Who Drive Company-Owned OSS Projects: Employees or Volunteers?

Luiz Felipe Dias[1], Jhoylan Santos[2], Igor Steinmacher[1], Gustavo Pinto[2]

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

[2]Federal University of Pará (UFPA) – Belém – PA - Brazil

luizdias@alunos.utfpr.edu.br, jhoylangs@gmail.com
igorfs@utfpr.edu.br, gpinto@ufpa.br

Abstract. *It is well known that Open-source software (OSS) communities leverage the workforce of volunteers to keep the projects sustainable. Some companies support open-source software projects by paying developers to contribute, while others share their products under open-source licenses keeping their employees in charge of maintaining the projects. In this paper, we investigate who drives open-source products: paid (internal) or non-paid (external) developers. We explore two well-known, open-source software:* `atom` *and* `hubot`*. Using a mix of quantitative and qualitative approaches, we found that both internal and external developers are rather active, when it comes to pull-requests submitted, and that the projects are very receptive for external developers. Considering both projects, internal developers are responsible for 45.54% and external developers placed 54.46% of the pull-requests. Finally, we observed that external developers' contributions range from documentation to complex code.*

Keywords: Company-Owned OSS Projects, Employees, Volunteers.

1. Introduction

Open-source software is one of the cornerstones of modern software development practice. Many existing software projects rely on open-source solutions either at compile time (*e.g.,* IDEs, build tools, or testing tools) or runtime (*e.g.,* webservers, databases, or queues). In spite of its ubiquitousness, several open-source software rely on a single contributor to perform most of the needed tasks [Avelino et al. 2016]. Due to this unfriendly scenario, it is not uncommon core developers becoming tired and abandoning their own software projects [Coelho and Valente 2017]. To make the matters worse, only few open-source projects are sponsored by some organization [Choi 2012].

Although few software companies support open-source activities [Choi 2012], there is a recurrent belief that most of the open-source contributions software are made by paid developers. Indeed, a recent article pointed out that: *"More than 80 percent of kernel development is done by developers who are being paid for their work."*[1] While commercial contributions to the Linux kernel have been widely acknowledged, in a large-scale study of more than 9,000 open-source projects, Riehle and colleagues [Riehle et al. 2014] observed that about 50% of the open-source contributors are actually paid ones. However, in this work, the authors consider "paid developers" the ones that performed commits from

[1]`https://www.linuxfoundation.org/news-media/announcements/2015/02/linux-foundation-releases-linux-development-report`

9am to 5pm, local time. Using this simple rule, students, unemployed, or workers with flexible time schedules could be wrongly sampled as "paid developers".

We believe it is important to shed additional light on this direction, due two at least two reasons:

1. If there are, indeed, too many paid developers, open-source communities may need to better explore these workforces. For instance, instead of concentrating too many paid developers in one single open-source project, open-source communities could try to gather some paid developers to well-needed open-source projects.
2. On the other hand, if there are too few paid developers, this finding might not only refute previous studies, but yet can be used to better motivate software companies on the importance of supporting open-source projects.

It is important to note that the source of payment can vary greatly. For instance, one can get payed to fix a bug, whereas others can be full time open-source contributors. In this study, we pay particular attention to developers that contribute to open-source as part of their daily job (*e.g.,* the company that they work for maintain an open-source software project). Throughout this paper, we refer to them as "internal developers". Developers that do not work for the company that maintains the open-source project are refereed as "external developers".

Although in its early stages, this paper provides an investigation about the contribution behavior of pull-requests provided by internal and external developers in two well-known, non-trivial open-source software products: `atom` and `hubot`. We chose these projects because they were initially developed by (and are maintained at) GitHub, therefore we could take advantage of GitHub features to understand whether a contributor is a internal or external one (more details at Section 2). Through a set of quantitative and qualitative analysis, this paper makes the following contributions:

- Bringing a quantitative analysis on the contributions performed by internal and external developers on company-owned OSS projects;
- Understanding the receptivity of external developers on company-owned OSS projects;
- Shedding the light that the contribution behavior is project-dependent, and it is necessary to study the projects individually to better understand the phenomenon.

