

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

Armstrong Foundjem<sup>†</sup>, Ellis E. Eghan<sup>‡</sup>, Bram Adams<sup>†</sup>  
MCIS Laboratory — (<sup>†</sup>Queen’s University, <sup>‡</sup>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<sup>1</sup> 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;

<sup>1</sup>Any usage of the words “impact” or “influence” refers to the correlation sense of these terms, and does not imply causality.

- 95 4) a median 45% (35%) shorter time to get code contribu- 149  
96 tions accepted for female (other) contributors; 150
- 97 5) a 35% (10%/4.5%) longer average retention rate for 151  
98 female (male/non-binary) contributors in the SECO; 152
- 99 6) a median 13.5% higher pull request acceptance rate. 153

## 100 II. BACKGROUND AND RELATED WORK

### 101 A. The SECO Onboarding Process

102 Given that SECOs constitute a complex set of inter- 154  
103 dependent project/cross-project teams working together for 155  
104 a common goal [1], a SECO’s onboarding program is a 156  
105 “continuous” process that usually has two phases [13]: (i) 157  
106 top-level training, and (ii) (more traditional) project-specific 158  
107 training [8], [10], [14]. 159

108 First, the top-level training introduces new contributors to 160  
109 the SECO’s overall complexities, such as its organization, 161  
110 overall workflow, SECO-wide tools, processes, etc. Such ac- 162  
111 tivities also provide networking opportunities between new- 163  
112 comers and mentors across the SECO’s sub-projects. Then, 164  
113 newcomers move to (sub-)project-specific training, under the 165  
114 guidance of a personal mentor, to learn the ins and outs of 166  
115 a specific sub-project in the SECO. The expected outcome of 167  
116 the overall SECO onboarding process is that new contributors 168  
117 can make their first accepted contribution. 169

118 For example, the OpenStack SECO has a dedicated Open- 170  
119 Stack Upstream Institute (OUI) [15] responsible for organizing 171  
120 its onboarding process. OUI is necessary since OpenStack 172  
121 ranks among the largest open-source collaborative commu- 173  
122 nities globally with a codebase size of over 20M LOC and 174  
123 produces a new SECO release every six months [16]. Due 175  
124 to its vast diversity in projects (with over 2,000 project/sub- 176  
125 project, technical standards, and social norms), new contribu- 177  
126 tors may experience difficulties understanding the roadmap of 178  
127 OpenStack, which can significantly slow down contributions 179  
128 to the codebase. 180

129 The OUI organizes the OpenStack onboarding process in 181  
130 two phases — a two-day physical top-level training event, 182  
131 followed by several months of one-to-one online mentoring. 183  
132 The physical event serves to share knowledge on the cross- 184  
133 project processes (planning and dependencies) and tools such 185  
134 as ZUUL (for CI/CD) and Storyboard (for issues tracking) 186  
135 designed to coordinate SECO-level activities. Likewise, the 187  
136 online mentoring phase focuses on processes and tools specific 188  
137 to sub-projects, as well as each project’s own work culture. 189  
138 Since OpenStack SECO is distributed across different geo- 190  
139 locations, the OUI has to balance the in-person top-level 191  
140 training event’s location and time to be equally accessible 192  
141 across new contributors. 193

### 142 B. Related Work

143 Prior studies mostly focused on the project-specific on- 194  
144 boarding phase. 195

145 Sharma et al. [8] explored the relationship between suc- 196  
146 cessful (short-term) onboarding results and job satisfaction 197  
147 (contributors’ intention to either leave or remain active with 198  
148 an organization). Their results suggest that job satisfaction 199

is directly related to both onboarding success and turnover 200  
intention. However, they found no relationship in workplace 201  
quality. Our study identified eight benefits of onboarding at the 202  
SECO level and found that contributors who did onboarding 203  
stay longer in the SECO than those who did not. 204

Fagerholm et al. [10] explored onboarding in a pilot pro- 205  
gram organized and sponsored by Facebook (under the Educa- 206  
tion Modernization Program for OSS projects) in collaboration 207  
with universities across the globe. A study conducted with 208  
120 students showed that participants who were deliberately 209  
mentored during the entire onboarding process were more 210  
motivated and committed than their counterparts who did not 211  
follow the onboarding process. Our study also shows that 212  
contributors who did onboarding were self-motivated and more 213  
productive than those who did not do onboarding. 214

