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

Armstrong Foundjem†, Ellis E. Eghan‡, Bram Adams†
MCIS Laboratory — (†Queen's University, ‡Polytechnique Montréal, Canada)
{a.foundjem, bram.adams}@queensu.ca, ellis.eghan@polymtl.ca

***Abstract*—Despite the growing success of open-source software ecosystems (SECOs), their sustainability depends on the recruitment and involvement of ever-larger contributors. As such, onboarding, i.e., the socio-technical adaptation of new contributors to a SECO, forms a significant aspect of a SECO's growth that requires substantial resources. Unfortunately, despite theoretical models and initial user studies to examine the potential benefits of onboarding, little is known about the process of SECO onboarding, nor about the socio-technical benefits and drawbacks of contributors' onboarding experience in a SECO. To address these, we first carry out an observational study of 72 new contributors during an OpenStack onboarding event to provide a catalog of teaching content, teaching strategies, onboarding challenges, and expected benefits. Next, we empirically validate the extent to which diversity, productivity, and quality benefits are achieved by mining code changes, reviews, and contributors' issues with(out) OpenStack onboarding experience. Among other findings, our study shows a significant correlation with increasing gender diversity (65% for both females and non-binary contributors) and patch acceptance rates (13.5%). Onboarding also has a significant negative correlation with the time until a contributor's first commit and bug-proneness of contributions.**

***Index Terms*—Onboarding, Mentoring, Collaboration, contributors, knowledge-transfer, Software ecosystems, Open source.**

I. Introduction

Substantial research conducted by both the academic and industrial sectors over the past two decades has attributed most of the success of open-source software (OSS) projects and ecosystems (SECOs) to the strong involvement of contributors, both volunteers and paid employees of involved companies [1]–[4]. Apart from attracting and retaining talented contributors, another major challenge faced by software projects and SECOs is the practical training of new contributors [5], [6], specifically, the onboarding experience of new contributors.

Despite sharing similar goals, SECO-level onboarding programs differ from onboarding programs of individual projects since a SECO is not just the sum of its parts but also "a set of independent and interrelated OSS projects working together for a common objective" [1]. On the one hand, individual projects use different workflows and technologies (requiring different skill-sets) and have independent sets of features and release roadmap. On the other hand, projects have to collaborate with other projects that they depend on. Such cross-project coordination implies the need for onboarding to cover inter-disciplinary processes and tools, compared to the more domain-specific training individual projects provide. SECOs have to ensure that, despite differences in roadmaps, all of their projects can be integrated at set times and can achieve major SECO milestones such as a joint SECO release (e.g., Eclipse, OpenStack, Linux distributions).

Thus, SECO-level onboarding programs should enable new contributors to learn and master both the general SECO processes and concepts and the specific workflows and tools of the individual SECO project(s) in which they want to be active contributors. Several existing works have explored onboarding as an event within proprietary and open-source software communities [6]–[9]. However, these studies focus on individual projects. Only a few studies have investigated the benefits and drawbacks of contributors' (one-time) onboarding event in large organizations [10], [11]. Thus, little is known yet about the benefits and drawbacks of contributors onboarding in the context of SECOs.

Therefore, we aim at reducing the gap in current literature regarding **understanding the process and impact[1] of onboarding in/on open-source SECOs** by conducting an empirical study of the OpenStack SECO. We choose the OpenStack SECO among other contenders such as GNOME, the Apache foundation, Eclipse, CRAN, or the Linux kernel because it is one of the world's fastest-growing open-source software ecosystems [12]. OpenStack has over 100K community members distributed across 182 countries, managed by a consortium of about 693 supporting companies, and organizes two major onboarding events yearly in different geographical locations.

First, we follow a mixed-method research approach by first performing a direct observational study of 72 new contributors to identify the activities performed during a two-day OpenStack onboarding event and identify any perceived challenges and benefits of SECO onboarding. Next, we conduct a quantitative study of the submitted code changes, code reviews, and issues of 1,281 contributors of the OpenStack ecosystem to measure the correlation between onboarding experience and contributor diversity, productivity, and contribution quality.

Our findings show that the OpenStack SECO uses a wide variety of content and strategies to train new contributors during SECO-level onboarding, trying to address 13 challenges involved in SECO onboarding. We also identified eight benefits expected by SECO onboarding stakeholders. Our quantitative validation of three of these expected benefits shows that participating in onboarding correlates with (amongst others):

1) 65% more gender diversity (both female and non-binary);
2) a median of 14% less buggy code contributions;
3) a median increase of 61% in the average code churn;

[1] Any usage of the words "impact" or "influence" refers to the correlation sense of these terms, and does not imply causality.

4) a median 45% (35%) shorter time to get code contributions accepted for female (other) contributors;
5) a 35% (10%/4.5%) longer average retention rate for female (male/non-binary) contributors in the SECO;
6) a median 13.5% higher pull request acceptance rate.

II. Background and Related Work

A. The SECO Onboarding Process

Given that SECOs constitute a complex set of interdependent project/cross-project teams working together for a common goal [1], a SECO's onboarding program is a "continuous" process that usually has two phases [13]: (i) top-level training, and (ii) (more traditional) project-specific training [8], [10], [14].

First, the top-level training introduces new contributors to the SECO's overall complexities, such as its organization, overall workflow, SECO-wide tools, processes, etc. Such activities also provide networking opportunities between newcomers and mentors across the SECO's sub-projects. Then, newcomers move to (sub-)project-specific training, under the guidance of a personal mentor, to learn the ins and outs of a specific sub-project in the SECO. The expected outcome of the overall SECO onboarding process is that new contributors can make their first accepted contribution.

For example, the OpenStack SECO has a dedicated OpenStack Upstream Institute (OUI) [15] responsible for organizing its onboarding process. OUI is necessary since OpenStack ranks among the largest open-source collaborative communities globally with a codebase size of over 20M LOC and produces a new SECO release every six months [16]. Due to its vast diversity in projects (with over 2,000 project/subproject, technical standards, and social norms), new contributors may experience difficulties understanding the roadmap of OpenStack, which can significantly slow down contributions to the codebase.

The OUI organizes the OpenStack onboarding process in two phases — a two-day physical top-level training event, followed by several months of one-to-one online mentoring. The physical event serves to share knowledge on the cross-project processes (planning and dependencies) and tools such as ZUUL (for CI/CD) and Storyboard (for issues tracking) designed to coordinate SECO-level activities. Likewise, the online mentoring phase focuses on processes and tools specific to sub-projects, as well as each project's own work culture. Since OpenStack SECO is distributed across different geolocations, the OUI has to balance the in-person top-level training event's location and time to be equally accessible across new contributors.

B. Related Work

Prior studies mostly focused on the project-specific onboarding phase.

Sharma et al. [8] explored the relationship between successful (short-term) onboarding results and job satisfaction (contributors' intention to either leave or remain active with an organization). Their results suggest that job satisfaction is directly related to both onboarding success and turnover intention. However, they found no relationship in workplace quality. Our study identified eight benefits of onboarding at the SECO level and found that contributors who did onboarding stay longer in the SECO than those who did not.

Fagerholm et al. [10] explored onboarding in a pilot program organized and sponsored by Facebook (under the Education Modernization Program for OSS projects) in collaboration with universities across the globe. A study conducted with 120 students showed that participants who were deliberately mentored during the entire onboarding process were more motivated and committed than their counterparts who did not follow the onboarding process. Our study also shows that contributors who did onboarding were self-motivated and more productive than those who did not do onboarding.

Viviani et al. [14] took a different approach and focused on onboarding in smaller companies that follow a fast software release cycle. They observed a stronger bond among developers, mainly due to close mentoring relationships between core reviewers and younger developers. Contrarily, our study focuses on large and complex SECO. However, we also found new contributors collaborating with mentors (expert-novice collaboration) and expert-expert and novice-novice collaboration.

Britto et al. [17] adopt a model to measure the state of onboarding in software organizations. Steinmacher et al. [18] qualitatively study systematic literature reviews and responses from various practitioners (including an interview study) across several OSS projects to understand the obstacles new developers in an ecosystem from actively contributing to projects. In our research, we found 13 challenges associated with SECO onboarding.

Using the GitHub ecosystem as a case study, Casalnuovo et al. [11] investigate the effects of socialization on a developer joining a new project, a process which the authors refer to as onboarding. They analyze the information of 1,255 developers contributing to a total of 58,092 GitHub projects. Their analysis shows that both the social and technical factors of prior connections and experiences that developers established with experienced team members of a new project have a lasting effect that substantially affects these new members' productivity. Our work found that contributors who participated in the mentoring program were more productive than those who did not participate.

