A Partial Replication of Onboarding vs. Diversity, Productivity, and Quality – Empirical Study of the OpenStack Ecosystem

Armstrong Foundjem *MCIS Laboratory — Queen's University, Canada*
a.foundjem@queensu.ca

***Abstract*—Open science enables researchers to verify the result of a given work and either replicate or build future works based on the available results. Thus, this artifact aims to provide the data used in our study on Onboarding at the software ecosystem-level, "Onboarding vs. Diversity, Productivity, and Quality – Empirical Study of the OpenStack Ecosystem." We used mixed-method research (Qualitative and Quantitative) to answer our research question and provide the data accompanying descriptions on using our public access data online. Except for the qualitative study, we could not provide all the data we used for confidentiality purposes. All other data are available online.**

***Index Terms*—Available, open data, open science, replication, verifiable, transparency**

I. General Description and Background

A. Qualitative Dataset

Open science is essential and facilitates studies such as replication [1], and the likes. In this artifact, we make available and provide insight into how researchers could obtain our publicly available data at [2]. This data comes in two categories. First, qualitative data originate from a two-day observational study on 72 OpenStack new contributors who are willing to join the ecosystem. We show the seating configuration ("**seating_arrangement.pdf**") on how the observed participants and mentors were assigned to each table (T1,...,T12), this configuration enabled us to randomly select participants for the think-aloud protocol (explaining a task as they perform it). We used four high-0quality professional audio-visual equipment (C#1,...,C#4) to record the entire events, which we later transcribed and analyze.

Observation Data

This data was video recorded and later transcribed. Audiences can find details on the transcription of our archived data set online in the "**1.codebook**" folder. In this folder, we have five files: including "1.day1-2_observation.pdf" and "2.technical_activities_onboarding.pdf", which are the transcribed files of the 2-days observation study. The "3.onboarding_emerging_codes_irr.pdf" file constitute emerging codes from the above mentioned files. Next, the "4.think_aloud_protocol.pdf" file was generated when the principal observer OB1 would randomly ask the 72 participants to explain a task they are performing. Then, we built our codebook "**0.Codebook_with_Examples.pdf**," which is the outcome of the qualitative coding, activity showing codes in the first column, description/rationale of the code in the second column, and examples of how we used the codes in the third column.

Codebook

To build a codebook, both coders of the qualitative data did inductive (the result of the inductive coding is available online in the first file; 1.qualitatve...csv) and deductive coding containing four rounds of inter-rater reliability (IRR). We capture these rounds in the folder "**2.irr-iterations**" (3.irr., ..., 6.irr.), and report the outcome of the IRR in the"**2.irr_onboarding.csv**" file.

The outcome of the IRR has three high-levels of qualitative codes, H1, ..., H3, with several Labels that either coder one or coder two used to indicate a (dis)agreement of a code in the text; for example, TESTIMONY1r1, ..., OVERVIEW2r2, with entering of either a 1/0 in each cell.

Affinity After the high-level codes emerges, we categorized those codes in hierarchical structure (Affinity diagram; "**3.affinity_digrams_iterations**"). We did three iterations of a negotiated agreement, arranging the codes in different configurations to obtain the final structure "**3.affinity_final_iteration.pdf**," this affinity diagram becomes the outcome of the qualitative data with three categories (Teaching content, Challenges, and Benefits). Thus, we select the most prominent activities based on how much time participants/mentors spent. We validate those activities in the Quantitative study.

B. Quantitative Data

The dataset that we used for our quantitative analysis are based on these metrics"**Table1_Metrics.pdf**." Except for gender diversity, the rest of the data "**4.dataset**" were extracted from OpenStack codebase repository; Git[1], Gerrit[2], Issues tracking system[3], and internal dataset not available for the general public for confidentiality. In all the cases of our quantitative analysis, we compared three categories Cat-1 vs. Cat-2, and Cat-2 vs. Cat-3. If no difference exist between Cat-1 vs. Cat-2, then we assume that any different between Cat-2 vs. Cat-3 will be correlated to onboarding affects.

[1]https://opendev.org/openstack/
[2]https://review.opendev.org/q/status:open+-is:wip
[3]https://storyboard.openstack.org/\#!/page/about

Scripts to extract and analyze Data Our scripts are available in the "**5.scripts**," and use the **0.requirements.txt** python packages requirements along side R-scripts. To install the dependencies to replicate our result, do:

```
#!/bin/bash
$ pip3 install -r requirements.txt
```

The script to extract and analyze contributors' data from OpenStack repositories; such as issues tracker are **15.bug_inducing_commits.py** and **18.getting_the_bugInducingCommits.py**. The main script to analyze data for various metrics during new contributors mentoring activities is ("**11.onboarding.html**") and the statistical analysis is ("'**12.Statistical_testing.html**'). Each of these files is an *.html* copy of the Jupyter notebook.