Viviani et al. [14] took a different approach and focused on 215  
onboarding in smaller companies that follow a fast software 216  
release cycle. They observed a stronger bond among devel- 217  
opers, mainly due to close mentoring relationships between 218  
core reviewers and younger developers. Contrarily, our study 219  
focuses on large and complex SECO. However, we also found 220  
new contributors collaborating with mentors (expert-novice 221  
collaboration) and expert-expert and novice-novice collabora- 222  
tion. 223

Britto et al. [17] adopt a model to measure the state of 224  
onboarding in software organizations. Steinmacher et al. [18] 225  
qualitatively study systematic literature reviews and responses 226  
from various practitioners (including an interview study) 227  
across several OSS projects to understand the obstacles new 228  
developers in an ecosystem from actively contributing to 229  
projects. In our research, we found 13 challenges associated 230  
with SECO onboarding. 231

Using the GitHub ecosystem as a case study, Casalnuovo et 232  
al. [11] investigate the effects of socialization on a developer 233  
joining a new project, a process which the authors refer to as 234  
onboarding. They analyze the information of 1,255 developers 235  
contributing to a total of 58,092 GitHub projects. Their anal- 236  
ysis shows that both the social and technical factors of prior 237  
connections and experiences that developers established with 238  
experienced team members of a new project have a lasting 239  
effect that substantially affects these new members’ produc- 240  
tivity. Our work found that contributors who participated in 241  
the mentoring program were more productive than those who 242  
did not participate. 243

Labuschagne et al. [19] studied the impact of the onboarding 244  
program at Mozilla and found that onboarding does not relate 245  
to contributor retention. They did not control for prior expe- 246  
rience or self-motivation of contributors. At the same time, 247  
we show that self-motivation and commitment are challenges 248  
SECOs should manage. The onboarding program correlates to 249  
a high retention rate, productivity, quality, and diversity. 250

On the other hand, Zhang et al. [2] studied how companies 251  
collaborate within OpenStack by measuring productivity at the 252  
release level (while we focus on the release before OpenStack 253  
introduces onboarding). Even though their work is not directly 254  
related to onboarding at the SECO-level, it, however, explores 255

205 contributors' paid and volunteered productivity, which, in  
206 our case, refers to project-level mentoring for onboarding  
207 contributors.

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

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

### 217 III. OBSERVATIONAL STUDY OF TOP-LEVEL SECO 218 ONBOARDING PHASE

#### 219 A. Study Design

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

224 In particular, we aim at investigating the following prelimi-  
225 nary research questions:

- 226 – **PRQ<sub>1</sub>**: What (and how) are the topics taught during a  
227 SECO onboarding event?
- 228 – **PRQ<sub>2</sub>**: What are the challenges involved with organizing  
229 and sustaining a SECO onboarding program?
- 230 – **PRQ<sub>3</sub>**: What are the benefits of a SECO onboarding  
231 program?

232 We describe this observational study's design and present  
233 the results of the PRQ<sub>1</sub> below. Meanwhile, we will discuss the  
234 results of PRQ<sub>2</sub> and PRQ<sub>3</sub> in section IV.

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

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

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

255 At any given instance, each of the 12 tables has six  
256 participants and a mentor, with at least one mentor leading a  
257 task or an exercise. Participants are encouraged to choose their  
258 seats and team members freely. Besides the high-quality audio-  
259 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 tran-  
scribed audio-visual recordings and field notes of all the 72  
participants. The first and second author used a combina-  
tion 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 cod-  
ing 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<sub>1</sub>: 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