Labuschagne et al. [19] studied the impact of the onboarding program at Mozilla and found that onboarding does not relate to contributor retention. They did not control for prior experience or self-motivation of contributors. At the same time, we show that self-motivation and commitment are challenges SECOs should manage. The onboarding program correlates to a high retention rate, productivity, quality, and diversity.

On the other hand, Zhang et al. [2] studied how companies collaborate within OpenStack by measuring productivity at the release level (while we focus on the release before OpenStack introduces onboarding). Even though their work is not directly related to onboarding at the SECO-level, it, however, explores contributors' paid and volunteered productivity, which, in our case, refers to project-level mentoring for onboarding contributors.

Given that related work has focused mostly on project-specific onboarding, this paper first studies in detail the top-level SECO onboarding phase through an observational study.

Onboarded SECOs' participants can start contributing to the codebase after obtaining both the (i) top-level and (ii) project-specific know-how. Thus, we quantitatively study the correlation between their later contributions (in terms of productivity, code quality, and diversity) and the *overall* onboarding process that they followed.

III. Observational Study of Top-level SECO Onboarding Phase

A. Study Design

To understand how a regular, top-level onboarding training is organized in a SECO, we conducted an observational study of 72 new contributors at an OpenStack onboarding event held in Berlin, Germany, on November 11-12, 2018.

In particular, we aim at investigating the following preliminary research questions:

- **PRQ_1**: What (and how) are the topics taught during a SECO onboarding event?
- **PRQ_2**: What are the challenges involved with organizing and sustaining a SECO onboarding program?
- **PRQ_3**: What are the benefits of a SECO onboarding program?

We describe this observational study's design and present the results of the PRQ_1 below. Meanwhile, we will discuss the results of PRQ_2 and PRQ_3 in section IV.

Participant Selection. Participants for our observational study consist of the pre-registered individuals who completed the two-days onboarding event in Berlin. All participants signed a non-disclosure agreement (consent form) with OpenStack, willfully granting OpenStack the permission to record and document all activities during the entire onboarding event.

These participants command good programming skills in at least Python, have formal college/university education in Computer Science or a related field, and no prior experience with OpenStack or similar SECO. Their average age was 25±5 years, and they exhibited a high demographic diversity in terms of continents and gender (male, female, and non-binary). We obtained this confidential demographic information data either from the participants themselves before the observation study started or from the OpenStack D&I working group's private records of contributors' identities [20], to which we obtained access.

Study Procedure. The observational study involved 72 participants (P1, ..., P72) and 13 mentors (M1, ..., M12), including the observer (OB1; first author).

At any given instance, each of the 12 tables has six participants and a mentor, with at least one mentor leading a task or an exercise. Participants are encouraged to choose their seats and team members freely. Besides the high-quality audio-visual equipment that OpenStack provided, OB1 also used field notes to document mentors' and participants' activities during the entire onboarding event.

To understand the participants' various activities, OB1 used an observational approach with a low degree of interaction with participants but a high Hawthorne effect [21]: all the 72 participants were aware that they were under observation. Moreover, as mentors assign new tasks to participants, OB1 would randomly ask a participant to describe the actions taken during the task using the think-aloud protocol on 24 (2x12) randomly selected participants of the 12 tables.

Qualitative Data analysis The first author initially transcribed audio-visual recordings and field notes of all the 72 participants. The first and second author used a combination of inductive and deductive coding at sentence/paragraph level [22]–[25] to analyze the transcribed text to find patterns and themes relevant to the three PRQs. These themes are further grouped/regrouped to form a hierarchical structure known as an affinity diagram [26], which enables us to visualize how concepts of high-level themes are emerging from basic low-level codes/labels.

Inductive Coding With no pre-conceived themes/patterns, the first and second authors independently apply inductive coding on 15% of the transcriptions in the first iteration to create an initial coding scheme. At the end of this iteration, the coders had 66 and 200 codes, respectively. After several discussions and three more iterations of coding, more informative codes emerged, and we merged low-level codes. Both authors agree on a set of 128 codes and a three-level hierarchical structure of code categories.

Deductive Coding In this step, two coders independently apply the existing codes (from the inductive coding step) on the entire transcribed text to identify code examples. Then, calculate the inter-rater reliability (IRR) score using the Cohen kappa coefficient [27]. We perform three iterations of deductive coding and achieved IIR scores of 51%, 62.6%, and a final score of 100%. These iterations involved merging five existing codes, renaming or moving codes to fit different categories, and splitting up some code categories. The result of our coding is available online [28], and we present the final abstraction of high-level codes in the affinity diagram in Fig. 1.

B. PRQ_1: What (and how) are the topics taught during a SECO onboarding event?

We grouped the teaching contents (TC) under THeoretical material (TH), Hands-on content (HO), and the strategies used to implement both the TH and HO, see (Fig. 1). Our online repository [28] contains a detailed set of activities and tasks that participants performed. Using the observational study's transcripts and notes, we could also determine the relative weight of the three groups of TC based on the allocation of time and resources to their corresponding content.

Mentors dedicate 40% of the training materials to TH, which aims to establish a solid foundation for understanding the OpenStack community and the major concepts involved in making open-source contributions to the SECO. Examples of TH contents are `knowledge on community`

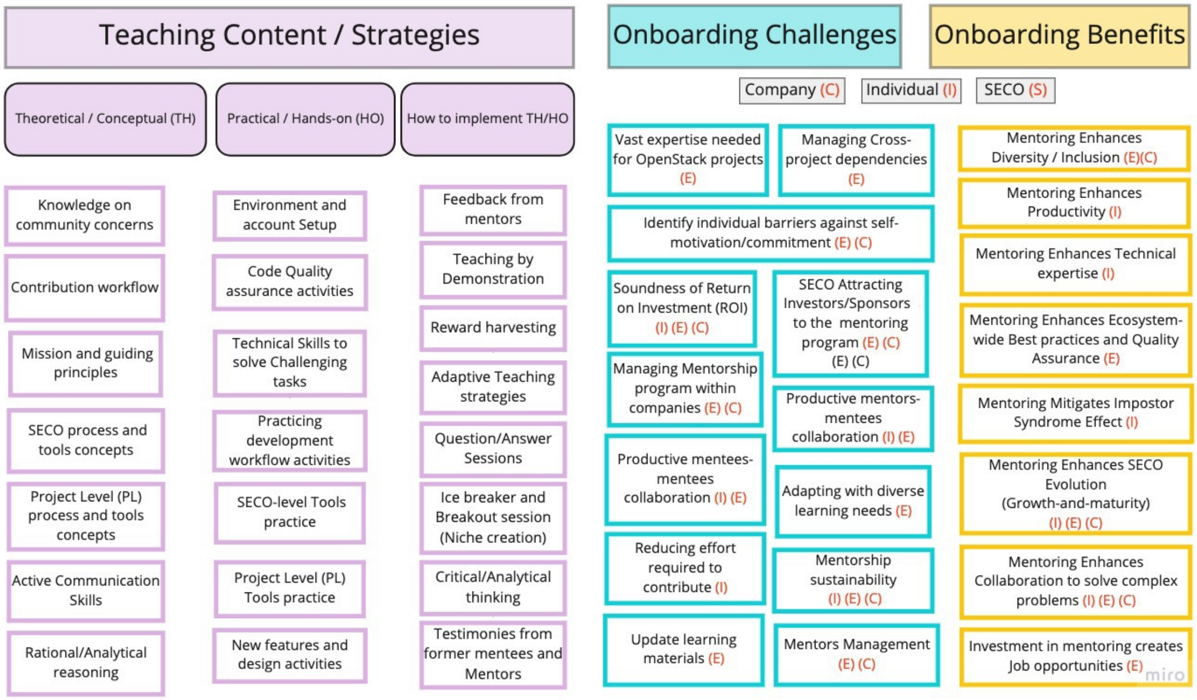


Fig. 1. Materials taught during onboarding and their observed impacts on individual mentees (I), the SECO (E), and companies in the SECO (C).

`concerns`, `mission and guiding principles`, and `contribution workflow`, but also more personal skills like `active communication skills` (why it is crucial to develop this skill, and later on, practice on these skills) and `rational/analytical reasoning`; participants are encouraged through puzzles (training archives) [15] to develop critical thinking abilities [29].

