

# Do Onboarding Programs Work?

Adriaan Labuschagne and Reid Holmes  
School of Computer Science  
University of Waterloo  
Waterloo, ON, Canada  
alabusch,rtholmes@cs.uwaterloo.ca

**Abstract**—Open source software systems rely on community source code contributions to fix bugs and develop new features. Unfortunately, it is often difficult to become an effective contributor on open-source projects due to the complexity of the tools required to develop and test new patches and the challenge of breaking into an already-formed social organization. To help new contributors learn their development practices, OSS projects have created onboarding programs that, for example, identify easy ‘first bugs’ and mentor new developers’ contributions. However, we found that developers who join an organization through these programs are half as likely to transition into long-term community members than developers who do not use these programs. Measuring the impact of these programs is important, as coordinating and staffing onboarding projects is expensive. This paper examines onboarding programs employed by Mozilla and demonstrates that they are not as effective at transitioning new developers into long-term contributors as might be hoped, although developers who do succeed through these programs find them valuable.

## I. INTRODUCTION

Open Source Software (OSS) projects need a constant stream of newcomers to help fix defects and build new features [1]. Unfortunately, there are many social and technical barriers that make joining OSS projects daunting. In a literature survey, Steinmacher et al. found 15 barriers to entering OSS projects [2]. Others have also found that developers have difficulties finding appropriate tasks to start with [1], [3]–[5], and finding mentors or experts to help if needed [6]–[8].

The Mozilla Corporation depends on volunteer contributions for its continued success and is aware of the problems faced by new contributors. Mozilla has created several long-running onboarding programs to help ease developers into their development tools and processes. The primary two onboarding programs are called *Good First Bugs (GFB)* and *mentored bugs* [9]. GFBs are bugs that have been identified as easy to fix by current developers, often requiring a change to a single line of code. These make ideal first contributions as they are small isolated changes that minimize the effort needed to make the change while still requiring the contributor to set up a complete development environment. Mentored bugs may be more challenging but have been sponsored by an experienced developer to help guide newcomers through the process of fixing the bug. These bugs can be found by searching Bugzilla (Mozilla’s bug tracker) as well as through two sites dedicated

to helping surface these kinds of issues in a way that considers the expertise and interests of potential contributors.<sup>1</sup>

While tagging ‘easy’ bugs and providing mentorship programs are widely-adopted strategies for improving the onboarding processes, the effectiveness of these strategies in aiding new contributors has not been evaluated. Only one past study has evaluated the effectiveness of onboarding programs in OSS; this study involved graduate students receiving extensive mentoring (a one week mentored hackathon with continued support thereafter) during a 16 week university course [10]. Due to the nature of the onboarding support and the extrinsic motivation of completing a course requirement, these newcomers cannot be compared to the more typical volunteer contributors OSS projects are trying to attract.

In this paper we investigate the effectiveness of onboarding programs through the following research questions:

- RQ1 Are new developers who make use of onboarding programs more likely to have a successful first contribution than those who do not use these programs?
- RQ2 Are developers who use onboarding programs more likely to continue contributing after their initial attempt?
- RQ3 How do successful OSS contributors whose first experience was an onboarding program reflect on the onboarding process?

To validate these questions, we performed a quantitative study of 13 years of Mozilla contributions. During this time, 75 new contributors entered (i.e., made their first contribution attempt) the Mozilla community through the GFB program, 193 through mentored programs, and 220 through bugs assigned to both programs, while 3,070 developers entered through issues that were not involved in onboarding programs. In general, we found that onboarding programs do not seem to increase the likelihood of newcomers becoming frequent committers, although some committers who enter through these programs *do* end up being successful over the long term.

To gain further insight into these successes, we surveyed 35 developers who made 10 or more contributions after starting in a program about their onboarding experience. We found that newcomers truly appreciate positive feedback and that making a first contribution is hard, even for bugs that are marked as GFB or are mentored. This suggests that these programs are valuable, even if they do not seem to be increasing the likelihood of developers becoming long-term contributors.

<sup>1</sup><http://codefirefox.com/>, <http://www.joshmatthews.net/bugzahoy/>

While these developers did find aspects of the onboarding process beneficial to their success, we cannot know whether they would have been successful even without the program.

