An Initial Exploration of the "Good First Issue" Label for Newcomer Developers

Alderliesten, Jan Willem Davis; Zaidman, Andy

DOI
10.1109/CHASE52884.2021.00023

Publication date
2021

Document Version
Final published version

Published in
2021 IEEE/ACM 13th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)

Citation (APA)
https://doi.org/10.1109/CHASE52884.2021.00023

Important note
To cite this publication, please use the final published version (if applicable). Please check the document version above.
Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

https://www.openaccess.nl/en/you-share-we-take-care

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.
An Initial Exploration of the “Good First Issue” Label for Newcomer Developers

Jan Willem David Alderliesten
Delft University of Technology
The Netherlands
Email: david.j.w@hotmail.com

Andy Zaidman
Delft University of Technology
The Netherlands
Email: a.e.zaidman@tudelft.nl

Abstract—In order to help newcomers to open source projects identify tasks that are suitable to them and their level of expertise within the project, issues can get the good first issue label on the GitHub platform. In this paper we report on a preliminary investigation of good first issues in terms of how they effective they are for developer onboarding and task completion. We find that, although good first issues are effective at developer onboarding, and developers perceive good first issues as being useful, changes can be made to the types of tasks suggested as good first issues to match the types of initial contributions made by newcomers1.

I. INTRODUCTION

Open-source software development largely relies on contributions made by developers on their own accord [2]. A community of open source developers typically work together asynchronously to further the state of an open source project. However, due to both technical and socio-technical complexities, it is often not easy for a new developer to join an existing open source project [3], [4], [5], [6]. That is why existing developers in these projects can recommend certain tasks by labeling them as appropriate for newcomers.

Within the GitHub ecosystem, tasks can be stored as issues and these recommended issues for newcomers can be given the label “good first issue.” These tasks are aimed to be of such a level that someone unfamiliar with the processes and technical intricacies of the particular project can be introduced to the project and gain a deeper understanding. Since these labels are given to issues by existing developers, the question arises: are good first issues good introductory tasks for new contributors?

This exploration investigates a total of 105 repositories and introduces a dataset2 consisting of first contributions per developer and tasks recommended as good first issues within the GitHub platform. Of the 301,380 issues sampled, a total of 4,792 consisted of good first issues. Additionally, 1,272 first contribution commits were sampled. At most 30 first commits and 30 good first issues were manually analyzed per repository to better understand these issues. In addition to the analysis, a survey was sent out to developers within the sample commit population, which indicated that developers find good first issues to be a useful endeavour with an average score of 70.45.

1This paper is based on the MSc thesis of J.W.D. Alderliesten [1]
2Dataset and associated analysis files are available [7].

Our work builds upon the work of Tan et al. in that we have also carried out a preliminary investigation into the types of tasks, and how these tasks relate to the experience level that developers have (i.e., prior work on GitHub projects outside of the project under consideration) [8].

II. OVERVIEW OF RESULTS

In our preliminary investigation, we have performed both a mining repositories type of study, to gauge the prevalence of good first issues, and a survey among developers to understand whether they consider good first issues as a welcome addition to stimulate developer onboarding.

A. Adoption Rate of Good First Issues

Based upon the 105 sampled repositories3, a total of 46 repositories (or 43% of our sample) contain issues labelled as good first issues. From these repositories, a total of 301,380 issues were available for sampling, of which 4,792 were found to have the label good first issue or an equivalent label, meaning the representation of good first issues within the total body of issues is 1.5%.

B. Are Good First Issues Implemented by Newcomers?

From the 4,792 good first issues that we have sampled, we have selected 858 individual issues for manual analysis. We ended up with these 858 issues by selecting at most 30 issues per repository. Of these 858 issues, 279 were implemented by a new contributor to the repository, whereas 340 good first issues were implemented by developers that had already made a contribution to that particular project. For 239 good first issues we have not found an associated pull request, e.g., because they were not yet implemented. These data are given in Table I.

C. Types of Good First Issue Tasks

When mining the project repositories, we found that the most prominent types of tasks that are being suggested by experienced developers to newcomer developers are: fixing bugs (29.2%), feature enhancements (24.2%), refactoring operations (19.9%), and creating or editing documentation (18.5%). However, when we cross-match which tasks newcomer developers actually pick up, we notice that developers recommend

3Details of these results can be found in our replication package [7].
newcomer developers to perform bug fixes more often than first contributors actually implement them. Furthermore, issues revolving around documentation that are suggested as good first issues are less frequent when compared to the number of first commits made in which a documentation change or addition is the contribution. This raises the question of whether developers utilize the documentation to obtain the necessary knowledge for follow-up contributions. To investigate this finding further, 30 additional second contributions were identified from developers that had provided an initial contribution to a project that was assigned a documentation label. From these 30 sampled first contributions by newcomer developers, 10 secondary contributions were made, indicating a 33.3% rate of documentation labels leading to a second contribution. Of these 10 secondary contributions, one secondary contribution was related to a new feature, one was related to the enhancement of a feature, one to testing, and one to refactoring. The majority of secondary contributions, however, consisted of additional documentation changes.

We have also looked into the effect of developer experience on the initial contribution that is made to a project. Our findings indicate that novice developers (<1 year of experience) favor documentation-related tasks for their initial contribution. On the other hand, the fixing of bugs and the enhancement of existing features is not influenced by the developer experience level, as we witness all newcomer experience levels to contribute to it equally.

D. Developer Perception

Our survey among developers within the sample commit population of our mining study received 22 responses. Results to highlight are that developers perceive good first issues as useful (average score of 70.5/100). However, the most important complaint that we recorded with regard to good first issues is that tasks labeled as such may not be directly connected to the reality of open source development. Developers indicate they do not join a project for the sake of contributing, but because they already identified issues within the codebase. As a result, developers indicate additional consideration should be made for issues that are within a broader range of domains and regions within the codebase. In the survey, developers also indicate that they prefer to work on bug fixes and documentation-related initial contributions (37.14% and 20% respectively), which stands in partial contrast to the mined results in which the results indicate 21.3% and 35.1%, respectively. Developers in the survey indicate that refactoring is one of their least preferred tasks with a 2.85% response rate. The majority of respondents indicate that they would utilize issue labelling to some degree, with 53.57% stating outright they would keep the recommendation system as a label. One project maintainer mentioned that they would also include the notion of effort and impact labelling.

III. SUMMARY AND OUTLOOK TOWARDS FUTURE WORK

In this paper we have carried out an initial investigation into issues on GitHub that carry the label “good first issue”; in particular, we have gauged their potential for easing developer onboarding. We have established that good first issues are being used in 43.8% of the GitHub projects that we have sampled, indicating widespread adoption. Within the projects that use good first issues, 1.5% of tasks are labeled as an issue that a newcomer could start to work on to get acquainted with the project in terms of domain and modus operandi. We have also seen indications that there is a degree of mismatch in the types of tasks that experienced developers label for newcomers. When approaching developers through a survey, we do see that good first issues are appreciated with an average score of 70.5/100.

In terms of future work, we envision that we need to: (1) carry out a deeper investigation into whether good first issues lead to one-time commits from new developers, or whether good first issues really draw in new developers. That leads to the next question, namely (2) whether developers have the intent to become regular contributors out of a specific interest or curiosity for the project, or whether they have a specific purpose in mind when contributing to a project (e.g., because they are a user of the project and they urgently require a fix or additional functionality).

REFERENCES