An essential part of this training material focuses on the specifics and differences of SECO-level (`SECO process and tools concepts`) and project-level (`project level (PL) process and tools concepts`) communities and workflows. For example, the need to synchronize each project's release cycle with that of the SECO, stimulate cross-project collaboration, and deal with different workflows and tools (e.g., Storyboard issue tracker at SECO-level vs. Launchpad in several individual projects). Participants reacted to the **TH** differently: "*I am now getting more confident with my understanding of Zuul and rechecks, especially when M6 explained the concepts a few minutes ago; that was a great explanation!*"(**P51**). Yet, another participant appreciates the mentors' efforts: "*I think a load of materials has been too overwhelming, but the mentors make it look too easy for me to follow the concepts.*"(**P29**)

Mentors dedicate 60% of the training materials to HO, which involves hands-on training (50%) and deep-dives into challenging (hackathon) tasks (10%).

The HO component provides participants with a walk-through of typical real-world scenarios and tasks that OpenStack contributors face regularly. The HO component starts with the necessary steps of creating accounts with the OpenStack foundation, Gerrit (code review tool), storyboard (issue tracker), mailing-list, and IRC channels (for communication). Mentors also guide participants to install and configure their (virtual) working environment, which comprises a VirtualBox with possibly a Ubuntu image pre-installed, a copy of the OpenStack development environment (aka DevStack on Sandbox), issue trackers such as Launchpad and Storyboard, the code review environment (Gerrit), and git. Moreover, the OpenStack Sandbox environment (repository) provides virtual servers for testing OpenStack projects/functionalities in an isolated environment. Also, mentors ask participants to perform tasks of varying difficulties covering technical areas. Such as documentation, implementing new features, tracking issues (using storyboard/Launchpad), reviewing source code, best practices on commit messages and code quality, and CI/CD. **OB1** asked a participant to think aloud while performing a HO task: " *I want to run several unit test cases and an integration test. I use the 'tox framework' to run unit testing, so I call the 'tox' command on my terminal [typing ...]* "(**P7**)

Mentors use a variety of teaching strategies that facilitate collaboration and competitiveness among participants throughout the training sessions (Fig. 1). These strategies enhance participants' understanding of the teaching content by making the sessions interactive. The most observed strategies include the following:

`Ice-breaker and breakout session`. Training sessions begin with an introductory activity by both mentors and participants to create an atmosphere of familiarity that facilitates collaboration among participants (novice-novice collaboration) and mentors (novice-expert collaboration). Breakout sessions during the event further strengthen this collaboration. `Expert-novice feedback`. Mentors usually use this strategy to teach practical skills that require a "trial-and-error" approach. Therefore, they allow participants to make several attempts, while the mentors keep providing constructive feedback until the participants arrive at the answer.

`Teaching by demonstration`. Mentors often demonstrate how things work while explaining the underlying concepts; this approach enriches participants with confidence towards the mentors and the ecosystem.

`Reward harvesting`. Mentors use reward strategies to motivate participants to be competitive and work in a group while completing challenging exercises within an allocated time frame. The first participant to figure out the best solution to a task within that time-frame is **rewarded** with a token, a swag, or a sticker. This strategy required participants to apply critical and analytical thinking.

`Novice-novice collaboration`. Mentors encourage participants to work in small groups of two people at each table and discuss their problems/solutions table-wise.

Participants were mostly positive regarding the strategies, which mentors used. **P48** said: "*I like the hands-on section most and, of course, the sticker prizes.*", besides, other participants appreciated different strategies differently: "*The testimony on mentoring was great! I love it.*"(**P15**) Meanwhile, **P31** congratulates the strategy and know-how of the mentors: "*Mentors were great inspirations and knew their stuff well.*" Also, mentors use real-life scenarios to explain difficult concepts: "*I admired the explanations of different projects and how they form an ecosystem.*"(**P1**)

IV. Perceived Challenges and Benefits of SECO Onboarding

Based on the observed onboarding activities shown in Figure 1, 13 challenges and 8 benefits emerged. During our observation, **3/13** challenges and **5/8** benefits encountered substantially more and deeper discussions than others, leading to significantly more words in the transcriptions of the audio-visual recordings. Below, we discuss in detail these three challenges (PRQ_2) and five benefits (PRQ_3).

A. PRQ_2: What are the challenges involved with organizing and sustaining a SECO onboarding program?

Challenge 1: `Vast expertise needed for SECOs`

Onboarding at the SECO-level has several challenges beyond the project-level onboarding. In particular, since a SECO is not just the union of hundreds of smaller projects but involves the collaboration of hundreds of cross-project teams with diverse interacting technologies (see the `cross-project dependencies` challenge). Given that the onboarding participants do not know the different SECO projects, the initial onboarding event cannot make any assumptions. It should target the overall SECO contribution process. To cover a wide variety of topics and tools (see PRQ_1), this also implies that mentors should have polyvalent skill-sets to guide the participants: "*Be prepared with the 'deep dives' exercise. Usually, participants have very different levels of knowledge and skill-set,*"(**M2**) (which in turn impacts `mentorship sustainability`. Furthermore, there should be ongoing communication between the SECO-level onboarding process and the onboarding process within individual projects of the SECO (see `mentorship within companies`), for example, to `update learning materials` to project-level developments.

Challenge 2: `Self-motivation and commitment`

It is challenging for SECO to identify individual barriers against self-motivation/commitment. Therefore, active participation in an onboarding experience is tantamount to a successful outcome, hence every stakeholder should be fully involved and committed. "*Successful mentoring requires active commitment both from the mentor and the mentee.*"(**M9**), also, another mentor advocates "*People learn in different ways at different speeds, which means a commitment to active mentoring requires more than a handful of quick IRC chats.*" (**M7**) This challenge has direct links to the `adapting with diverse learning needs` challenge.

Challenge 3: `Mentorship sustainability`

SECOs and companies face challenges finding available mentors to guide mentees. This is partly because of challenge 1 above, and partly because mentoring requires substantial effort to prepare and keep material up-to-date. Constrained companies may prefer to prioritize their experts' time on tasks that will bring more financial profit to the company, at the detriment of supporting mentees. At the observed onboarding event, participants were briefed that "*If there aren't enough mentors on every table, ... float around the room checking in on people, especially during exercises.*" (**M1**)

B. PRQ_3: What are the benefits of a SECO onboarding program?

Benefit 1: `Mentoring Enhances Diversity`.

Gender diversity (GD): out of the 72 participants at the observed onboarding event, 17 (23.6%) declared themselves as female, 23 (31.94%) as non-binary, and 32 (44.44%) as male. Moreover, for **corporate diversity (CD)**, we found evidence of different companies involved with OpenStack and sponsoring events, and hiring both Cat-2 and Cat-3 contributors. We also observed that mentors and participants had diverse technical skill-sets that cut across different project/cross-project teams. Such, **technical diversity (TD)** brings value to the SECO since it "*drives cross-project teams forward through more mixed reviews, contributions, and viewpoints. By expanding that diversity, we're able to develop a variety of opinions for the open infrastructure project as a whole, ultimately*". (**M9**)

Benefit 2: `Mentoring Enhances productivity`.

During onboarding, mentors assign real-life exercises and tasks to participants, such as creating patch sets, fixing bugs, testing and CI/CD (Zuul), and submitting new features and documentation. All 72 participants actively participated in the coding activities and successfully submitted acceptable commits. This not only trains the participants in the field, but also encourages them to adopt a collaborative workflow (often by themselves), both with other participants (novice-novice) and with mentors (experts-novice). OB1 observed how "*mentors were pairing participants to work on exercises, i.e., P33 and P35 seated on table/group 10, were exchanging ideas constantly throughout this exercise.*" Moreover, M11 asked participants to: "*run different test cases in each project that you cloned. 'If you need help, mentors are seated on your tables, and they will assist you in running the test cases.*'"

Benefit 3: `Enhances SECO QA / best practices`.
Mentors presented various techniques and best practices related to quality assurance (e.g., test-driven development, CI/CD, code reviews) and asked participants to practice those skills. Also, mentors presented a couple of bad and good examples of code that respect OpenStack standards. Some of these best practices include writing good commit messages and proper code documentation. **M9** "*shows a couple of bad examples of commits that reviewers rejected because they violated the best practices, which OpenStack enforces.*"

Benefit 4: `Overcoming imposter syndrome effect`.
New contributors to an ecosystem often feel overwhelmed and inadequate, preventing them from collaborating freely with the other contributors in the ecosystem perceived as having more talent [30]. Thus, it is important for SECOs to take measures to ensure that they help participants to identify and start fighting/eliminating the imposter syndrome [31], [32]. " *As a new developer fresh out of college, coming into any new team can be very intimidating. Everyone around you knows so much more than you, and you feel that you're an imposter with so much to learn* ... "(**P1**). The onboarding program is aware of the effects of the imposter syndrome and sensitizes participants to overcome those, especially by letting mentors and past mentees share their experiences.