Section II quantitatively examines the impact of onboarding programs. Section III describes a survey that looked into successful developers’ perceptions of these programs. Section IV discusses the threats to our study and possible explanations for the unexpected outcome of our quantitative study, while Section V concludes.

## II. QUANTITATIVE STUDY

We chose to study Mozilla’s onboarding programs because of the importance they place on volunteer contributions.<sup>2</sup> Mozilla is also the only organization we found that has multiple onboarding programs (both GFB and mentored bug programs) with publicly available data. While Gnome has a mentorship program, they do not assign mentors to bugs in their issue tracker.

We studied three of the most active Mozilla products: Core, Firefox, and Firefox OS. The Core product is made up of a number of shared components that are used by Firefox and other Mozilla products. Firefox bugs related to the user interface, bookmarks, and developer tools are filed against the Firefox product in Bugzilla. Firefox OS is an operating system based on the Linux kernel and is one of Mozilla’s most quickly growing products. We downloaded the bugs and attachment data of these products using the Bugzilla REST API; the characteristics of this dataset are shown in Table I.

TABLE I  
NUMBERS OF BUGS STUDIED FROM EACH ONBOARDING PROGRAM FOR THE THREE EXAMINED MOZILLA PRODUCTS BETWEEN MAY 23, 2001 AND JUNE 13, 2014.

	Core	Firefox	FirefoxOS	Combined
GFB	229	124	32	385
Mentored	560	202	208	970
Mentored & GFB	440	324	127	891
No Program	216,992	137,925	38,184	393,101
Combined	218,221	138,575	38,551	395,347

We define a ‘contribution attempt’ as attaching a patch to a bug in Bugzilla. Attaching multiple patches to one bug is considered a single attempt. A contribution is considered successful if at least one of the author’s patches for a particular bug is not marked obsolete, the bug resolution is `FIXED`, and the bug status is `VERIFIED` or `RESOLVED`. If multiple authors have successful patches on a single bug all authors are considered to have made a successful contribution. A failed contribution occurs when an author has submitted one or more patches to a bug whose resolution is `FIXED`, has a status of `VERIFIED` or `RESOLVED`, and *all* of the patches are obsolete. Bugs with other statuses (e.g. `WONTFIX`, `INVALID`) were not counted as contribution failures since the newcomer may not be to blame. `OPEN` bugs were also excluded.

<sup>2</sup><https://wiki.mozilla.org/Contribute>

We excluded contributors with email addresses containing the string `@mozilla` from the study, as these individuals are already Mozilla contributors. A surge of contributors joined the onboarding programs during 2014. To allow newcomers sufficient time to become effective contributors, we excluded those developers who made their first contribution near the end of our study. From the data, we found that newcomers who make 10 contributions take an average of 203 days to complete this work; as such, we excluded any developer whose first contribution occurred during the last 203 days of our study. We identified a GFB by the presence of the string `good first bug` in the whiteboard field<sup>3</sup> and a mentored bug by the string `mentor` in the whiteboard field (or by a mentor being assigned to the `mentors` field); this is the standard mechanism Mozilla uses for labelling such bugs.

To examine RQ1 (do onboarding programs increase the likelihood of success for the first contribution?), we examined the number of contributors who made their first contribution attempt through an onboarding program, as well as their success rate; this data is shown in Figure 1. We found that a newcomer’s first attempt is more likely to succeed if it is on a bug that is mentored or on a bug that is both mentored and a GFB than on bugs that are not in any program. On the other hand, first attempts on bugs that are only labeled as GFB are less likely to succeed than attempts on bugs that are not in any program. We also examined bugs that are mentored or labeled GFB (i.e., inclusive or), but do not include the results because they did not provide additional insights.

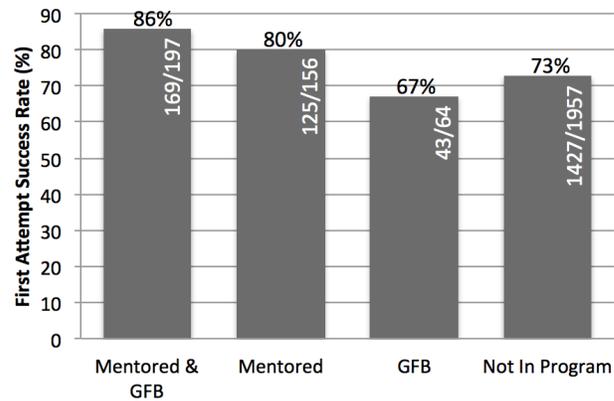


Fig. 1. RQ1: The likelihood that a developer’s first contribution is successful for each of the four measured conditions.

