# Onboarding to Open Source Projects with Good First Issues: A Preliminary Analysis

Hyuga Horiguchi
*Graduate School of Systems Engineering*
*Wakayama University*
Wakayama, Japan
hhyuga201515@gmail.com

Itsuki Omori
*Faculty of Systems Engineering*
*Wakayama University*
Wakayama, Japan
omori.itsuki@g.wakayama-u.jp

Masao Ohira
*Faculty of Systems Engineering*
*Wakayama University*
Wakayama, Japan
masao@wakayama-u.ac.jp

***Abstract*—While open source software (OSS) development projects always seek contributions from newcomers to make the projects sustainable, newcomers often face with a challenge of onboarding. For developers with little experience, it is difficult to decide which of the many OSS projects to contribute to and which issues to tackle. To mitigate the barrier against onboarding, some OSS projects start using a label so called *good first issue* (GFI) which indicates the issue is easy to resolve and suitable for newcomers to tackle. The final goal of this study is to construct a model for recommending good first issues to help reduce maintainers' manual effort of selecting and labeling issues as GFIs and at the same time to help find GFIs suitable for each newcomer. Toward the final goal, we analyze GFIs in GitHub to deeply understand the current state of the GFI mechanism and its impact on new members' onboarding. This paper reports a result of our preliminary analysis of GFIs, based on 9,475 GFIs collected from 11 famous OSS projects in GitHub. We find that (1) developers who have resolved GFIs have 92 fewer median PR (pull request) posts than developers who have resolved regular issues (i.e., developers tackling GFIs have less experience, compared to other developers), (2) on average, GFIs are resolved 32.5% more than regular issues (i.e., GFIs are easier to resolve. The dataset of GFIs might be able to be used as a training set for recommending GFIs to developers with less experience), and however (3) the percentage of developers who keep contributing to the same project even after resolving GFIs varies greatly from project to project (24.9% to 83.9%) (i.e, GFIs guide newcomers' onboarding only in particular projects). Based on the analysis result, we discuss the contents of GFIs which tend to keep newcomers on OSS projects.**

***Index Terms*—Good First Issues, Newcomers, Onboarding, Open Source Software**

## I. INTRODUCTION

Common open source software (OSS) projects heavily rely on contributions such as bug reports, bug fixes, and feature enhancements from volunteer developers [1]. Since volunteer developers are free to participate in the development and also free to leave, OSS projects always seek newcomers and their contributions to make the projects sustainable [2] [3]. One of the motivations for newcomers to contribute to OSS projects is to improve their careers and skills through participating in the OSS development [4], but they often face the difficulty in deciding which project to contribute [5]. A previous study [6] has proposed a model to recommend OSS projects suitable for developers, based on the developers' past development activities. However, even if newcomers find interesting projects, they need to take much time to learn how to address unfamiliar, various tasks in the projects [7].

To mitigate the barrier against newcomers' onboarding to projects, some OSS projects start using a label so called "*good first issue* (GFI)[1]" to provide newcomers with an opportunity to contribute to the projects. Although the GFI mechanism seems to be effective in lowering the barrier for newcomers to participate in OSS projects, it is pointed out that the effort of manually selecting and labeling issues as GFIs will be a burden on maintainers in the projects[2].

The recent study [8] on GFIs in GitHub revealed that the number of projects using the GFI mechanism have been increasing in the last ten years. It also found that 40.9% of GFIs have not been resolved by newcomers and the GFI mechanism does not work well for attracting contributors for a long term. While Tan *et al.* [8] analyzed 9,368 GFIs from 816 GitHub projects using GFIs to provide a whole picture of the GFI mechanism in GitHub, in this paper we use 9,475 GFIs collected from 11 famous GitHub projects to intensively analyze the current state of the GFI mechanism in the top projects which actively utilize GFIs and to understand its impact on newcomers' onboarding.

The final goal of this study is to construct a model for recommending good first issues to help reduce maintainers' effort of selecting and labeling issues as GFIs and at the same time to help find GFIs suitable for newcomers in the context of supporting onboarding. Toward the final goal, in this paper we address the following research questions.

