. Comput. Support. Coop. Work Soc. Comput. - CSCW '15*, pp. 1379–1392, 2015.

[17] I. Steinmacher, I. S. Wiese, T. Conte, M. A. Gerosa, and D. Redmiles, "The hard life of open source software project newcomers," *Proc. 7th Int. Work. Coop. Hum. Asp. Softw. Eng. - CHASE 2014*, pp. 72–78, 2014.

[18] F. Fagerholm, A. S. Guinea, J. Münch, and J. Borenstein, "The role of mentoring and project characteristics for onboarding in open source software projects," *ESEM conf.*, pp. 1–10, 2014.

[19] P. Meirelles, C. Santos, J. Miranda, F. Kon, A. Terceiro, and C. Chavez, "A Study of the Relationships between Source Code Metrics and Attractiveness in Free Software Projects."

[20] J. Colazo and Y. Fang, "Impact of license choice on open source software development activity," *J. Am. Soc. Inf. Sci. Technol.*, 2009.

[21] C. Santos, G. Kuk, F. Kon, and J. Pearson, "The attraction of contributors in free and open source software projects," *J. Strateg. Inf. Syst.*, vol. 22, no. 1, pp. 26–45, 2013.

[22] K. J. Stewart and S. Gosain, "The impact of ideology on effectiveness in open source software development teams," *MIS Q.*, vol. 30, no. 2, pp. 291–314, 2006.

[23] S. Krishnamurthy, S. Ou, and A. K. Tripathi, "Acceptance of monetary rewards in open source software development," *Res. Policy*, vol. 43, no. 4, pp. 632–644, 2014.

[24] M. Zhou and A. Mockus, "What make long term contributors: Willingness and opportunity in OSS community," in *34th International Conference on Software Engineering*, 2012, pp. 518–528.

[25] I. S. Wiese, J. T. da Silva, I. Steinmacher, C. Treude, and M. A. Gerosa, "Who is Who in the Mailing List? Comparing Six Disambiguation Heuristics to Identify Multiple Addresses of a Participant," *2016 IEEE Int. Conf. Softw. Maint. Evol.*, pp. 345–355, 2016.

[26] J. T. Rich, J. G. Neely, R. C. Paniello, B. Voelker, C. C. J., Nussenbaum, and E. W. Wang, "A practical guide to understanding Kaplan-Meier curves," *Otolaryngol. - Head Neck Surg.*, vol. 143, no. 3, p. 6, 2010.

[27] P. M. E.L. Kaplan, "Nonparametric Estimation from Incomplete Observations," *Am. Stat. Assoc.*, vol. 53, no. 282, pp. 457–481, 1958.

[28] C. Bird, A. Gourley, P. Devanbu, and G. H. Anand Swaminathan, "Open Borders? Immigration in Open Source Projects," in *Proceedings of the Fourth International Workshop on Mining Software Repositories*, 2007.

[29] I. Samoladas, L. Angelis, and I. Stamelos, "Survival analysis on the duration of open source projects," *Inf. Softw. Technol.*, vol. 52, pp. 902–922, 2010.

[30] M. Goeminne and T. Mens, "Towards a survival analysis of database framework usage in Java projects," *2015 IEEE 31st Int. Conf. Softw. Maint. Evol. ICSME 2015 - Proc.*, pp. 551–555, 2015.

[31] B. Lin, G. Robles, and A. Serebrenik, "Developer Turnover in Global, Industrial Open Source Projects: Insights from Applying Survival Analysis," *Proc. 12th Int. Conf. Glob. Softw. Eng.*, pp. 66–75, 2017.

[32] P. N. Sharma, J. Hulland, and S. Daniel, "Examining Turnover in Open Source Software Projects Using Logistic Hierarchical Linear Modeling Approach," in *International Conference on Open Source Systems*, 2012.

[33] D. Izquierdo-Cortazar, G. Robles, F. Ortega, and J. M. Gonzalez-Barahona, "Using software archaeology to measure knowledge loss in software projects due to developer turnover," in *HICSS*, 2009, pp. 1–10.

[34] P. Daniel, "Want to be selected for Google Summer of Code 2016?," 2015. [Online]. Available: http://danielpocock.com/getting-selected-for-google-summer-of-code-2016. [Accessed: 10-Feb-2017].