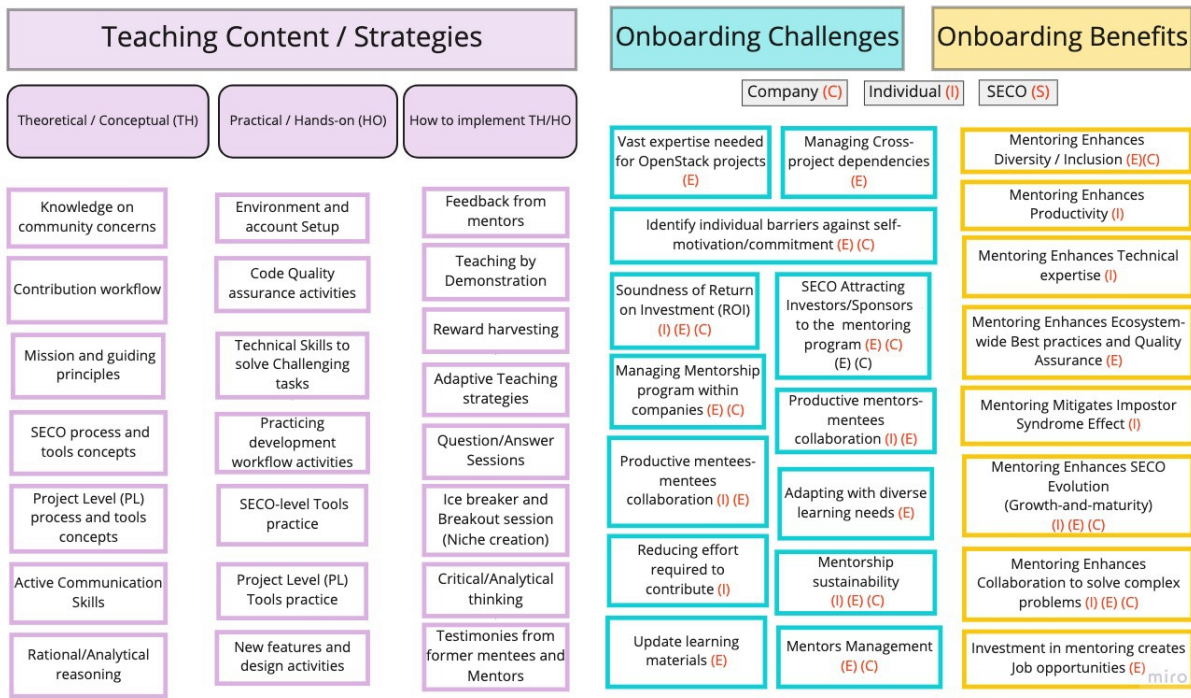


Fig. 1. Materials taught during onboarding and their observed impacts on individual mentees (I), the SECO (E), and companies in the SECO (C).

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

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

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

341 The HO component provides participants with a walk-  
 342 through of typical real-world scenarios and tasks that Open-  
 343 Stack contributors face regularly. The HO component starts  
 344 with the necessary steps of creating accounts with the Open-  
 345 Stack foundation, Gerrit (code review tool), storyboard (issue

346 tracker), mailing-list, and IRC channels (for communication).  
 347 Mentors also guide participants to install and configure their  
 348 (virtual) working environment, which comprises a Virtual-  
 349 Box with possibly a Ubuntu image pre-installed, a copy of  
 350 the OpenStack development environment (aka DevStack on  
 351 Sandbox), issue trackers such as Launchpad and Storyboard,  
 352 the code review environment (Gerrit), and git. Moreover, the  
 353 OpenStack Sandbox environment (repository) provides virtual  
 354 servers for testing OpenStack projects/functionalities in an iso-  
 355 lated environment. Also, mentors ask participants to perform  
 356 tasks of varying difficulties covering technical areas. Such  
 357 as documentation, implementing new features, tracking issues  
 358 (using storyboard/Launchpad), reviewing source code, best  
 359 practices on commit messages and code quality, and CI/CD.  
 360 **OB1** asked a participant to think aloud while performing a HO  
 361 task: “I want to run several unit test cases and an integration  
 362 test. I use the ‘tox framework’ to run unit testing, so I call the  
 363 ‘tox’ command on my terminal [ typing ... ]”(P7)

364 **Mentors use a variety of teaching strategies that facili-**  
 365 **tate collaboration and competitiveness among participants**  
 366 **throughout the training sessions (Fig. 1).** These strategies  
 367 enhance participants’ understanding of the teaching content by  
 368 making the sessions interactive. The most observed strategies  
 369 include the following:

370 Ice-breaker and breakout session. Training ses-  
 371 sions begin with an introductory activity by both mentors and  
 372 participants to create an atmosphere of familiarity that facili-  
 373 tates collaboration among participants (novice-novice collab-  
 374 oration) and mentors (novice-expert collaboration). Breakout  
 375 sessions during the event further strengthen this collaboration.  
 376 Expert-novice feedback. Mentors usually use this

377 strategy to teach practical skills that require a “trial-and-error”  
378 approach. Therefore, they allow participants to make several  
379 attempts, while the mentors keep providing constructive feed-  
380 back until the participants arrive at the answer.