To answer RQ2 (are developers more likely to continue contributing if they start through an onboarding program?), we analyzed the attrition rate of contributors who started through an onboarding program with those contributors that did not; these results are shown in Figure 2.

A large percentage of contributors whose first attempt is on a mentored bug (or a bug in both programs) succeed (80.1% and 85.7% respectively). However a much smaller

<sup>3</sup>A text field for adding notes and tags to a bug.

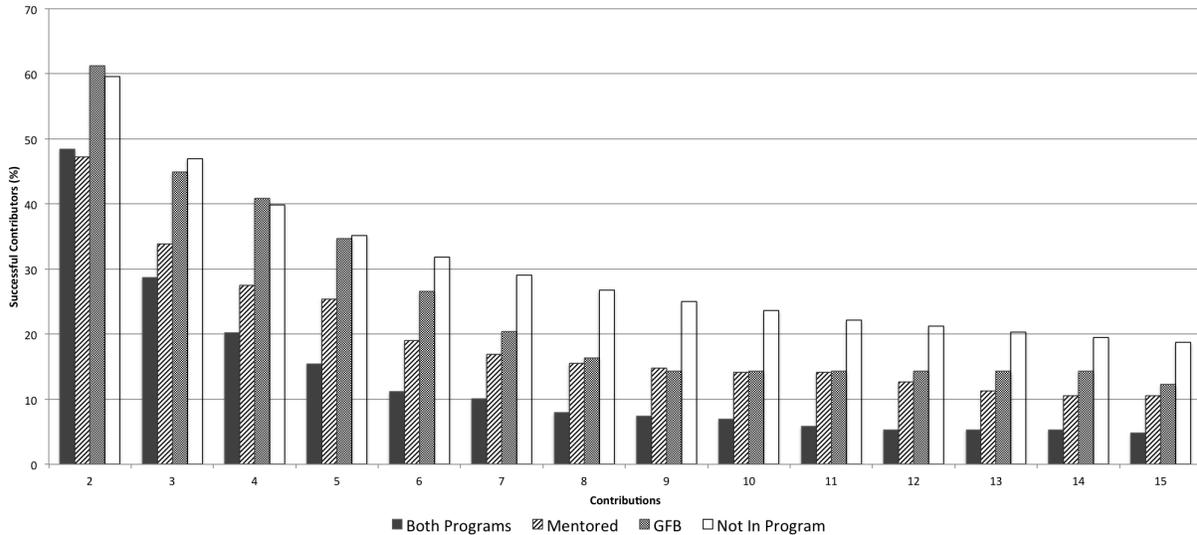


Fig. 2. The percentage of successful contributors (i.e., those who made at least one successful contribution), who made  $x$  or more contributions

percentage of successful contributors who start in this context, regardless of their success on the first attempt, make two or more contributions (47.2% and 48.4% for mentored bugs and bugs in both programs respectively). This stands in contrast with the large percentage of GFB starters (61.2%) that make two or more contributions, despite their lower success rate on the first attempt (67.1%).

The dropout rate for program participants is much higher than for developers that did not start in a program (i.e., the gap between participants and non-participants continues to widen. For example, fewer than 15% of participants make 10 or more contributions while more than 20% of non-participants make 10 or more contributions).

In summary, the data suggests that while developers in onboarding programs are more likely to succeed with their first attempts they are generally less likely to become long-term contributors.

It is important to note that we identified 35 contributors who made more than 10 contributions after starting with an onboarding program; these are clearly the kinds of developers these programs want to engage. To better understand the role of onboarding programs in these individuals' success, we contacted them to gain insight into their onboarding experience.

### III. QUALITATIVE STUDY

While the quantitative study casts a shadow on the influence of onboarding programs on long-term participation, we sought feedback from successful long-term contributors who joined Mozilla through these programs to gain insight from their experience. In this survey we focused on developers who ended up being successful; prior research has focused on reasons why developers drop out of OSS projects finding that inadequate answers to questions and difficulty finding a place to start may influence their decision to abandon a project [11].