Benefit 5: `Evolution of Ecosystem`
As mentors transfer skills to mentees, they produce a larger pool of talent and enable the perpetual growth of the SECO (growth-and-maturity). In turn, previous mentees return to the onboarding program as mentors to help encourage participants to grow within the SECO: "*M7 mentored me during my last year of college, and I have been very fortunate to work with *them* and continue being *their* mentee. ... mentoring helps manage immature skill sets required to grow into a senior engineering role in the future.*"(**M3**).

V. Quantitative Validation of Perceived Benefits

In this section, we empirically evaluate the extent to which onboarding can achieve the three major perceived benefits identified in PRQ3. We could quantify and measure these three benefits by studying 84 months of historical contributions (code changes, issue reports, and code reviews) in the OpenStack SECO. Indeed, we measure Diversity[2], Productivity, and Quality. Specifically, we investigate these three research questions:

– RQ1: Does onboarding correlate with SECO diversity?

[2]To measure Gender diversity at OpenStack, contributors' identity is not publicly available for confidentiality purposes.

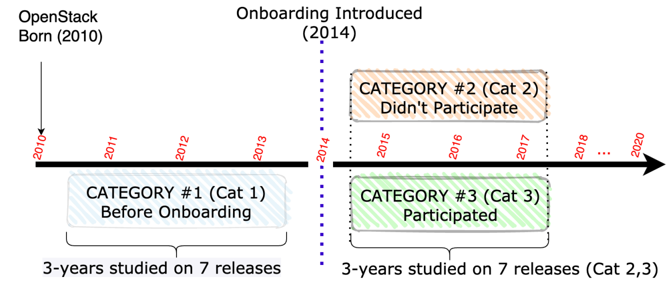


Fig. 2. Timeline of stratified categories used in our study. Cat-1 is our control group, while Cat-2 and Cat-3 are the experimental groups. Each group uses data of seven OpenStack releases (42 months).

– RQ2: Does onboarding correlate with new contributors' productivity?
– RQ3: Does onboarding correlate with new contributors' code quality?

A. Study Design

Categorization of Contributors. OpenStack's onboarding program is publicly advertised, with free training events (travel support is available) taking place in different countries. Hence, anyone is encouraged to do onboarding, not just people who could afford the travel expenses. Therefore, to measure the impact of onboarding on the OpenStack SECO, we considered three categories of contributors in our study (see Fig. 2). The first category (Cat-1) constitutes contributors who joined OpenStack before onboarding events were introduced and could not benefit from any official onboarding. The second category (Cat-2) comprises new contributors who did not participate in any onboarding event, even though the onboarding program did exist when joining OpenStack. Finally, the third category (Cat-3) are contributors who participated in the onboarding program.

Each of the three categories plays an essential role in our study. In particular, for each RQ and metric, we first compare the distribution of the metric values between Cat-2 and Cat-3. If significant differences exist, we perform a second comparison between Cat-1 and Cat-2 to control for any confounding factors such as changes in the development process that were put in place simultaneously when OpenStack introduced the onboarding program. Only if no significant changes exist between Cat-1 and Cat-2 (both of whom consist of contributors who did not do onboarding) can we correlate the differences between Cat-2 and Cat-3 with the introduction of onboarding.

Data Collection. Given the three categories of contributors (Cat-1, Cat-2, and Cat-3), we first use the clustered random sampling technique [33] to randomly select Cat-3 first-time contributors who joined through the OUI onboarding program from different geographic areas, reflecting the distributed nature of SECOs in our sample. This yielded 427 Cat-3 participants across seven OpenStack releases, from Juno to Pike (Fig. 5). Then, we used random sampling to select an equal number of individuals in Cat-1/2. For those two categories, we made sure to exclude any OpenStack contributor who later on (after making contributions) decided to participate in onboarding (720 exclusions).

Finally, we mapped the 1,281 (3x427) selected contributors across all three categories to their activities in the following OpenStack repositories: Gerrit (code review system), git, and Launchpad/Storyboard (issues trackers). Based on this integrated information, we extract contributors' activities related to commits/patch-sets, bugs reported, reviews, blueprints, declared gender, and affiliation for each category's studied period. All experimental data and relevant materials are hosted online [28] for replication or third-party reuse.

Metrics and statistical tests. We adapt existing metrics from the CHAOSS project [34], Meyer et al. [35] and Jansen [36] (see Table I) to measure the extent to which expected benefits of onboarding are achieved at OpenStack.

Our study analyzes these metrics at the individual contributors' level, then aggregates them to the SECO-level, split across the three categories of contributors. Some metrics are general, while others (like Technical and Corporate diversity) are SECO-specific. Note that there is only a weak Pearson correlation of 0.324 between Effort and TFC, i.e., they measure different phenomena.

We then analyze and compare contributor activities among the three categories using several statistical tests. We use survival analysis [37] to measure the amount of time it takes for an event, such as making the first commit, to occur. A (non-parametric) log-rank test is further used to compare the survival curves of multiple groups. If $\rho < \alpha(0.001)$, the tested survival curves are non-overlapping.

For other metrics, we use the Kruskal-Wallis H-test (KW-H) [38] to compare metric distributions of the three contributor categories at once. In case of a statistically significant difference ($\rho < \alpha(0.01)$), a Dunn (posthoc) test [39] is used to identify which of the three categories has a different distribution of metric values. As such, Dunn evaluates Cat-1 vs. Cat-2, Cat-1 vs. Cat-3, and Cat-2 vs. Cat-3. Finally, we measure the effect size (Cliff's delta) [40], which quantifies the effect of significant differences. As explained earlier, we expect that if onboarding correlates with a change in, say, a productivity metric, then Cat-1 (the control group) and Cat-2 (treatment group) should have no statistically significant difference. In contrast, there should be a statistically significant difference between Cat-3 and Cat-2 (and, hence, Cat-1).

B. *RQ1: Does onboarding correlate with SECO diversity?*

This RQ aims to understand the correlation between onboarding and (i) gender representation (gender diversity) within the OpenStack SECO, (ii) the distinct skill sets of contributors (technical diversity), and (iii) the degree to which different corporations/organizations contribute code or sponsor events (corporate diversity).

1) Gender Diversity: **We observed a statistically significant increase of 65%, with large (L) effect sizes, in terms of contributors declared as either female or non-binary within Cat-3 (compared with Cat-2), at the expense of contributors who reported male gender** [20]. Fig. 3a shows how the percentage of contributors who declared themselves as female increased to 33% compared to the 18% (20%) values for Cat-1 (Cat-2). Similarly, for contributors who declared themselves non-binary, the percentage significantly increased from 7% (10%) to 23%.

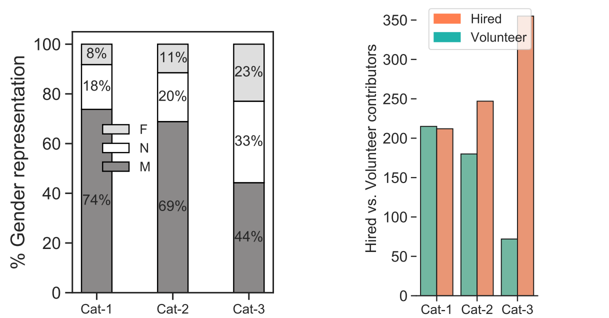
TABLE I
CONTRIBUTOR-LEVEL$*$, SECO-LEVEL†, AND/OR COMPANY-LEVEL♯ METRICS USED IN OUR STUDY.

RQs.	Metrics	Description
RQ1 — Diversity	Gender (GD)†	Proportion of new contributors who self-declare as Male (m), Female (f) or non-binary (n) [20].
	Technical (TD)$*$	The number of different project teams (technology) new contributors are involved in [41].
	Corporate (CD) ♯	The number of sponsoring companies that contribute commits to the SECO [2] [34].
RQ2 — Productivity	Density (Den)$*$	Commit density, i.e., the median proportion of contributed churn over the submitted commits [42].
	Time to first commit (TFC)$*$	Number of days it takes for contributors to have their first commit accepted and merged into the codebase. [34]
	Retention (Rt)$*$	The proportion of contributors, per category, still contributing to the codebase after N days [8] [34].
	Patch Acceptance Rate (PAR)$*$	Probability of a contributor's contribution (‡pull-request; PR) to be accepted (higher values are better) [34]: $$PAR = \frac{\#Accepted\_PRs}{\#Submitted\_PRs} \quad (1)$$
RQ3 — Quality	Effort (Eft)$*$	A measure of the number of ‡pull request versions (attempts) necessary before a contribution is accepted (lower values are better; minimum value of 1) [34]: $$Eft = \frac{Median\_\#Attempts}{\#Actual\_Commits} \quad (2)$$
	Bug-Inducing commits (SZZ)$*$,♯	Percentage of submitted commits that introduce bugs [43].