2. Method

Studied projects. We provide an in-depth investigation on the contributions (*i.e.*, a pull-request) made at two well-known open-source projects.

- `hubot`, a customizable life embetterment robot. It has ∼2,000 commits, ∼700 pull-requests, 248 source code contributors, ∼13,000 stars, and ∼3,000 forks. It is mostly written in JavaScript, and has ∼5 years of historical records.
- `atom`, a cross-platform text editor. It has ∼32,500 commits, ∼3,500 pull-requests, 363 source code contributors, ∼38,500 stars, and ∼6,800 forks. It is mostly written in CoffeScript and JavaScript, and has ∼6 years of historical records.

We chose these projects because they were initially developed by a software company (GitHub), but became open-source at some point of their life-time. The focus of this

paper is to better understand the contribution behavior of paid developers (internal) and non-paid developers (external) in company-owned open-source software projects.

Since these projects are developed by (and maintained at) GitHub, we reduce false positives by taking advantage of GitHub features used to identify developers roles. For instance, within GitHub organizations, one coordinator can set the `site_admin` flag true for another user. If enabled, this flag promotes an ordinary user to be a site administrator. According to GitHub official documentation, a site administrator can *"manage high-level application and VM settings, all users and organization account settings, and repository data"*[2]. Therefore, for each pull request investigated, we verified whether the author has the `site_admin` flag enabled. If so, we marked she as *internal*, *external* otherwise. To avoid false negatives (a paid developer that does not have its `site_admin` flag enabled), we also manually match their GitHub profiles with their LinkedIn profile (details at Section 5).

Approach. For the two analyzed projects, we start by investigating all performed pull-requests. A pull request can be found in three different stages:

- *open*: waiting for code reviews and/or a final decision;
- *closed*: the code reviews were done, but the pull-request was not accepted;
- *merged*: the code reviews were done, and the pull-request was accepted.

We studied the contribution behavior of internal and external developers taking into account each possible stage of a pull-request. We also investigated additional characteristics such as the number of commits per pull-requests and the number of comments of code reviews per pull-request. The data reported in this paper are based on pull-requests that were performed from the very beginning of the projects, up to June, 2017 — when we collected data. Still, in order to uncover the reasons for acceptance, we selected a representative sample (confidence level of 95% with a $\pm$5% confidence interval) of 334 accepted pull-requests at `atom` for manual analysis. We also validated this analysis with another manual analysis in a random sample of 150 pull-requests accepted at `hubot`. All data used in this study is available online at: `https://github.com/fronchetti/VEM-2017`.

For statistics, we used Mann-Whitney-Wilcoxon (MWW) tests [Wilks 2011] and Cliff's delta effect-size measures [Grissom and Kim 2005]. We used MWW tests to verify if two distributions come from the same population (α = 0.05). We used Cliff's delta to verify how often values in one distribution are larger than values in another distribution. The thresholds are defined as follows: $delta < 0.147$ (*negligible*), $delta < 0.33$ (*small*), $delta < 0.474$ (*medium*), and $delta >= 0.474$ (*large*) [Romano et al. 2006].

3. Results

Generally speaking, both internal and external developers are rather active, when it comes to pull-requests submitted. On `atom`, internal developers submitted 1,772 pull-requests whereas external ones submitted 1,574; on `hubot` external developers submitted 692 pull-requests, and internal provided 123. If we consider both projects, we come to 1,895 pull-requests provided by internal developers (45,54%) and 2,266 by external ones (54.46%). However, the number of contributors are much lesser than the number of contributions: While 664 external developers performed contributions to `atom`, only 33 internal

[2] `https://enterprise.github.com/security`

developers were found (for `hubot`, the numbers are, respectively, 355 and 19). That is, although the number of external developers are up to 20× greater than internal ones, the amount of contributions is somewhat similar. Interestingly, we found that casual contributors (developers that made only one contribution [Pinto et al. 2016]) are commonplace in our dataset, even for the internal group (21% are internal casuals and 76% are external casuals).