Our questionnaire included three questions:

- 1) Your first patch was on a bug tagged as 'Good First Bug' (GFB) or 'mentored'; do you recall how you selected this bug?
- 2) Did starting with a GFB/mentored bug influence your perception of the contribution you were making?
- 3) Do you have any other recollection of your onboarding experience, or any reflections about why you have chosen to make so many valuable contributions to Mozilla?

11 of the 35 (31%) contributors surveyed responded. We performed an open-coding card sort with the survey responses; each card consisted of a single sentence. The two coders achieved an intercoder reliability score (percent agreement) of 96%. Four main categories emerged from the data: Mentorship, Motivation, Starting Points, and Complaints.

The primary subcategories under Mentorship were *guidance* ("[mentorship] certainly helped me not get lost"), *positive feedback* ("small words like 'great', 'please', 'thanks' etc. are really motivating for first timer"), and *comfort* ("I think [mentorship] certainly made the process less scary"). In general, it was clear that mentors played an important role in helping the developers feel like they were making valued contributions.

Respondents revealed that they were motivated in three different ways: *Mozilla's values* ("I contribute to Mozilla because I value a free and open internet"), *learning Mozilla's processes* ("The point wasn't to make a valuable contribution but to set up my development environment"), and *gaining development experience* ("I wanted some programming experience on a big project"). While it was clear that many developers were motivated to contribute specifically to Mozilla, several respondents made it clear that they just wanted to improve their development experience by working on a large project.

Determining where to start is more challenging when considering an initial contribution than for subsequent tasks. Developers referenced BugsAhoy, Bugzilla search, IRC, Code-

Firefox, and Bugmail as sources they used to plan their first contribution.

Developers faced several challenges working on their first contribution. The first of these could be considered *barriers*: these include a lack of documentation and problems finding a place to start (e.g., “Most of the bugs [on BugsAhoy] are assigned to people, though many have stalled”). Another challenge was dealing with the *perception of contribution size* (“Sometimes, you even get frustrated by getting such a simple patch wrong”); it was clear from several respondents that the ratio of the effort required to fix a GFB compared to its size and overall impact on the Mozilla project was much larger than new contributors expected.

#### IV. DISCUSSION

In this Section we discuss possible threats to our analysis and offer some reasonable scenarios that may be relevant to our results.

##### A. Threats to Validity

There are several threats that may have influenced the validity of our quantitative study as well as the analysis of the responses to our survey.

Internal validity refers to the extent to which the causal relationships described in the study are accurate, in other words, the extent to which the confounding variables are controlled. We identified two such threats. The first is that contributors were identified by the email address recorded with their patch submissions. It is possible that an experienced contributor changed their email address and could therefore have been erroneously counted as making a first contribution in an onboarding program. This would likely bias our results in favour of onboarding programs. The opposite effect is also possible where a contributor who started in a program changes their email address after making one contribution and then continues to make contributions under this new email outside of an onboarding program. The future contributions of this individual can no longer be tied to an individual who entered through an onboarding program. The second confounding factor is that some developers may have been paid for their work. Of the 11 developers who responded to our survey, two were paid, one joined Mozilla after making some unknown number of contributions, and one started an internship after making more than ten contributions.

External validity refers to the extent to which the results of a study can be generalized. We only studied onboarding programs of the Mozilla Corporation and gathered data from three of their products. This may not be a representative sample of mentoring and good first bug programs in general and could limit the generalizability of our results.

Construct validity is the degree to which our analysis measures what it claims to measure. It may be that newcomers that enter through the studied onboarding programs continue to be actively involved in the project in other ways, such as reporting bugs, and that the programs therefore have great value in unexpected ways. As the main purpose of the onboarding

programs we studied is to encourage code contributions, this informed the definitions we used for contributions and success in our study.

##### B. Contribution Scenarios

Our quantitative results reveal that although onboarding programs increase the chance that a new contributors’ first attempt succeeds, they do not increase their chances of making subsequent contributions. This section explores possible reasons for this phenomenon.