**RQ1:** *Are good first issues suitable for inexperienced developers?* A good first issue should be labeled for newcomers. Labeling issues are done by maintainers manually, but it is not sure if good first issues are actually easy to be tackled by newcomers. Tan *et al.* [8] defined a developer with less than three commits to a particular project as a newcomer and found that 59.1% of GFIs were resolved by newcomers. By this definition of a newcomer, even if s/he had made many contributions to other projects in the past (i.e., even if s/he is an experienced developer), s/he may be included as a newcomer. Instead of defining a newcomer by the number

[1]https://docs.github.com/en/free-pro-team@latest/github/managing-your-work-on-github/about-labels

[2]https://github.blog/2020-01-22-how-we-built-good-first-issues/

of contributions (commits) to a particular project, we analyze each developer's contributions across GitHub. By comparing the past contributions of developers who resolved GFIs and developers who resolved regular issues, we confirm if GFIs can be resolved by developers with less experience.

**RQ2:** *Are good first issues easier to resolve than regular issues?* In [8], the qualitative analysis (N=164) revealed that 59.1% of GFIs were resolved by newcomers. However, it is still unclear whether GFIs are easier to tackle than regular issues, as [8] had not compared the resolution rate of GFIs and regular issues. In addition, we assume that some projects make good use of GFIs and some projects do not. It would be worth of comparing those projects to obtain the best practices for utilizing GFIs. In RQ2, we analyze the resolution rate of GFIs and regular issues for each project.

**RQ3:** *What is the percentage of developers who posted a pull request after resolving a good first issue?* The GFI mechanism expects newcomers to contribute to projects for a long time, by accustoming OSS development through the process in resolving GFIs. Therefore, in order to evaluate the usefulness of the GFI mechanism, it is important to know whether or not newcomers can address regular issues after resolving GFIs. Although Tan *et al.* [8] found that 58.4% of newcomers who resolved a good first issue left the projects, it did not analyze each project. For each project, we analyze the percentage of developers who posted a pull request (PR) to resolve a regular issue after resolving a good first issue.

The contributions of this paper are as follows.

- We confirmed GFIs are easier to be tackled by newcomers with less development experience than regular issues (RQ1 and RQ2). It indicates that GFIs could be used as training data to build a model for recommending issues suitable to newcomers.
- We also confirmed that the GFI mechanism does not always work to support newcomers' onboarding in all projects but newcomers' onboarding is likely guided depending on the contents of GFIs (RQ3). It suggests that the training data should be carefully screened depending on the contents of GFIs before building an issue recommendation model.

The rest of the paper is organized as follows. Section II introduces related work and positions our study. Section III describes the dataset created for our analysis. Section IV reports a result of our analysis and answers our research questions. Section V discusses the analysis result in depth, in particular for RQ3. Section VI remarks the conclusions and our future work.

## II. Related Work

Previous studies [7] [9] [10] have shown that newcomers face barriers to participate in OSS projects and that results in leaving the projects. Supporting newcomers to overcome the barriers are required for onboarding to OSS projects. In order to support newcomers' onboarding, several studies have proposed the use of a mentoring system [11] [12], easy bug resolution tasks [8] [13], and a portal site for newcomers. Although these approaches to support onboarding have certain effect, many OSS projects still cannot leverage the approaches due to the burden on maintainers (i.e., core members). We believe that automating the GFI mechanism without manual efforts of maintainers would be helpful for both newcomers and OSS projects. Through addressing the research questions (especially RQ3), in this paper we show GFIs work for newcomers' onboarding to GitHub projects.

As regards the issue recommendation approach, many studies [14] [15] have proposed algorithms to automate "bug triage" to help maintainers find appropriate developers for bug-fixing tasks. The existing bug triaging algorithms recommend bug-fixing tasks to the appropriate developers with the aim of fixing bugs quickly and correctly, based on developers' activities in the past. Newcomers often cannot judge which bug fixing tasks are suitable for them. It would be useful for newcomers' onboarding if bug triaging approaches could find tasks for newcomers. However, the existing approaches cannot be applicable or are very limited to support newcomers, because many of newcomers have less experience with OSS development and hence less activities in the past. Therefore, we need to construct a new algorithm to recommend issues to newcomers with less experience instead of using the bug triaging algorithms.

GitHub has started providing GitHub projects with the feature[3] of recommending GFIs so that newcomers can work on. The GFI recommendation feature is implemented using machine learning algorithms such as CNNs (convolutional neural networks) and RNNs (recurrent neural networks) and the contents of issues which can be regarded as GFIs labeled with *beginner friendly*, *easy bug fix* and so forth. The feature helps maintainers reduce the manual effort of labeling issues as GFIs because it is available for projects which do not have issues with the GFI-related labels. However, previous studies [8] [13] on GFIs concluded that the GFI mechanism was less effective to motivate newcomers to be long-term contributors. It might imply that GitHub's model can recommend GFIs suitable for newcomers but cannot help them keep contributing for long time. Based on the result of RQ3, we will discuss the contents of GFIs which will be helpful for onboarding.