‡Pull-request (GitHub) or change-request (Gerrit)

The main reason for these increases seems to be the fact that a significantly smaller proportion of contributors explicitly declared themselves as having male gender, which thus far has been the over-represented gender in open source development [44]. There are different interpretations possible. The most likely explanation, supported by the fact that we did not find a significant difference in gender between Cat-1 and Cat-2, is that onboarding helped to attract a larger proportion of contributors of female gender, while providing confidence to others to declare themselves as non-binary instead of sticking to a binary gender. Self-disclosed gender at OpenStack [20] is not open to the general public; it is available in internal profiles for confidentiality purposes. However, there could still be confounding factors. For example, maybe contributors with male gender are less likely to participate in onboarding events. More research is needed to better understand this.

2) Technical Diversity (TD): **People who followed onboarding (irrespective of gender) are more polyvalent than**



(a) Gender Diversity (GD)

(b) Hired vs. Vols

Fig. 3. (Left) Median GD (in %) of each category ((F)emale, (M)ale, and (N)on-binary) ; (Right) Hired vs. volunteer (Vols) contributors in Cat-1, Cat-2, and Cat-3.

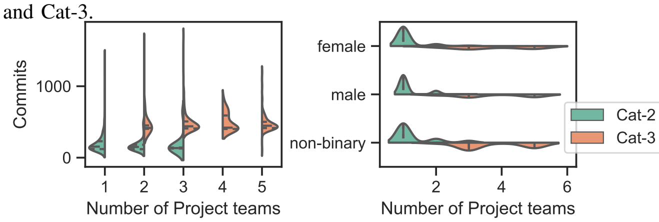


Fig. 4. Overview of technical diversity, showing the number of commits made across different numbers of projects (Left) and the number of projects people contribute to per declared gender (Right).

other contributors. Technical/code diversity measures the number of distinct projects (modules) to which a developer contributes source code. Fig. 4 shows that people who joined OpenStack without onboarding (Cat-1, not shown, & Cat-2) contribute to at most three projects, whereas people who joined through onboarding (Cat-3) often are contributors in more than three projects. For example, in Cat-2, 82.7% of individuals contribute to only one project, 16.6% contribute to two, and only 0.7% contribute to three projects; only contributors with non-binary or male gender contributed to two or more projects in Cat-2. On the other hand, in Cat-3, 52.7% contribute to three core projects, 31.9% contribute to four projects, and 15.5% to five or more projects; contributors who declared male or non-binary gender are mostly contributing to three and four projects, while contributors who declared female gender are even contributing to five or more projects (significant difference between female and other genders). This supports our earlier findings about gender diversity (Section V-B1).

Furthermore, we find a statistically significant difference (large effect size) between Cat-2 and Cat-3 in terms of TD, and the number of commits made by Cat-3 contributors is significantly higher than those by Cat-2 contributors (median of 150 compared to 375).

3) Corporate Diversity (CD): refers to the way in which the Cat-2 and Cat-3 contributors who contribute code to a SECO are distributed across companies. It also measures if a particular company has a monopoly of over 50% or more of these contributions, which could influence the work culture of the SECO or, in the worst case (departure of key contributors), could cripple the SECO [45]–[48].

Studies [2], [9] show that companies contributing to the OpenStack codebase have an uneven distribution of com-

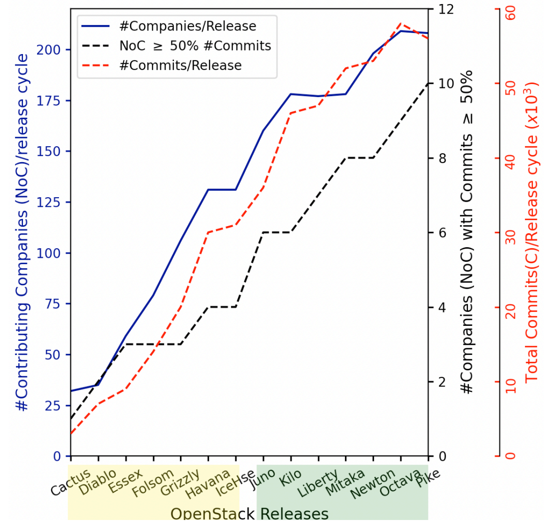


Fig. 5. The evolution of the number of companies (NoC, solid blue line) for each of the 7 studied OpenStack releases before (Cat-1, yellow) and after (Cat-2/3, green) the introduction of onboarding. The black dashed line represents the top NoC responsible for 50% of a release's commits and the red dashed lines shows the total commits per release cycle.

mit across those companies. **Also, we found that none of the sponsoring companies (NoC) had a disproportionate amount of contribution either by Cat-2 or Cat-3 contributors. Furthermore, 83% of Cat-3 contributors were hired by companies compared to 51% of Cat-2 contributors**, and this difference is statistically significant with a ρ-value of $3.006x10^{-40}$ and a large (L) effect size. We also observed that no single company has a dominating share of contributors (and contributions).

Furthermore, Fig. 3b shows how the number of new contributors that remained volunteers instead of being hired dropped substantially from 48% in Cat-2 to 17% in Cat-3. In other words, onboarding seems to be correlated with higher chances of being hired by OpenStack companies.

Only 13% of Cat-3 contributors were hired by the companies that sponsor the onboarding events 70% of the 83% hired Cat-3 contributors were employed by companies within OpenStack that do not sponsor onboarding (median days to hire for Cat-3 is 33.0 vs. 212.0 for Cat-2). While, overall, the high percentage of 83% is positive for the ecosystem as a whole, the sizeable proportion of contributors hired by non-sponsoring companies could be interpreted as a form of "brain drain" and "low return of interest" for the companies organizing the onboarding training.

On a positive note, though, we observe that seven of the top[3] 10 Cat-3 contributors in the SECO were hired by sponsoring companies, which improves their onboarding ROI. On the other hand, Cat-3's hired contributors switch more easily from one company to another. This could indicate that the expertise of Cat-3 contributors is useful and sought-after in different

[3]We used rankdata [49] on *TFC*, *SZZ*, and *Effort* to rank and sort the vectors of contributors in ascending order according to each of these three metrics separately. Since a contributor can be better in one metric but worse in the other, rankdata then aggregates the scores to identify the top 10 contributors.

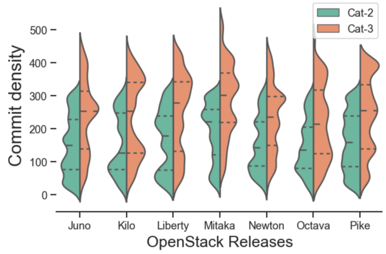


Fig. 6. Comparison of commit density between Cat-2 and Cat-3.

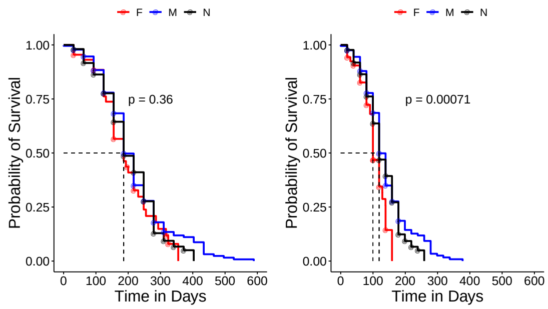


Fig. 7. Survival curves for time until the first accepted contribution per gender in Cat-2 (left) and Cat-3 (right).

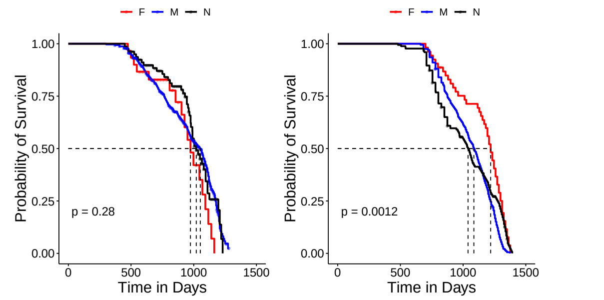


Fig. 8. Survival curves for the time until Cat-2 (left) and Cat-3 (right) contributors leave.