One possible reason for the relatively steep dropout curve from mentored programs might be that these programs attract individuals who would not otherwise have attempted to make a contribution at all. In other words, while assisting these developers certainly increases their odds of succeeding with an initial contribution, they may be less equipped to transition into long-term contributors. In contrast, developers who enter through other means may have a strong intrinsic motivation to fix the bugs they select instead of relying on mentorship.

One concern is that some GFB entrants seem to be shocked and discouraged by how difficult it is to fix even the simplest of bugs. From the responses to our survey, we gathered that successful contributors also found the development process difficult and possibly felt self conscious “at first, it felt really odd to contribute by changing a single line of code.” This suggests that downplaying the difficulty of tackling a GFB may in fact increase the chances of failure since the expectation may be that these bugs should be ‘easy’.

The final surprising point is that although contributors who start with GFBs are more likely to fail on their first attempt, they are the most likely of all newcomers to make a second contribution. We speculate that this is because fixing a GFB is in fact much harder than fixing a mentored issue since the developers are more self-reliant; this initial filter decreases their chances of a successful first contribution but creates a survival bias for subsequent contributions. That said, by the fifth contribution developers who did not enter through an onboarding program are more likely to continue contributing than even those who were successful on their first GFB.

#### V. CONCLUSION

OSS onboarding programs aim to make it easier for new developers to become long-term contributors. Our analysis of several years of Mozilla onboarding shows that developers whose initial contribution is on a GFB or mentored bug are less likely to become long-term contributors; however, *some* of these developers do end up making a significant number of contributions. While it is unclear whether these successful contributors would have started contributing without these programs, this paper provides quantitative evidence that these programs alone do not automatically improve the odds of a new developer becoming a long-term contributor.

#### ACKNOWLEDGMENT

The authors would like to thank the anonymous survey respondents for their view of their onboarding experience.

## REFERENCES

- [1] Y. Park and C. Jensen, "Beyond pretty pictures: Examining the benefits of code visualization for open source newcomers," in *Proceedings of the International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT)*, 2009, pp. 3–10.
- [2] I. Steinmacher, M. A. G. Silva, and M. A. Gerosa, "Barriers faced by newcomers to open source projects: a systematic review," in *Open Source Software: Mobile Open Source Technologies*. Springer, 2014, pp. 153–163.
- [3] X. Ben, S. Beijun, and Y. Weicheng, "Mining developer contribution in open source software using visualization techniques," in *Proceedings of the International Conference on Intelligent System Design and Engineering Applications*, 2013, pp. 934–937.
- [4] A. Capiluppi and M. Michlmayr, "From the cathedral to the bazaar: An empirical study of the lifecycle of volunteer community projects," in *Open Source Development, Adoption and Innovation*. Springer, 2007, pp. 31–44.
- [5] G. V. Krogh, S. Spaeth, and K. R. Lakhani, "Community, joining, and specialization in open source software innovation: A case study," *Research Policy*, vol. 32, no. 7, pp. 1217–1241, 2003.
- [6] G. Canfora, M. Di Penta, R. Oliveto, and S. Panichella, "Who is going to mentor newcomers in open source projects?" in *Proceedings of the International Symposium on the Foundations of Software Engineering (FSE)*, 2012, pp. 44:1–44:11.
- [7] D. Cubranic, G. C. Murphy, J. Singer, and K. S. Booth, "Hipikat: A project memory for software development," *IEEE Transactions on Software Engineering (TSE)*, vol. 31, no. 6, pp. 446–465, 2005.
- [8] I. Steinmacher, I. S. Wiese, and M. A. Gerosa, "Recommending mentors to software project newcomers," in *Proceedings of the International Workshop on Recommendation Systems for Software Engineering (RSSE)*, 2012, pp. 63–67.
- [9] "Contributing to the mozilla codebase," [<https://developer.mozilla.org/en/docs/Introduction>], accessed: 2015-02-09.
- [10] F. Fagerholm, A. S. Guinea, J. Münch, and J. Borenstein, "The role of mentoring and project characteristics for onboarding in open source software projects," in *Proceedings of the International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2014, p. 55.
- [11] I. Steinmacher, I. Wiese, A. P. Chaves, and M. A. Gerosa, "Why do newcomers abandon open source software projects?" in *Proceedings of the International Workshop on the Cooperative and Human Aspects of Software Engineering (CHASE)*, 2013, pp. 25–32.