381 Teaching by demonstration. Mentors often demon-  
382 strate how things work while explaining the underlying con-  
383 cepts; this approach enriches participants with confidence  
384 towards the mentors and the ecosystem.

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

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

395 Participants were mostly positive regarding the strategies,  
396 which mentors used. **P48** said: “*I like the hands-on sec-  
397 tion most and, of course, the sticker prizes.*”, besides, other  
398 participants appreciated different strategies differently: “*The  
399 testimony on mentoring was great! I love it.*”(P15) Meanwhile,  
400 **P31** congratulates the strategy and know-how of the mentors:  
401 “*Mentors were great inspirations and knew their stuff well.*”  
402 Also, mentors use real-life scenarios to explain difficult con-  
403 cepts: “*I admired the explanations of different projects and  
404 how they form an ecosystem.*”(P1)

#### 405 IV. PERCEIVED CHALLENGES AND BENEFITS OF SECO 406 ONBOARDING

407 Based on the observed onboarding activities shown in  
408 Figure 1, 13 challenges and 8 benefits emerged. During our  
409 observation, 3/13 challenges and 5/8 benefits encountered  
410 substantially more and deeper discussions than others, leading  
411 to significantly more words in the transcriptions of the audio-  
412 visual recordings. Below, we discuss in detail these three  
413 challenges (PRQ<sub>2</sub>) and five benefits (PRQ<sub>3</sub>).

414 *A. PRQ<sub>2</sub>: What are the challenges involved with organizing  
415 and sustaining a SECO onboarding program?*

##### 416 Challenge 1: Vast expertise needed for SECOs

417 Onboarding at the SECO-level has several challenges  
418 beyond the project-level onboarding. In particular, since  
419 a SECO is not just the union of hundreds of smaller  
420 projects but involves the collaboration of hundreds of cross-  
421 project teams with diverse interacting technologies (see the  
422 cross-project dependencies challenge). Given that  
423 the onboarding participants do not know the different SECO  
424 projects, the initial onboarding event cannot make any assump-  
425 tions. It should target the overall SECO contribution process.  
426 To cover a wide variety of topics and tools (see PRQ<sub>1</sub>), this  
427 also implies that mentors should have polyvalent skill-sets to  
428 guide the participants: “*Be prepared with the ‘deep dives’ exer-  
429 cise. Usually, participants have very different levels of knowl-  
430 edge and skill-set.*”(M2) (which in turn impacts mentorship

sustainability. Furthermore, there should be ongoing 431  
communication between the SECO-level onboarding process 432  
and the onboarding process within individual projects of 433  
the SECO (see mentorship within companies), for 434  
example, to update learning materials to project- 435  
level developments. 436

##### 437 Challenge 2: Self-motivation and commitment

438 It is challenging for SECO to identify individual bar- 439  
riers against self-motivation/commitment. Therefore, active 440  
participation in an onboarding experience is tantamount to a 441  
successful outcome, hence every stakeholder should be fully 442  
involved and committed. “*Successful mentoring requires active  
443 commitment both from the mentor and the mentee.*”(M9), also,  
444 another mentor advocates “*People learn in different ways  
445 at different speeds, which means a commitment to active  
446 mentoring requires more than a handful of quick IRC chats.*”  
447 (M7) This challenge has direct links to the adapting with  
448 diverse learning needs challenge.

##### 449 Challenge 3: Mentorship sustainability

450 SECOs and companies face challenges finding available 451  
mentors to guide mentees. This is partly because of challenge 452  
1 above, and partly because mentoring requires substantial 453  
effort to prepare and keep material up-to-date. Constrained 454  
companies may prefer to prioritize their experts’ time on tasks  
455 that will bring more financial profit to the company, at the  
456 detriment of supporting mentees. At the observed onboarding  
457 event, participants were briefed that “*If there aren’t enough  
458 mentors on every table, ... float around the room checking in  
459 on people, especially during exercises.*” (M1)

460 *B. PRQ<sub>3</sub>: What are the benefits of a SECO onboarding  
461 program?*