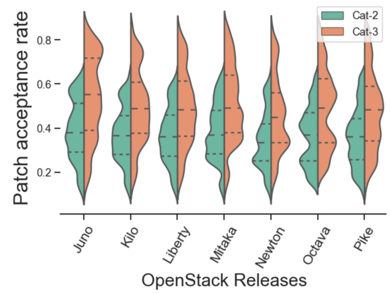


Fig. 9. Comparison of patch acceptance rate between Cat-2 and Cat-3

contexts, or could be due to technology transfer between ecosystem companies. For example, one given contributor started contributing to the Nova project in the Pike release cycle with *IBM*, switched to *Huawei* and later to *Futurewei*, all between February 20$^{\text{th}}$ – August 1$^{\text{st}}$ (2017).

C. RQ2: Does onboarding correlate with new contributors' productivity?

1) Commit Density (Den): **Onboarding correlates with increased contributor productivity**. Fig. 6 shows a 61% increase in the median density of Cat-3 contributions compared to Cat-2 contributions, which is a statistically significant difference with large effect size (while no difference was observed between Cat-1 and Cat-2 contributions). This indicates that people who did onboarding consistently produce a higher average churn across their contributions.

2) Time to first commit (TFC): **Onboarding correlates with a median 45% or 35% lower time to first commit for female (male/non-binary) contributors.** Fig. 7 shows the survival curves [50] (with ρ-values obtained using the log-rank test) for the time until first commit (in number of days) for the three categories of contributors, split across the three genders. Only for Cat-3, we obtained statistically significant results among the genders. Furthermore, we obtained a significant difference with large effect size between Cat-2 against Cat-3, across all three genders. It takes 100 (120) days for half of the female (male/non-binary) contributors in Cat-3 to make their first commits, while in Cat-2, it takes at least 185 days for any contributor (either gender) to get their first commit accepted.

3) Retention rate: **Onboarding correlates with a 16% longer average retention rate across the three genders in the SECO**, i.e., Cat-3 contributors are active much longer than Cat-2 (and Cat-1) contributors, which is beneficial for the sustainability and cohesion within a community. We observed from the survival analysis chart (Fig. 8) that 95% of contributors were active for 450 days in Cat-2 and 750 days (four SECO release cycles) in Cat-3. While there is a 50% probability of Cat-1 contributors (either gender) abandoning the SECO/project within 1,000 days (not shown), this retention period is 1,100 (1,000) days for Cat-3 (Cat-2) non-binary contributors, 1,150 (1,100) days for males, and 1,290 (950) days for females. Therefore, contributors, on average, were productive for a significantly longer time in Cat-3 than in both Cat-1 and Cat-2 (large effect sizes), with self-declared female contributors with onboarding experience persisting longer than any other declared gender.

4) Patch Acceptance Rate (PAR): **Onboarding correlates with a significant increase in the percentage of accepted pull requests (i.e., Gerrit "change requests")**, i.e., contributors are more successful in getting their patches accepted. Fig. 9 (top) shows that the median PAR for Cat-3 contributors is 35.7% to 49.2% times higher compared to Cat-2 contributors. Our evidence suggests that contributors self-declared as female outperformed the other genders (not shown), in both Cat-2 and Cat-3, in terms of PAR (large effect size).

D. RQ3: Does onboarding correlate with new contributors' code quality?

1) Effort: **Cat-3 contributors require less effort to have their commit accepted.** Based on our observation and results in Fig. 10 , contributors who joined the ecosystem without an onboarding training (Cat-1 & 2) on average require significantly more attempts to get their contributions accepted than those who were onboarded (Cat-3), with ρ-value of $6.621x10^{-77}$ and large effect size.

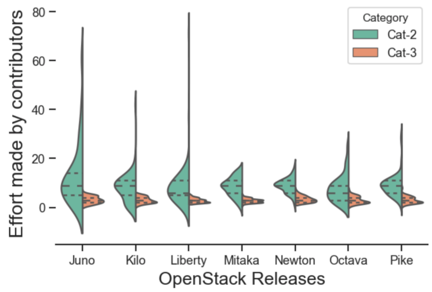


Fig. 10. An Overview of effort needed by Cat-2/3 contributors.

Fig. 11. Likelihood of bug-inducing commits across Cat-2/3.

Since this observation only holds for Cat-3, this provides initial evidence for the hypothesis that onboarding enables contributors to better master the codebase, workflow and guidelines of an ecosystem. More research is needed to further validate this claim.

2) Bug-inducing Changes: **Contributors who did onboarding produce code that is 14% less likely to introduce bugs**. Using the PyDriller [51] implementation of the SZZ algorithm [52]–[56], our results show that the median probability of a commit introducing a bug is 25% for Cat-2 compared to 14% for Cat-3 (Fig. 11). In other words, accepted patches are less buggy for Cat-3, even though Cat-3 contributors submit a higher quantity (with more complexity) of code changes than contributors from the other categories (as previously discussed in RQ1 for TD). These differences are significant with a ρ-value of $4.290x10^{-57}$ and a large effect size. Not only are patches of Cat-3 contributors less buggy, they also required less attempts to be accepted (see previous metric).

VI. Discussion

Based on the observational study findings (Fig. 1), we notice how the themes in the affinity diagram form a holistic set of socio-technical activities relevant to onboarding in a complex SECO. Such onboarding is more than giving a tutorial on creating a feature branch or running a test suite. Mentors spent substantial effort explaining the interactions and differences between the OpenStack SECO and the individual projects inside the SECO. `Knowledge on community concerns` is another essential pillar of the teaching content, as well as activities to train participants' `active communication skills` and `rational/analytical reasoning`. Combining such topics with the more technical hand-on activities requires (i) the use of a host of engaging teaching strategies, as well as

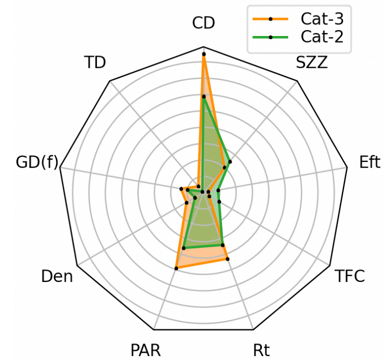


Fig. 12. Radar chart of the studied metrics showing that onboarding (Cat-3) has significant differences and improvements over Cat-2. The metrics are those of Table I: Bug-inducing-commits (SZZ), Effort (Eft), Time to first commit (TFC), Retention (Rt), Patch Acceptance Rate (PAR), Density (Den), Diversity: Gender (GD(f)), Technical (TD), and Corporate Diversity (CD).

(ii) a continuous (online) onboarding process that goes well beyond the initial onboarding event.

While such an onboarding process requires an investment, both financially and in terms of in-kind, SECOs expect that the process can boost new recruits' productivity and the quality of their contributions and foster an inclusive and diverse community, able to sustain the SECO.

In particular, we observed that as the community grooms new contributors, they later become resourceful to the community by impacting other new contributors' growth by becoming mentors themselves. The idea is that the community evolves; mentees become mentors, and contributors stay longer within the community.

Our quantitative evaluation found evidence that some of these major expectations indeed seem to hold. The radar chart in Fig. 12 shows the extent to which the diversity, productivity, and quality of onboarded contributors (Cat-3) differ from contributors without onboarding (Cat-2). For each metric, the chart plots the median values for Cat-2/3 at the contributor-, SECO- (GD) or company-level (CD, SZZ), using log-scale.

In particular, onboarding correlates with improved diversity (GD(f), TD, and CD) and productivity (TFC, Rt, PAR, and Den) metrics, since contributors in Cat-3 recorded significantly higher values in these metrics against Cat-2 contributors. However, onboarding correlates with reduced bug-inducing commits (SZZ) and efforts (Eft). Given that Cat-3 contributors seem to spend less effort in making quality code changes (commits). On the other hand, Cat-2 spent more time making their first accepted contributions (TFC) in terms of productivity; they also expend more effort, which are more likely to be bug inducing. Onboarded contributors stay longer in the SECO and make diversity more visible, but not necessarily within one SECO project or company or with a company sponsoring the onboarding process. Other potential benefits still need to be empirically evaluated.

Finally, several challenges could potentially complicate or even inhibit the onboarding process. A substantial amount of these challenges relate to people management—notably, the steady supply of motivated participants and capable mentors.

While successful onboarding could yield new future mentors, both the SECO and academia should monitor this continuity carefully not to overload the same group of experts. At the same time, the latter have to keep on reinventing their teaching strategies to effectively teach the minimum material covering as much as possible the workflow and requirements of both the overarching SECO and the individual projects to be productive as fast as possible. Future research should explore and address these challenges.

VII. Threats to validity

Construct validity. This study uses existing diversity, productivity, and quality metrics from the literature [2], [35], [57], [58] and open source communities such as CHAOSS [34]. However, concerning gender, we relied on the self-declared gender available in OpenStack's internal profiles [20]. Furthermore, we observed an onboarding event and mined readily available data from version control, issue reports, and code review repositories but did not have access to the private online communication between mentors and mentees after the onboarding event.