## III. Dataset

This section describes our dataset created to address the research questions. We used GitHub API[4] for all data collection. Firstly, we collected issues labeled as *good first issue* from the beginning of the projects to June 16, 2020. Next, we counted the number of GFIs for each repository and excluded repositories with less than 500 GFIs. This is because we can expect that repositories with a large number of GFIs involve more newcomers. As a result of the filtering, 11 GitHub projects remained for the analysis. Next, we collected all the regular issues except for GFIs from the targeted 11 repositories. As with the collection of GFIs, the regular issues were collected

[3]https://github.blog/2020-01-22-how-we-built-good-first-issues/

[4]https://developer.github.com/v4/

TABLE I
DATASET OVERVIEW

| Repository | Age (years) | Num. of GFIs | Num. of PRs | Num. of resolved issues | Num. of resolved GFIs | Num. of devs | Num. of devs who resolved GFIs |
|---|---|---|---|---|---|---|---|
| osmlab/name-suggestion-index | 6 | 1,464 | 1,635 | 811 | 639 | 107 | 65 |
| radareorg/radare2 | 7 | 1,169 | 9,252 | 827 | 158 | 210 | 85 |
| zulip/zulip | 4 | 1,030 | 10,046 | 495 | 109 | 120 | 51 |
| mui-org/material-ui | 5 | 982 | 10,080 | 3,279 | 912 | 935 | 542 |
| pandas-dev/pandas | 9 | 948 | 16,596 | 5,436 | 638 | 1,067 | 316 |
| CleverRaven/Cataclysm-DDA | 7 | 880 | 26,634 | 6,148 | 630 | 566 | 228 |
| elastic/elasticsearch | 10 | 623 | 34,063 | 6,363 | 344 | 366 | 136 |
| rust-lang/rust-clippy | 5 | 610 | 2,868 | 1,051 | 344 | 223 | 164 |
| tgstation/tgstation | 8 | 601 | 34,190 | 8,306 | 484 | 417 | 143 |
| WordPress/gutenberg | 3 | 587 | 11,505 | 3,165 | 362 | 292 | 154 |
| elastic/kibana | 7 | 581 | 43,969 | 7,492 | 350 | 292 | 102 |
| total |  | 9,475 | 200,838 | 43,373 | 4,970 | 4,595 | 1,986 |

until June 16, 2020. Next, we examined whether the issues were resolved. We defined an issue satisfying the following conditions as a resolved issue.

1) The issue is closed.
2) The issue is mentioned from a pull request (PR) which includes mentions such as #1 and #3 in the body text.
3) The PR in 2) was merged.

Finally, we counted the number of GFIs, regular issues, PRs, and developers who resolved GFIs and/or regular issues. Table I shows an overview of our dataset[5].

## IV. RESULTS

This section describes our analysis on the GFI mechanism and its impact on newcomers' onboarding to OSS projects.

### A. Developers' Experience with OSS Development

We first try to answer the following research questions. **RQ1:** *Are good first issues suitable for inexperienced developers?*

**Motivation:** Tan *et al.* [8] defined a developer with less than three commits to a particular project as a newcomer. By this definition of a newcomer, even if s/he had made many contributions to other projects in the past, s/he is regarded as a newcomer. We analyze each developer's contributions across GitHub. Comparing the past contributions of developers who resolved GFIs and developers who resolved regular issues, we confirm if GFIs can be resolved by developers with less experience.

**Approach:** We count the total number of pull requests (PRs) posted by each developer before resolving a GFI or a regular issue in one of the target repositories (projects). Note that the total number of PRs means contributions to other external repositories (i.e., other GitHub projects except for the target projects) and that we excluded PRs to own repositories and/or own organization's repositories because those PRs are not considered as contributions to external repositories. We then divide into a group of developers who have resolved a GFI and a group of developers who have resolved a regular issue, and compare the distributions of PRs between the two groups with a violin plot. We also examine a statistically significant difference between the distributions with the Mann-Whitney $U$ test.

[5]The dataset will be available from http://floss-lab.org/?p=1812.