##### 462 Benefit 1: Mentoring Enhances Diversity.

463 **Gender diversity (GD):** out of the 72 participants at the 464  
observed onboarding event, 17 (23.6%) declared themselves 465  
as female, 23 (31.94%) as non-binary, and 32 (44.44%) as 466  
male. Moreover, for **corporate diversity (CD)**, we found  
467 evidence of different companies involved with OpenStack and  
468 sponsoring events, and hiring both Cat-2 and Cat-3 contribu-  
469 tors. We also observed that mentors and participants had di-  
470 verse technical skill-sets that cut across different project/cross-  
471 project teams. Such, **technical diversity (TD)** brings value  
472 to the SECO since it “*drives cross-project teams forward  
473 through more mixed reviews, contributions, and viewpoints.*  
474 *By expanding that diversity, we’re able to develop a variety  
475 of opinions for the open infrastructure project as a whole,  
476 ultimately.*” (M9)

##### 477 Benefit 2: Mentoring Enhances productivity.

478 During onboarding, mentors assign real-life exercises and 479  
tasks to participants, such as creating patch sets, fixing bugs,  
480 testing and CI/CD (Zuul), and submitting new features and  
481 documentation. All 72 participants actively participated in  
482 the coding activities and successfully submitted acceptable  
483 commits. This not only trains the participants in the field,  
484 but also encourages them to adopt a collaborative workflow

485 (often by themselves), both with other participants (novice-  
 486 novice) and with mentors (experts-novice). OB1 observed how  
 487 “mentors were pairing participants to work on exercises, i.e.,  
 488 P33 and P35 seated on table/group 10, were exchanging ideas  
 489 constantly throughout this exercise.” Moreover, M11 asked  
 490 participants to: “run different test cases in each project that  
 491 you cloned. ‘If you need help, mentors are seated on your  
 492 tables, and they will assist you in running the test cases.’”

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

503 **Benefit 4:** Overcoming imposter syndrome effect.  
 504 New contributors to an ecosystem often feel overwhelmed  
 505 and inadequate, preventing them from collaborating freely  
 506 with the other contributors in the ecosystem perceived as  
 507 having more talent [30]. Thus, it is important for SECOs to  
 508 take measures to ensure that they help participants to identify  
 509 and start fighting/eliminating the imposter syndrome [31],  
 510 [32]. “As a new developer fresh out of college, coming into  
 511 any new team can be very intimidating. Everyone around you  
 512 knows so much more than you, and you feel that you’re an  
 513 imposter with so much to learn ... ”(P1). The onboarding  
 514 program is aware of the effects of the imposter syndrome  
 515 and sensitizes participants to overcome those, especially by  
 516 letting mentors and past mentees share their experiences.

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

## 527 V. QUANTITATIVE VALIDATION OF PERCEIVED BENEFITS

528 In this section, we empirically evaluate the extent to which  
 529 onboarding can achieve the three major perceived benefits  
 530 identified in PRQ3. We could quantify and measure these  
 531 three benefits by studying 84 months of historical contributions  
 532 (code changes, issue reports, and code reviews) in the Open-  
 533 Stack SECO. Indeed, we measure Diversity<sup>2</sup>, Productivity,  
 534 and Quality. Specifically, we investigate these three research  
 535 questions:

- 536 – RQ1: Does onboarding correlate with SECO diversity?

<sup>2</sup>To measure Gender diversity at OpenStack, contributors’ identity is not publicly available for confidentiality purposes.

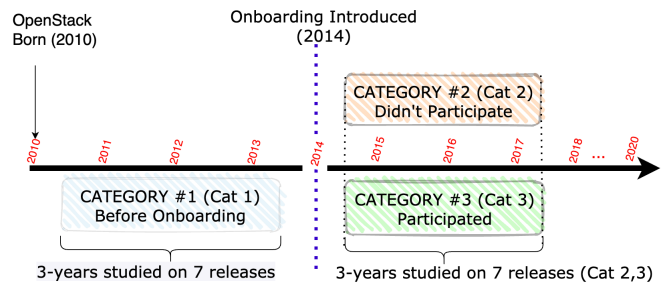


Fig. 2. Timeline of stratified categories used in our study. Cat-1 is our control group, while Cat-2 and Cat-3 are the experimental groups. Each group uses data of seven OpenStack releases (42 months).