To provide a more detailed overview, Figure 1 depicts the evolution of pull-requests, grouped by its state (open, closed, and merged).

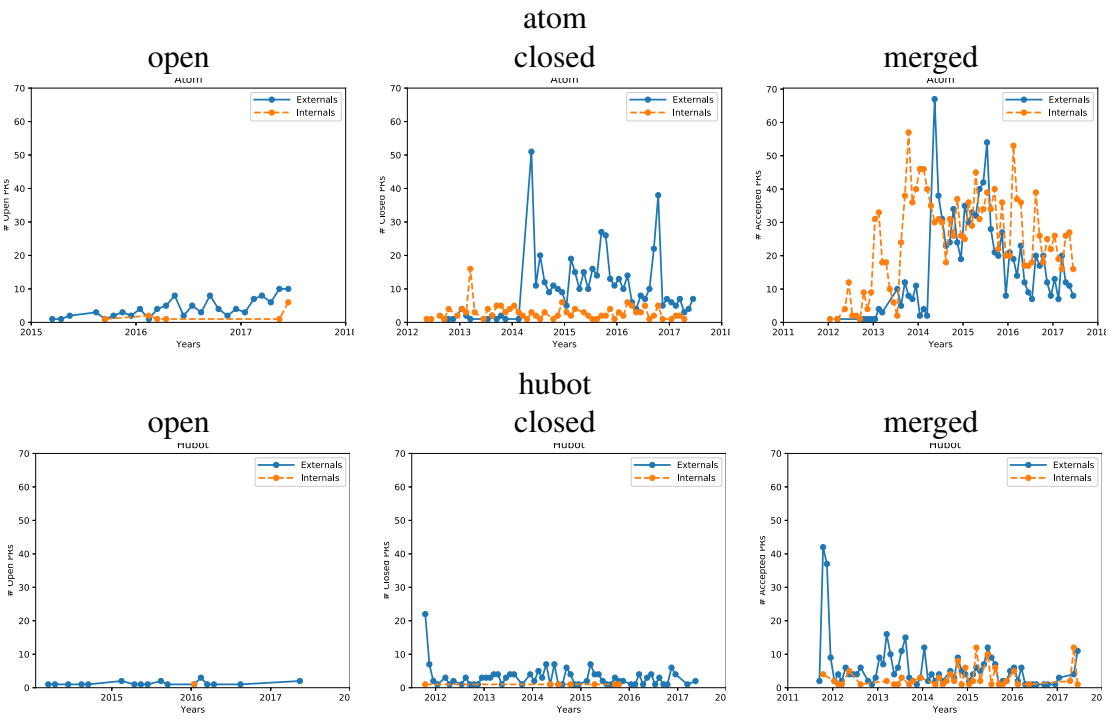

Figure 1. Figures on the top are for project `atom`, whereas figures in the bottom are for `hubot`. From the left to right, each figure shows the number of pull requests open, closed, and merged. Orange lines represent internal developers, whereas blue lines represent external developers.

From the figures, one might believe that external developers have more open pull requests than internal ones. However, this hypothesis was refuted (*p-value*: 0.9169, effect-size: -0.01 (negligible)), which suggests feedback might be provided for both groups (which is not always the case in open-source communities [Dias et al. 2016]). Although some old pull-requests are still open, ultimately pull-requests ended up being closed or merged.