On the other hand, all our R-script have their dependencies at the header of each file. If the dependencies are not installed in the users' system, the script will run the installation process before running the script.

Since our datasets are available online on public achieved repository[4], users can run our script against any particular dataset to obtain desired results. For example, based on the metrics (Table1_Metrics.pdf) in the quantitative study, users can either run *13.survival_analysis.R* to analyze the time to event (retention) for each new OpenStack contributor in Cat-1, Cat-2, and Cat-3. To measure the impact of onboarding on contributors, we run the python script (**12.Statistical_testing.html**), a Jupyter notebook we converted to a *.html* file.

```
#!/bin/bash
$jupyter nbconvert --to html FORMAT *.ipynb
```

II. Usage and Summary of Artifacts

To use/replicate our research, we have made available our dataset and scripts in sequential order: *1.codebook, 2., ..., 5.scripts.* This data comes both from qualitative and quantitative research methods. In particular, researchers can decide to run only the quantitative or the qualitative script/data. We used python 3.8.5 with the dependencies mentioned in *requirement.txt*. R version 4.0.3. To run a python script, download a particular data file and the data that the file is pointing at, for example, the *20.multiplot.py* script points at the *11.corporate_diversity.csv* dataset.

```
#!/bin/bash
$ python3 multiplot.py
```

This chart in Figure 1 uses multiple axes to plot different parameters in a single chart, these way researchers can customize our chart to fit in multiple metrics with similar characteristic to visualize data more accurately.

Indeed, our "*21.radar.py*" script summarizes the metrics that we used in the quantitative analysis comparing Cat-2 vs. Cat-3 contributors and shows a statistical significant different in all

[4]https://github.com/conferencepapers/ICSE_2021.git

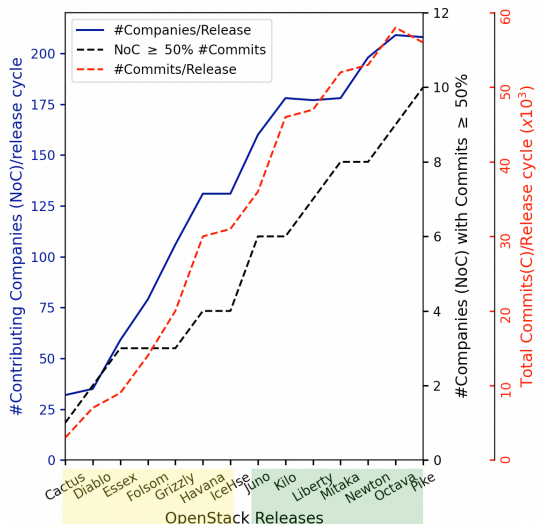

Figure 1. The evolution of the number of companies (NoC, solid blue line) for each of the 7 studied OpenStack releases before (Cat-1, yellow) and after (Cat-2/3, green) the introduction of onboarding. The black dashed line represents the top NoC responsible for 50% of a release's commits and the red dashed lines shows the total commits per release cycle.

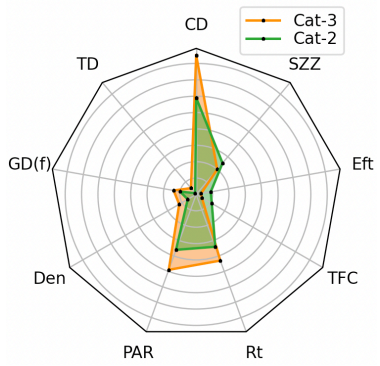

Figure 2. Radar chart of the studied metrics showing that onboarding (Cat-3) has significant differences and improvements over Cat-2. The metrics are those of "**Table1_Metrics.pdf**.": Bug-inducing-commits (SZZ), Effort (Eft), Time to first commit (TFC), Retention (Rt), Patch Acceptance Rate (PAR), Density (Den), Diversity: Gender (GD(f)), Technical (TD), and Corporate Diversity (CD).

the studied metrics. With such a chart as shown in Figure 2, users can customize it to compare two or more groups under a studied phenomena. We use log-log scale to normalize the data for both groups.

Similarly, to run any of the scripts in this package, researchers should note that we have cleaned and pre-processed all datasets using data mining techniques [3].

References

[1] A. Rahman, M. R. Rahman, C. Parnin, and L. Williams, "Security smells in ansible and chef scripts: A replication study," *ACM Trans. Softw. Eng. Methodol.*, vol. 30, no. 1, Jan. 2021. [Online]. Available: https://doi.org/10.1145/3408897

[2] conferencepapers, "conferencepapers/icse_2021: Final release icse21," Jan. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4457683

[3] S. Tang, S. Yuan, and Y. Zhu, "Data preprocessing techniques in convolutional neural network based on fault diagnosis towards rotating machinery," *IEEE Access*, vol. 8, pp. 149 487–149 496, 2020.