- RQ2: Does onboarding correlate with new contributors’ productivity? 537
- RQ3: Does onboarding correlate with new contributors’ code quality? 538

### 539 A. Study Design 540

541 **Categorization of Contributors.** OpenStack’s onboarding  
 542 program is publicly advertised, with free training events (travel  
 543 support is available) taking place in different countries. Hence,  
 544 anyone is encouraged to do onboarding, not just people who  
 545 could afford the travel expenses. Therefore, to measure the  
 546 impact of onboarding on the OpenStack SECO, we considered  
 547 three categories of contributors in our study (see Fig. 2).  
 548 The first category (Cat-1) constitutes contributors who joined  
 549 OpenStack before onboarding events were introduced and  
 550 could not benefit from any official onboarding. The second  
 551 category (Cat-2) comprises new contributors who did not  
 552 participate in any onboarding event, even though the onboarding  
 553 program did exist when joining OpenStack. Finally, the  
 554 third category (Cat-3) are contributors who participated in the  
 555 onboarding program. 556

557 Each of the three categories plays an essential role in  
 558 our study. In particular, for each RQ and metric, we first  
 559 compare the distribution of the metric values between Cat-  
 560 2 and Cat-3. If significant differences exist, we perform a  
 561 second comparison between Cat-1 and Cat-2 to control for  
 562 any confounding factors such as changes in the development  
 563 process that were put in place simultaneously when OpenStack  
 564 introduced the onboarding program. Only if no significant  
 565 changes exist between Cat-1 and Cat-2 (both of whom consist  
 566 of contributors who did not do onboarding) can we correlate  
 567 the differences between Cat-2 and Cat-3 with the introduction  
 568 of onboarding.

569 **Data Collection.** Given the three categories of contributors  
 570 (Cat-1, Cat-2, and Cat-3), we first use the clustered random  
 571 sampling technique [33] to randomly select Cat-3 first-time  
 572 contributors who joined through the OUI onboarding program  
 573 from different geographic areas, reflecting the distributed  
 574 nature of SECOs in our sample. This yielded 427 Cat-3 partic-  
 575 ipants across seven OpenStack releases, from Juno to Pike  
 576 (Fig. 5). Then, we used random sampling to select an equal  
 577 number of individuals in Cat-1/2. For those two categories,  
 578 we made sure to exclude any OpenStack contributor who

579 later on (after making contributions) decided to participate in  
 580 onboarding (720 exclusions).

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

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

594 Our study analyzes these metrics at the individual contrib-  
 595 utors’ level, then aggregates them to the SECO-level, split  
 596 across the three categories of contributors. Some metrics are  
 597 general, while others (like Technical and Corporate diversity)  
 598 are SECO-specific. Note that there is only a weak Pearson cor-  
 599 relation of 0.324 between Effort and TFC, i.e., they measure  
 600 different phenomena.

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

608 For other metrics, we use the Kruskal-Wallis H-test (KW-  
 609 H) [38] to compare metric distributions of the three contrib-  
 610 utor categories at once. In case of a statistically significant  
 611 difference ( $\rho < \alpha(0.01)$ ), a Dunn (posthoc) test [39] is  
 612 used to identify which of the three categories has a different  
 613 distribution of metric values. As such, Dunn evaluates Cat-1  
 614 vs. Cat-2, Cat-1 vs. Cat-3, and Cat-2 vs. Cat-3. Finally, we  
 615 measure the effect size (Cliff’s delta) [40], which quantifies  
 616 the effect of significant differences. As explained earlier, we  
 617 expect that if onboarding correlates with a change in, say, a  
 618 productivity metric, then Cat-1 (the control group) and Cat-  
 619 2 (treatment group) should have no statistically significant  
 620 difference. In contrast, there should be a statistically significant  
 621 difference between Cat-3 and Cat-2 (and, hence, Cat-1).

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

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

629 *1) Gender Diversity: We observed a statistically signifi-*  
 630 *cant increase of 65%, with large (L) effect sizes, in terms*  
 631 *of contributors declared as either female or non-binary*  
 632 *within Cat-3 (compared with Cat-2), at the expense of*  
 633 *contributors who reported male gender [20]. Fig. 3a shows*

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)

634 how the percentage of contributors who declared themselves  
 635 as female increased to 33% compared to the 18% (20%) values  
 636 for Cat-1 (Cat-2). Similarly, for contributors who declared  
 637 themselves non-binary, the percentage significantly increased  
 638 from 7% (10%) to 23%.