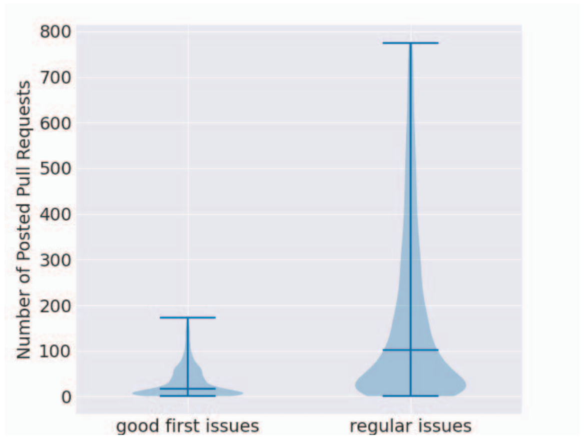

Fig. 1. The number of pull requests (PRs) posted by developers before resolving a GFI (left) and a regular issue (right) in the target projects. The number of PRs represents an approximation of developer's experience before contributing to the target projects.

**Result:** Figure 1 shows a result of a comparison of the distributions of PRs after eliminating outliers. We can see a clear difference between the two distributions. The median number of PRs posted by developers before resolving a GFI was 26 while the median number of PRs posted by developers before resolving a regular issue was 118. We also confirmed a statistically significant difference ($p < 0.010$) and a large effect size (Cliff's $d = 0.655 \geq 0.474$). From the result, we can answer RQ1 as follows.

Answer to RQ1:

Compared to developers who resolved regular issues, developers who resolved GFIs have less experience with PRs (i.e., OSS development). In other words, GFIs are more suitable for newcomers to tackle with.

TABLE II
THE RESOLUTION RATE OF REGULAR ISSUES AND GOOD FIRST ISSUES.

| repository | % of resolved regular issues | % of resolved GFIs |
|---|---|---|
| **mui-org/material-ui** | 21.0 | **92.9** |
| **tgstation/tgstation** | 45.0 | **80.5** |
| CleverRaven/Cataclysm-DDA | 37.6 | 71.6 |
| pandas-dev/pandas | 26.3 | 67.3 |
| WordPress/gutenberg | 24.1 | 61.7 |
| elastic/kibana | 28.2 | 60.2 |
| rust-lang/rust-clippy | 24.8 | 56.4 |
| elastic/elasticsearch | 25.1 | 55.2 |
| osmlab/name-suggestion-index | 7.3 | 43.6 |
| **radareorg/radare2** | **9.0** | **13.5** |
| **zulip/zulip** | **7.2** | **10.6** |
| average | 23.2 | 55.8 |

## B. *Easiness of good first issues*

Next, we aim to answer the following research question.
**RQ2:** *Are good first issues easier to resolve than regular issues?*
**Motivation:** In [8], the qualitative analysis (N=164) revealed that 59.1% of GFIs were resolved. However, it is still unclear whether GFIs are easier to tackle than regular issues. In addition, we assume that some projects make good use of GFIs and some projects do not. It would be worth of comparing those projects to obtain the best practices for utilizing the GFI mechanism. We analyze the resolution rate of GFIs and regular issues for each project.
**Approach:** We calculate the the resolution rate of GFIs and regular issues for each project and compare them between the target repositories.
**Result:** Table II shows a result of calculating the resolution rate which are sorted in descending order of the resolution rate for GFIs. For all the target projects, the resolution rate of GFIs is higher than that of regular issues. The resolution rate of GFIs is 55.8% on average, which is almost consistent with the analysis by Tan *et al.* [8]. On the other hand, the resolution rate of regular issues is much lower (23.2% on average) than that of GFIs. The result indicates GFIs are easier to tackle than regular issues.

However, we can also see the large differences of the resolution rate between the target projects. For instance, 92.9% and 80.5% of GFIs in the projects of mui-org/material-ui and tgstation/tgstation were resolved respectively while it is only 13.5% and 10.6% in radareorg/radare2 and zulip/zulip respectively. Furthermore, radareorg/radare2 and zulip/zulip also showed the low resolution rate of regular issues (9.0% and 7.2% respectively). These projects might be vigilant against accepting the contributions for resolving both GFIs and regular issues. In summary, we answer RQ2 as follows.

Answer to RQ2:

Good first issues are easier to tackle than regular issues. However, some projects may not actively accept contributions for resolving GFIs.