Another threat relates to the impact of the participants' awareness of our observation study on their behavior. To mitigate this, we observed selected people on a given task. We watched the onboarding event's video recording to validate how other participants performed the same activity when not directly observing them.

Internal validity. Confounding factors may have been responsible for some of the observed differences between Cat-2 and Cat-3 contributors, i.e., factors other than the introduction of onboarding could explain some of our findings. Our study design included the Cat-1 control group, which, similar to Cat-2, consists of participants that did not do onboarding to mitigate this threat. Hence, if, for a given metric, no changes are observed between Cat-1 and Cat-2, the likelihood of confounding factors reduces (but not to zero). None of our quantitative analyses observed statistical differences between Cat-1 and Cat-2.

Another threat concerns the effect of unreported bugs on the result of the SZZ bug-inducing commit analysis, which uses an implementation of the original SZZ algorithm [52]. To mitigate this, we run SZZ on the entire history of OpenStack's official issue tracking systems (Launchpad/Storyboard). Also, our study window spans 14 releases (7 for Cat-1 and 7 for Cat-2/3), which gives ample time for contributors to make active contributions. We base our study on the assumption that participants/contributors had no prior experience with any SECO. However, since some educational institutions introduce their students to open-source development concepts and practices as part of their learning path, this could be a confounding factor that could affect our results. Since "generic" development concepts form only a minor part of the onboarding process, we believe this threat is minimal.

External validity. While OpenStack is a representative modern SECO, our results may not generalize to other ecosystems. That said, the methods that we use in our observational and quantitative studies are ecosystem-agnostic. Hence, practitioners and researchers could use our methods to identify and evaluate the impact of any ecosystem's onboarding program. As a side note, the post-Covid-19 era fosters a culture of online collaboration that could disrupt the dynamics of in-person [59] events. Even though Rodeghero et al. [60] studied onboarding during the Covid-19 pandemic at the project-level, it is still too early to understand the impact of this on the top-level SECO training events or the SECO onboarding process as a whole. For example, the recent OUI training event on October 22-23, 2020, was virtual, yet the turnout was much lower (8 mentors and 11 participants) than previous events. Future research is necessary for the new reality of in-person vs. virtual training events in OSS communities.

Reliability validity. Except for confidential participant information, we provide the necessary description and resources (OSS tools and dataset) [28] needed to replicate our research.

VIII. Conclusion

This paper provides the first large-scale, mixed-methods empirical study on onboarding in SECOs and is amongst the first empirical studies in the domain of software engineering onboarding in general. Though previous research has been conducted on onboarding within software projects, these works did not provide a deeper understanding of the overall SECO onboarding process, which involves an initial, top-level onboarding phase followed by one-to-one project-specific mentoring. Hence, this paper aimed to (1) understand the onboarding process at SECO level, as well as to (2) quantitatively validate the impact of SECO-level onboarding in terms of expected benefits regarding diversity, productivity and quality of contributions.

Our observation study of a top-level OpenStack onboarding event yields a catalogue of six conceptual and eight hands-on categories of socio-technical onboarding content, eight teaching strategies used, eight expected onboarding benefits, and 13 onboarding challenges. Furthermore, our quantitative analysis of OpenStack contributors and contributions shows that contributors who followed the onboarding process spend less time and effort to get their first commit accepted and produce larger, less bug-inducing commits. Moreover, we observe a strong correlation between onboarding and an increase in the gender and technical diversity of the OpenStack SECO. We provide our data set online [28].

The implications of this study are manifold and impact different stakeholders differently: (1) developers have empirical evidence that onboarding could be beneficial for them, since it correlates with increased productivity and chances of being hired by a company of the SECO; (2) (prospective) mentors have an overview of the relevant topics and strategies they should prepare for; and (3) organizations and SECOs as a whole have empirical evidence that investments in onboarding correlate with increased productivity, diversity and quality, while they also have a list of challenges they should be aware of while mounting or operating an onboarding program.

References

[1] G. Poo-Caamaño, E. Knauss, L. Singer, and D. German, "Herding cats in a foss ecosystem: a tale of communication and coordination for release management," *Journal of Internet Services and Applications*, vol. 8, 12 2017.

[2] Y. Zhang, M. Zhou, K.-J. Stol, J. Wu, and Z. Jin, "How do companies collaborate in open source ecosystems? an empirical study of openstack," *2020 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, 2020.

[3] Z. Wang, Y. Feng, Y. Wang, J. A. Jones, and D. Redmiles, "Unveiling elite developers' activities in open source projects," *ACM Trans. Softw. Eng. Methodol.*, vol. 29, no. 3, Jun. 2020. [Online]. Available: https://doi.org/10.1145/3387111

[4] S. Bayati, "Understanding newcomers success in open source community," in *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings*, ser. ICSE '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 224–225. [Online]. Available: https://doi.org/10.1145/3183440.3195073

[5] C. Liu, D. Yang, X. Zhang, H. Hu, J. Barson, and B. Ray, "Poster: A recommender system for developer onboarding," in *2018 IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion)*, May 2018, pp. 319–320.

[6] I. Steinmacher, C. Treude, and M. A. Gerosa, "Let me in: Guidelines for the successful onboarding of newcomers to open source projects," *IEEE Software*, vol. 36, no. 4, pp. 41–49, July 2019.

[7] I. Rehman, D. Wang, R. G. Kula, T. Ishio, and K. Matsumoto, "Newcomer candidate: Characterizing contributions of a novice developer to github," *arXiv preprint arXiv:2008.02597*, 2020.

[8] G. G. Sharma and K.-J. Stol, "Exploring onboarding success, organizational fit, and turnover intention of software professionals," *Journal of Systems and Software*, vol. 159, p. 110442, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016412121930216X

[9] Y. Zhang, M. Zhou, A. Mockus, and Z. Jin, "Companies' participation in oss development - an empirical study of openstack," *IEEE Transactions on Software Engineering*, pp. 1–1, 2019.

[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 8th ACM/IEEE international symposium on empirical software engineering and measurement*. ACM, 2014, p. 55.

[11] C. Casalnuovo, B. Vasilescu, P. Devanbu, and V. Filkov, "Developer onboarding in github: the role of prior social links and language experience," in *Proceedings of the 2015 10th joint meeting on foundations of software engineering*. ACM, 2015, pp. 817–828.

[12] S. Bhartiya, "What open means to openstack," https://www.linuxfoundation.org/blog/2017/12/what-open-means-openstack/, Linux Foundation, December 2017.

[13] Y. Dittrich, "Software engineering beyond the project – sustaining software ecosystems," *Information and Software Technology*, vol. 56, no. 11, pp. 1436 – 1456, 2014, special issue on Software Ecosystems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950584914000652

[14] G. Viviani and G. C. Murphy, "Reflections on onboarding practices in mid-sized companies," in *Proceedings of the 12th International Workshop on Cooperative and Human Aspects of Software Engineering*, ser. CHASE '19. Piscataway, NJ, USA: IEEE Press, 2019, pp. 83–84. [Online]. Available: https://doi.org/10.1109/CHASE.2019.00027

[15] OpenStack-Foundation. (2020, August) Links to oui archived materials. [Online]. Available: https://www.openstack.org,[OUI]https://docs.openstack.org/upstream-training/upstream-archives.html,

[16] J. A. Teixeira and H. Karsten, "Managing to release early, often and on time in the openstack software ecosystem," *Journal of Internet Services and Applications*, vol. 10, no. 1, p. 7, Apr 2019. [Online]. Available: https://doi.org/10.1186/s13174-019-0105-z

[17] R. Britto, D. S. Cruzes, D. Smite, and A. Sablis, "Onboarding software developers and teams in three globally distributed legacy projects: A multi-case study," *Journal of Software: Evolution and Process*, vol. 30, no. 4, p. e1921, 2018.

[18] I. Steinmacher, A. P. Chaves, T. U. Conte, and M. A. Gerosa, "Preliminary empirical identification of barriers faced by newcomers to open source software projects," in *2014 Brazilian Symposium on Software Engineering*. IEEE, 2014, pp. 51–60.

[19] A. Labuschagne and R. Holmes, "Do onboarding programs work?" in *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, May 2015, pp. 381–385.

[20] D. Izquierdo, N. Huesman, A. Serebrenik, and G. Robles, "Openstack gender diversity report," *IEEE Software*, vol. 36, no. 1, pp. 28–33, 2018.