As regarding closed – but unmerged – pull-requests (the ones that were not accepted), we could notice that many external developers are having a hard time attempting to get their contributions accepted (for `atom`, internals have 157 pull-requests not accepted, whereas externals have 508 ones). In fact, for atom, the hard time of external developers is statistically significant (*p-value*: 0.0002366), although with a small effect size (0.152628). For `hubot`, the rate is similar.

Finally, when studying the merged pull-requests (the accepted ones), we can see

that both groups are also fairly active (for atom, internals have 1,603 pull-requests accepted, whereas externals have 957 ones). However, a negligible, negative effect size (-0.09482301) refutes the hypothesis that most of the work is done by internal developers (*p-value*: 0.0005615). To better understand the characteristics of the accepted contributions, we conducted a qualitative analysis aimed at investigating the reasons for pull-request acceptance, in particular, the ones proposed by external members.

For the `atom` project, before creating a pull-requests, internal members create an issue that describes what are the project needs. Therefore, most of the pull-requests proposed are accepted because internal members were *expecting* it. Pull-requests that fix documentation problems are the most common ones (we found 27 instances of them). Some example include: broken URL[3], not enough information[4], and code comments[5]. In addition, contributions from external members are shorter than internal ones. As noted elsewhere, smaller changes can reduce the chance of breaking the continuous integration build [Rebouças et al. 2017]. Notwithstanding, non-trivial code changes often come with a detailed description (images are common). However, all pull-requests are subject to rigorous code review process. We found a similar pattern for `hubot`. Most of the pull-requests from external members are related to documentation issues[6], although complex code changes exist[7]. Finally, these two projects seem to welcome external users: they not only answer most of the requests from external members, but they also guide their contributions to an acceptable state.

We also investigated how these two group of contributors differ in terms of **number of commits** per pull-request and the **number of comments** received during pull-request's code review. Figure 2 shows the distribution of these two metrics, while Table 1 shows some descriptive statistics.

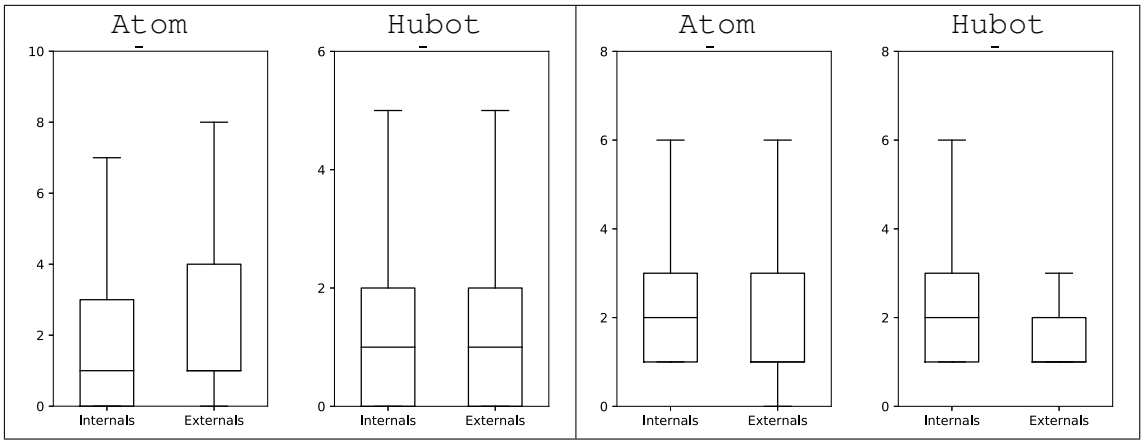

Figure 2. Distribuition of commits and comments (code reviews) per accepted pull-request.