639 The main reason for these increases seems to be the fact  
 640 that a significantly smaller proportion of contributors explicitly  
 641 declared themselves as having male gender, which thus far  
 642 has been the over-represented gender in open source devel-  
 643 opment [44]. There are different interpretations possible. The  
 644 most likely explanation, supported by the fact that we did not  
 645 find a significant difference in gender between Cat-1 and Cat-  
 646 2, is that onboarding helped to attract a larger proportion of  
 647 contributors of female gender, while providing confidence to  
 648 others to declare themselves as non-binary instead of sticking  
 649 to a binary gender. Self-disclosed gender at OpenStack [20]  
 650 is not open to the general public; it is available in internal  
 651 profiles for confidentiality purposes. However, there could still  
 652 be confounding factors. For example, maybe contributors with  
 653 male gender are less likely to participate in onboarding events.  
 654 More research is needed to better understand this.

655 *2) Technical Diversity (TD): People who followed on-*  
 656 *boarding (irrespective of gender) are more polyvalent than*

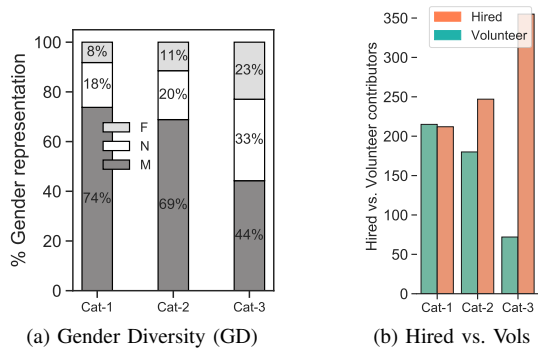


Fig. 3. (Left) Median GD (in %) of each category ((F)emale, (M)ale, and (N)on-binary) ; (Right) Hired vs. volunteer (Vols) contributors in Cat-1, Cat-2, and Cat-3.

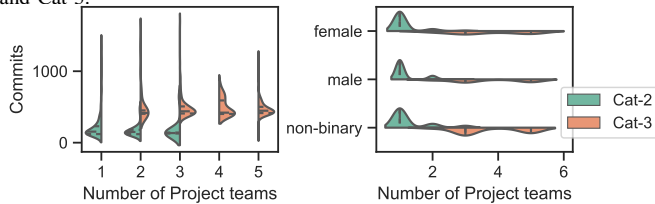


Fig. 4. Overview of technical diversity, showing the number of commits made across different numbers of projects (Left) and the number of projects people contribute to per declared gender (Right).

657 **other contributors.** Technical/code diversity measures the  
 658 number of distinct projects (modules) to which a developer  
 659 contributes source code. Fig. 4 shows that people who joined  
 660 OpenStack without onboarding (Cat-1, not shown, & Cat-2)  
 661 contribute to at most three projects, whereas people who joined  
 662 through onboarding (Cat-3) often are contributors in more than  
 663 three projects. For example, in Cat-2, 82.7% of individuals  
 664 contribute to only one project, 16.6% contribute to two, and  
 665 only 0.7% contribute to three projects; only contributors with  
 666 non-binary or male gender contributed to two or more projects  
 667 in Cat-2. On the other hand, in Cat-3, 52.7% contribute to  
 668 three core projects, 31.9% contribute to four projects, and  
 669 15.5% to five or more projects; contributors who declared  
 670 male or non-binary gender are mostly contributing to three and  
 671 four projects, while contributors who declared female gender  
 672 are even contributing to five or more projects (significant  
 673 difference between female and other genders). This supports  
 674 our earlier findings about gender diversity (Section V-B1).