TABLE III
PERCENTAGE OF DEVELOPERS WHO POSTED A PULL REQUEST TO THE SAME PROJECT AFTER RESOLVING A GOOD FIRST ISSUE

| repository | % of developers |
|---|---|
| **tgstation/tgstation** | **83.9** |
| elastic/kibana | 80.4 |
| zulip/zulip | 80.4 |
| CleverRaven/Cataclysm-DDA | 70.6 |
| radareorg/radare2 | 63.5 |
| WordPress/gutenberg | 62.3 |
| elastic/elasticsearch | 60.3 |
| rust-lang/rust-clippy | 40.9 |
| pandas-dev/pandas | 35.8 |
| osmlab/name-suggestion-index | 33.8 |
| **mui-org/material-ui** | **24.9** |

## C. *Newcomers' Onboarding*

Finally, we analyze the percentage of developers who posted a PR to the same project after resolving a GFI.
**RQ3:** *What is the percentage of developers who posted a pull request after resolving a good first issue?*
**Motivation:** The GFI mechanism expects newcomers to contribute to projects for a long time. It is important to evaluate whether or not newcomers can address regular issues after resolving GFIs. Although Tan *et al.* [8] found that 58.4% of newcomers who resolved a GFI left the projects, it did not analyze each project. As with RQ2, we suspect that the rate of leaving projects varies largely from project to project.
**Approach:** For each project, we analyze the percentage of developers who posted PRs to resolve a regular issue after resolving a GFI.
**Result:** Table III shows an analysis result. It is sorted in descending order of the percentage of developers who posted PRs after resolving a GFI. As we expected, whether developers keep contributing to the same project is largely different between the projects. Some projects successfully guide contributions for GFIs to contributions for regular issues (i.e., newcomers' onboarding) and some project does not.

In addition, we found that the percentage of developers posting PRs after resolving a GFI is not necessarily high even if GFIs are easy to resolve. For instance, although tgstation/tgstation has both the high resolution rate of GFIs (80.5% in Table II) and the high percentage of developers keep contributing (83.9% in Table III), mui-org/material-ui has the highest resolution rate of GFIs (92.9% in Table II) and the lowest percentage of developers keep contributing (24.9% in Table III). This result indicates that just preparing GFIs does not mean that newcomers will continue to contribute to the project. In order to further understand what kind of GFIs motivates newcomers to keep contributions, we discuss an additional analysis in section V.

Answer to RQ3:

The percentage of developers keep contributing to the same project is largely different between the projects. Just preparing GFIs cannot lead to newcomers' onboarding.

TABLE IV
TOP FIVE LABELS CO-OCCURRING WITH "GOOD FIRST ISSUE"

| mui-org/material-ui | | tgstation/tgstation | |
|---|---|---|---|
| label | % of GFIs | label | % of GFIs |
| bug | 40.9 | Bug | 70.7 |
| docs | 23.5 | Mapping | 16.8 |
| enhancement | 19.2 | Grammar and Formatting | 14.8 |
| component: Autocomplete | 8.6 | Sprites | 10.3 |
| typescript | 8.2 | Not a bug | 8.7 |

## V. DISCUSSION

We discuss the reason why % of developers keeping contributions was largely different between the projects in RQ3.

### A. Successful or Unsuccessful Onboarding?

To find clues to reveal the reason for successful or unsuccessful onboarding through resolving GFIs, we conducted an additional analysis of tgstation/tgstation which showed the highest percentage of developers keep contributing and mui-org/material-ui which showed the lowest percentage. In the additional analysis, we tried to understand the contents of GFIs in the two projects. We collected all the labels which were labeled together with the *good first issue* label and calculated the percentage of each label.

Table IV shows the top five labels co-occurring with *good first issue* in the two projects. For both of the projects, the top one label co-occurring with *good first issue* was related to bugs (*bug* in material-ui and *Bug* in tgstation). Especially in tgstation, 70.7% of issues labeled as *good first issue* also had the bug-related label. It may suggest that GFIs for resolving bugs encourage newcomers to keep contributing to the project after resolving GFIs.

Furthermore, the second top label (23.5%) in material-ui was related to documentations *docs* while only 0.5% (3 of 601 issues was labeled as *good first issue* ) in tgstation was the document-related label. It indicates that document-related GFIs might not work as a catalyst for motivating newcomers (i.e., for onboarding). Tan *et al.* [8] also revealed that many of resolved GFIs are involved with documentations. Although documentation tasks are very important for both users and developers to keep documents up to date, they are unlikely to be useful for newcomers onboarding. These findings from the additional analysis would be helpful to screen training data when constructing a model for recommending issues which have an effect on newcomers' onboarding.