As we can see, both groups have similar behavior regarding the number of commits performed and comments received. The result is statistically significant (*p-value* $<$

[3] https://github.com/atom/atom/pull/1929
[4] https://github.com/atom/atom/pull/2602
[5] https://github.com/atom/atom/pull/8452
[6] https://github.com/hubotio/hubot/pull/788
[7] https://github.com/hubotio/hubot/pull/489

0.01) for first three boxsplots, with a small effect size. As for the number of commits, this finding suggests that both groups follow well-known guidelines for contributing to open-source (few commits per pull-request [Gousios et al. 2014]). Moreover, although internal developers might be more aware of project domain, the integration process, and their peers, they face a similar pull-request review process (in terms of number of comments), when compared to external developers. This last finding corroborates with our previous one: our studied projects seem to welcome external developers.

Table 1. Descriptive statistics about the number of commits and comments.

	Commits			**Comments**		
	1st Quartile	Median	3rd Quartile	1st Quartile	Median	3rd Quartile
Internals	1	3	8	0	1	3
Externals	1	1	3	1	1	4

4. Related Work

In this section, we discuss some of the studies that relate with the scope of this work, which deal with proprietary software projects on GitHub, commercial involvement in OSS projects, and paid developers in OSS.

Proprietary Software Projects on GitHub . Kalliamvakou *et al.* [2015] examined how organizations with projects producing proprietary software use GitHub to develop software. They found that these projects apply typical software development practices used in OSS projects, including reduced communication, independent work, and self-organization. In our study, we also analyze proprietary projects that use GitHub. However, we are interested in those projects that became open-source, while Kalliamvakou and colleagues' work focused on the use of GitHub infrastructure to produce closed source, proprietary software in private repositories.

Commercial Involvement/Paid Developers in OSS Projects . It is possible to notice an increase in the participation of companies in OSS and in the contributions of employees paid to work on OSS projects [Riehle et al. 2014, Zhou et al. 2016]. Zhou *et al.* analyzed how commercial involvement in OSS communities influenced the onboarding of new developers. By studying OpenStack, Docker, and Android, they found that the way the commercial involvement takes place is associated with developers participation. Homscheid and Schaarschmidt [2016] investigated the role of external developers who are paid by third-party companies ("firm-sponsored developers"). By conducting a survey with Linux developers, they found that the perceived external reputation of the employing organization reduces turnover intention towards the company, and the perceived own reputation dampens turnover intention towards the OSS community. Atiq and Tripathi [2016] explored how the developers perceive the differences of rewards in OSS projects, by analyzing their opinion on how project's financial resources influences the progress of the project. By analyzing an open question sent to OSS developers, they found that OSS projects where only some people get directly paid may fail if they are mismanaged.

Riehle *et al.* [2014] analyzed more than 5,000 active open-source projects, from 2000 to 2007, and found that around 50% of all contributions have been paid work. Their perspective is that any contribution made from Monday to Friday, between 9am and 5pm

are paid contributions. However, as highlighted by Crowston [2016], even employed developers are not paid directly by the projects to which they contribute, so from the project perspective, they are volunteers. Thus, differently from Riehle and colleagues, we analyzed the amount of effort put by the developers of the company that open-sourced the project – directly paid by the "owner" – comparing with the contributions made by any external developer. Our results showed that, for the analyzed projects 45% of the pull-requests are placed by internal developers (GitHub employees). The results seem to be inline with previous work, except for the fact that the concept of paid developers used previously, is not the same as the concept of external developers applied here.

5. Limitations

First, we rely on our inference algorithm to verify whether a contributor is an internal or external one. We made use of a flag made available in the pull-request to make this decision. We acknowledge that this can be a threat. To minimize this, we investigated the affiliation of the contributors. We found that two members we classified as external presented GitHub as their organizations. We further analyzed their profile, and found that they left GitHub and are now working in other companies. For those classified as internal members, all listed themselves as GitHub staff in their profile. Second, one might argue that we could differentiate paid and non-paid developers by looking at the email address used at their contributions (if it is a corporative email, then the developer is a paid one). We argue that developers are free to choose whenever email account they want to use at the git repository. Therefore, a paid developer can also contribute with her personal email account (which would represent a false positive). We use the site_admin flag to mitigate this threat. Finally, as we analyzed just two projects from the same company, we understand that the results cannot be generalized. However, a small sample enabled us to validate the algorithm used to classify the contributors manually, and to manually analyze some of the pull-requests.