[21] A. N. Meyer, G. C. Murphy, T. Zimmermann, and T. Fritz, "Enabling good work habits in software developers through reflective goal-setting," *IEEE Transactions on Software Engineering*, pp. 1–1, 2019.

[22] J. Meinicke, C.-P. Wong, B. Vasilescu, and C. Kästner, "Exploring differences and commonalities between feature flags and configuration options," in *Proc. Int'l Conf. Software Engineering–Software Engineering in Practice (ICSE-SEIP). ACM*, 2020.

[23] N. Shrestha, C. Botta, T. Barik, and C. Parnin, "Here we go again: Why is it difficult for developers to learn another programming language?" in *Proceedings of the 42nd International Conference on Software Engineering, ICSE*, 2020.

[24] A. Rahman, C. Parnin, and L. Williams, "The seven sins: Security smells in infrastructure as code scripts," in *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, 2019, pp. 164–175.

[25] C. Sadowski, E. Söderberg, L. Church, M. Sipko, and A. Bacchelli, "Modern code review: A case study at google," in *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice*, ser. ICSE-SEIP '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 181–190. [Online]. Available: https://doi.org/10.1145/3183519.3183525

[26] S. Mizuno, *Management for quality improvement: the 7 new QC tools*. CRC Press, 2020.

[27] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, 1977.

[28] A. Foundjem. (2021, January) Replication packages for our study on seco onboarding –scripts/dataset. [Online]. Available: http://doi.org/10.5281/zenodo.4457683

[29] S. Yeasmin and L. A. Albabtain, "Escape the countries: A vr escape room game," in *2020 3rd International Conference on Computer Applications Information Security (ICCAIS)*, 2020, pp. 1–6.

[30] C. Mendez, A. Sarma, and M. Burnett, "Gender in open source software: What the tools tell," in *Proceedings of the 1st International Workshop on Gender Equality in Software Engineering*, ser. GE '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 21–24. [Online]. Available: https://doi.org/10.1145/3195570.3195572

[31] K. Kohl Silveira, S. Musse, I. H. Manssour, R. Vieira, and R. Prikladnicki, "Confidence in programming skills: Gender insights from stackoverflow developers survey," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 2019, pp. 234–235.

[32] A. Filippova, E. Trainer, and J. D. Herbsleb, "From diversity by numbers to diversity as process: Supporting inclusiveness in software development teams with brainstorming," in *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, 2017, pp. 152–163.

[33] M. Alvi. (2016) A manual for selecting sampling techniques in research. [Online]. Available: https://mpra.ub.uni-muenchen.de/70218/

[34] C. P. L. Foundation. (2020, August) Community health analytics open source software. [Online]. Available: https://chaoss.community/metrics/

[35] A. N. Meyer, G. C. Murphy, T. Fritz, and T. Zimmermann, *Developers' Diverging Perceptions of Productivity*. Berkeley, CA: Apress, 2019, pp. 137–146. [Online]. Available: https://doi.org/10.1007/978-1-4842-4221-6_12

[36] S. Jansen, "Measuring the health of open source software ecosystems: Beyond the scope of project health," *Information and Software Technology*, vol. 56, no. 11, pp. 1508–1519, 2014.

[37] P. Chapfuwa, C. Li, N. Mehta, L. Carin, and R. Henao, "Survival cluster analysis," in *Proceedings of the ACM Conference on Health, Inference, and Learning*, ser. CHIL '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 60–68. [Online]. Available: https://doi.org/10.1145/3368555.3384465

[38] S. D. Pawar and D. T. Shirke, "Nonparametric tests for multivariate multi-sample locations based on data depth," *Journal of Statistical Computation and Simulation*, vol. 89, no. 9, pp. 1574–1591, 2019.

[39] M. Allen, "The SAGE Encyclopedia of Communication Research Methods," Dec. 2019. [Online]. Available: https://methods.sagepub.com/reference/the-sage-encyclopedia-of-communication-research-methods

[40] F. Armstrong, F. Khomh, and B. Adams, "Broadcast vs. unicast review technology: Does it matter?" in *2017 IEEE International Conference on*

Software Testing, Verification and Validation (ICST), March 2017, pp. 219–229.

[41] C. Jergensen, A. Sarma, and P. Wagstrom, “The onion patch: Migration in open source ecosystems,” in *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering*, ser. ESEC/FSE ’11. New York, NY, USA: Association for Computing Machinery, 2011, p. 70–80. [Online]. Available: https://doi.org/10.1145/2025113.2025127

[42] S. Hönel, M. Ericsson, W. Löwe, and A. Wingkvist, “Importance and aptitude of source code density for commit classification into maintenance activities,” in *2019 IEEE 19th International Conference on Software Quality, Reliability and Security (QRS)*. IEEE, 2019, pp. 109–120.

[43] E. C. Neto, D. A. da Costa, and U. Kulesza, “Revisiting and improving szz implementations,” in *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, 2019, pp. 1–12.

[44] C. Mendez, H. S. Pedala, Z. Steine-Hanson, C. Hilderbrand, A. Horvath, C. Hill, L. Simpson, N. Patil, A. Sarma, and M. Burnett, “Open source barriers to entry, revisited: A tools perspective,” 2017.

[45] G. Avelino, E. Constantinou, M. T. Valente, and A. Serebrenik, “On the abandonment and survival of open source projects: An empirical investigation,” in *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, Sep. 2019, pp. 1–12.

[46] M. Valiev, B. Vasilescu, and J. Herbsleb, “Ecosystem-level determinants of sustained activity in open-source projects: A case study of the pypi ecosystem,” in *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 644–655. [Online]. Available: https://doi.org/10.1145/3236024.3236062

[47] V. Cosentino, J. L. C. Izquierdo, and J. Cabot, “Assessing the bus factor of git repositories,” in *2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*. IEEE, 2015, pp. 499–503.

[48] I. Steinmacher, I. Wiese, A. P. Chaves, and M. A. Gerosa, “Why do newcomers abandon open source software projects?” in *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, May 2013, pp. 25–32.

[49] J. Li, G. Sun, G. Zhao, and H. L. Li-wei, “Robust low-rank discovery of data-driven partial differential equations,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 767–774.

[50] H. S. Qiu, A. Nolte, A. Brown, A. Serebrenik, and B. Vasilescu, “Going farther together: The impact of social capital on sustained participation in open source,” in *Proceedings of the 41st International Conference on Software Engineering (ICSE) 2019, Montreal, Canada*. IEEE, 2019, to appear.

[51] D. Spadini, M. Aniche, and A. Bacchelli, *PyDriller: Python Framework for Mining Software Repositories*, 2018.

[52] J. Śliwerski, T. Zimmermann, and A. Zeller, “When do changes induce fixes?” *SIGSOFT Softw. Eng. Notes*, vol. 30, no. 4, pp. 1–5, May 2005. [Online]. Available: http://doi.acm.org/10.1145/1082983.1083147

[53] S. Kim, T. Zimmermann, K. Pan, E. James Jr *et al.*, “Automatic identification of bug-introducing changes,” in *21st IEEE/ACM International Conference on Automated Software Engineering (ASE’06)*. IEEE, 2006, pp. 81–90.

[54] C. Sadowski, C. Lewis, Z. Lin, X. Zhu, and E. J. Whitehead Jr, “An empirical analysis of the fixcache algorithm,” in *Proceedings of the 8th Working Conference on Mining Software Repositories*. ACM, 2011, pp. 219–222.

[55] E. C. Neto, D. A. da Costa, and U. Kulesza, “The impact of refactoring changes on the szz algorithm: An empirical study,” in *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2018, pp. 380–390.

[56] G. Rodríguez-Pérez, G. Robles, and J. M. González-Barahona, “Reproducibility and credibility in empirical software engineering: A case study based on a systematic literature review of the use of the szz algorithm,” *Information and Software Technology*, vol. 99, pp. 164–176, 2018.

[57] M. Hilton and A. Begel, “A study of the organizational dynamics of software teams,” in *2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*, May 2017, pp. 191–200.

[58] M. Borg, O. Svensson, K. Berg, and D. Hansson. (2019, 03) Szz unleashed: An open implementation of the szz algorithm - featuring example usage in a study of just-in-time bug prediction for the jenkins project.

[59] S.-F. Wen, “Learning secure programming in open source software communities: A socio-technical view,” in *Proceedings of the 6th International Conference on Information and Education Technology*, ser. ICIET ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 25–32. [Online]. Available: https://doi.org/10.1145/3178158.3178202

[60] P. Rodeghero, T. Zimmermann, B. Houck, and D. Ford, “Please turn your cameras on: Remote onboarding of software developers during a pandemic,” *arXiv preprint arXiv:2011.08130*, 2020.