### B. Threats to Validity

In this study, we analyzed the famous 11 GitHub projects with the highest number of GFIs, but there are other projects using GFIs. The trend of the GFI mechanism in small projects may not be captured by this study. However, we think that the overall tendency of the GFI mechanism was grasped because projects having a number of GFIs also have many developers working on GFIs.

## VI. CONCLUSION AND FUTURE WORK

This paper reported a result of our preliminary analysis of GFIs, based on 9,475 GFIs collected from 11 famous OSS projects in GitHub. From the analysis to address our research questions, we gained the following insights:

- We confirmed GFIs are easier to be tackled by newcomers with less development experience than regular issues (RQ1 and RQ2). It indicates that GFIs could be used as training data to build a model for recommending issues suitable to newcomers.
- We also confirmed that the GFI mechanism does not always work to support newcomers' onboarding in all projects but it is likely guided depending on the contents of GFIs (RQ3). It suggests that the training data should be carefully screened depending on the contents of GFIs.

We will systematically analyze the contents of GFIs which are effective to support onboarding and build an issue recommendation model based on the analysis results.

## ACKNOWLEDGMENT

This research is conducted as part of Grant-in-Aid for Japan Society for the Promotion of Science (JSPS) Research Fellow and Scientific Research JP18K11243.

## REFERENCES

[1] K. Crowston, H. Annabi, and J. Howison, "Defining open source software project success," in *ICIS '03*, 2003, pp. 327–340.

[2] I. Qureshi and Y. Fang, "Socialization in open source software projects: A growth mixture modeling approach," *Organizational Research Methods*, pp. 208–238, 2011.

[3] A. Forte and C. Lampe, "Defining, understanding, and supporting open collaboration: Lessons from the literature," *American Behavioral Scientist*, vol. 57, no. 5, pp. 535–547, 2013.

[4] A. Lee, J. C. Carver, and A. Bosu, "Understanding the Impressions, Motivations, and Barriers of One Time Code Contributors to FLOSS Projects: A Survey," *ICSE '17*, pp. 187–197, 2017.

[5] A. Capiluppi and M. Michlmayr, "From the cathedral to the bazaar: An empirical study of the lifecycle of volunteer community projects," in *OSS '07*, 2007, pp. 31–44.

[6] C. Liu, D. Yang, X. Zhang, B. Ray, and M. M. Rahman, "Recommending github projects for developer onboarding," *IEEE Access*, pp. 52 082–52 094, 2018.

[7] I. Steinmacher, T. U. Conte, and M. A. Gerosa, "Understanding and supporting the choice of an appropriate task to start with in open source software communities," in *HICSS '15*, 2015, pp. 5299–5308.

[8] X. Tan, M. Zhou, and Z. Sun, "A first look at good first issues on github," in *ESEC/FSE '20*, 2020, pp. 398–409.

[9] I. Steinmacher, T. Conte, M. A. Gerosa, and D. Redmiles, "Social barriers faced by newcomers placing their first contribution in open source software projects," in *CSCW '15*, 2015, pp. 1379–1392.

[10] C. Hannebauer and V. Gruhn, "On the relationship between newcomer motivations and contribution barriers in open source projects," in *OpenSym '17*, 2017, pp. 1–10.

[11] F. Fagerholm, A. Sanchez Guinea, J. Borenstein, and J. Borstler, "Onboarding in open source projects," *IEEE Software*, vol. 31, no. 6, pp. 54–61, 2014.

[12] R. Britto, D. Smite, L.-O. Damm, and J. Börstler, "Performance evolution of newcomers in large-scale distributed software projects: An industrial case study," in *ICGSE '19*, 2019, pp. 1–11.

[13] A. Labuschagne and R. Holmes, "Do onboarding programs work?" in *MSR '15*, 2015, pp. 381–385.

[14] J. Anvik, L. Hiew, and G. C. Murphy, "Who should fix this bug?" in *ICSE '06*, 2006, pp. 361–370.

[15] S.-R. Lee, M.-J. Heo, C.-G. Lee, M. Kim, and G. Jeong, "Applying deep learning based automatic bug triager to industrial projects," in *FSE '17*, 2017, pp. 926–931.