6. Conclusions

In this paper we analyzed the contribution behavior of internal (*i.e.,* paid) and external (*i.e.,* non-paid) developers of `atom` and `hubot` projects. We found that these projects are very receptive for external developers. We also found that internal and external developers placed around 50% of the pull-requests submitted when considering both projects together (45% from internal members vs. 55% from external). However, analyzing just `hubot`, which is smaller project, we observed that only 18% of the pull-requests had been placed by internal members. These differences indicate that it is necessary to analyze each project individually to better understand this phenomenon, since there can be different factors influencing the behavior, like: the priority the company is giving to the project; the project attractiveness; and vendors who make use of the project. We also noticed that, contribution from external developers are shorter than those sent by internal ones, and that external developers contribute more documentation related pull-request, although we also found complex code pull-request.

This study can be a fruitful research area which can benefit companies willing to open-source their codes, and developers who are afraid of contributing to recently open-sourced projects. For future work, we plan to expand the scope of this study by investigating additional OSS projects. In addition, we plan to conduct surveys and interviews with developers in order to cross-validate the findings from the repositories.

References

Atiq, A. and Tripathi, A. (2016). Impact of financial benefits on open source software sustainability. In 37^{th} *ICIS*.

Avelino, G., Passos, L. T., Hora, A. C., and Valente, M. T. (2016). A novel approach for estimating truck factors. In *ICPC 2016*, pages 1–10.

Choi, B. R. (2012). *Essays on Socialization in Online Groups*. Phd thesis, Tepper School of Business - Carneggie Mellon University, United States.

Coelho, J. and Valente, M. T. (2017). Why modern open source projects fail. In *FSE 2017*, pages 1–11.

Crowston, K. (2016). Open source technology development. In *Handbook of science and technology convergence*, page 475–486.

Dias, L., Steinmacher, I., Pinto, G., Costa, D., and Gerosa, M. (2016). How does the shift to github impact project collaboration? In 32^{nd} *ICSME*, pages 473–477.

Gousios, G., Pinzger, M., and Deursen, A. v. (2014). An exploratory study of the pull-based software development model. In *ICSE '14*, pages 345–355.

Grissom, R. and Kim, J. (2005). *Effect Sizes for Research: Univariate and Multivariate Applications*. Taylor & Francis.

Homscheid, D. and Schaarschmidt, M. (2016). Between organization and community: investigating turnover intention factors of firm-sponsored open source software developers. In *WebSci '16*, pages 336–337. ACM.

Kalliamvakou, E., Damian, D., Blincoe, K., Singer, L., and German, D. M. (2015). Open source-style collaborative development practices in commercial projects using github. In *ICSE '15*, pages 574–585.

Pinto, G., Steinmacher, I., and Gerosa, M. (2016). More common than you think: An in-depth study of casual contributors. In *SANER '16*, pages 112–123.

Rebouças, M., Santos, R. O., Pinto, G., and Castor, F. (2017). How does contributors' involvement influence the build status of an open-source software project? In *Proceedings of the 14th International Conference on Mining Software Repositories*, MSR '17, pages 475–478.

Riehle, D., Riemer, P., Kolassa, C., and Schmidt, M. (2014). Paid vs. volunteer work in open source. In *HICSS ' 14*, pages 3286–3295.

Romano, J., Kromrey, J., 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*.

Wilks, D. (2011). *Statistical Methods in the Atmospheric Sciences*. Academic Press.

Zhou, M., Mockus, A., Ma, X., Zhang, L., and Mei, H. (2016). Inflow and retention in oss communities with commercial involvement: A case study of three hybrid projects. *ACM TOSEM*, 25(2):13.