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

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

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

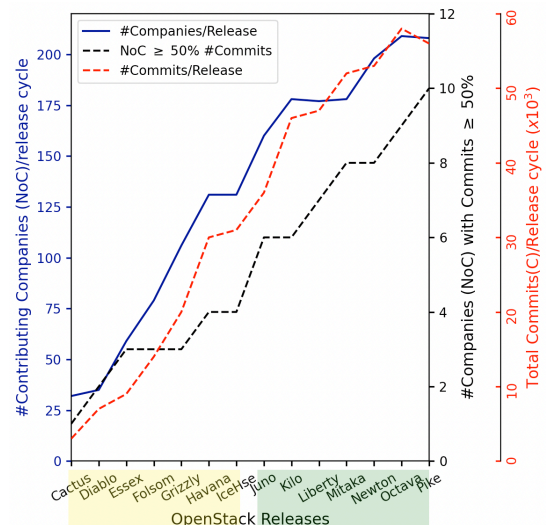


Fig. 5. The evolution of the number of companies (NoC, solid blue line) for each of the 7 studied OpenStack releases before (Cat-1, yellow) and after (Cat-2/3, green) the introduction of onboarding. The black dashed line represents the top NoC responsible for 50% of a release’s commits and the red dashed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

689 mit across those companies. **Also, we found that none of**  
 690 **the sponsoring companies (NoC) had a disproportionate**  
 691 **amount of contribution either by Cat-2 or Cat-3 contribu-**  
 692 **tors. Furthermore, 83% of Cat-3 contributors were hired**  
 693 **by companies compared to 51% of Cat-2 contributors,**  
 694 and this difference is statistically significant with a  $p$ -value  
 695 of  $3.006 \times 10^{-40}$  and a large (L) effect size. We also observed  
 696 that no single company has a dominating share of contributors  
 697 (and contributions).

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

703 **Only 13% of Cat-3 contributors were hired by the com-**  
 704 **panies that sponsor the onboarding events** 70% of the 83%  
 705 hired Cat-3 contributors were employed by companies within  
 706 OpenStack that do not sponsor onboarding (median days to  
 707 hire for Cat-3 is 33.0 vs. 212.0 for Cat-2). While, overall,  
 708 the high percentage of 83% is positive for the ecosystem  
 709 as a whole, the sizeable proportion of contributors hired by  
 710 non-sponsoring companies could be interpreted as a form of  
 711 “brain drain” and “low return of interest” for the companies  
 712 organizing the onboarding training.

713 On a positive note, though, we observe that seven of the top<sup>3</sup>  
 714 10 Cat-3 contributors in the SECO were hired by sponsoring  
 715 companies, which improves their onboarding ROI. On the  
 716 other hand, Cat-3’s hired contributors switch more easily from  
 717 one company to another. This could indicate that the expertise  
 718 of Cat-3 contributors is useful and sought-after in different

<sup>3</sup>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.



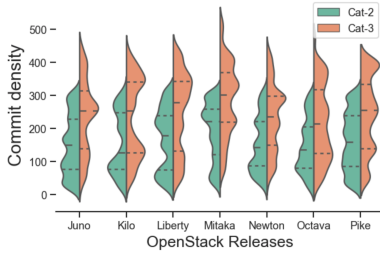


Fig. 6. Comparison of commit density between Cat-2 and Cat-3.

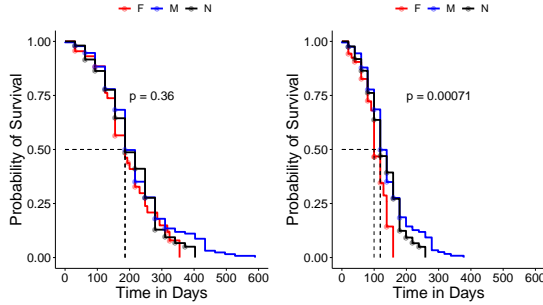


Fig. 7. Survival curves for time until the first accepted contribution per gender in Cat-2 (left) and Cat-3 (right).

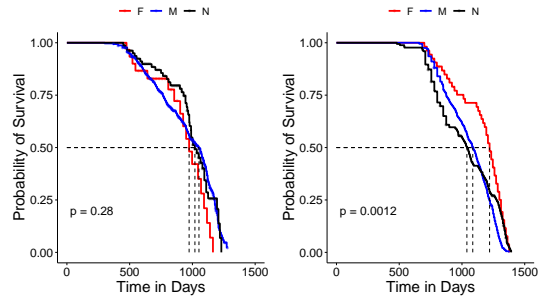


Fig. 8. Survival curves for the time until Cat-2 (left) and Cat-3 (right) contributors leave.

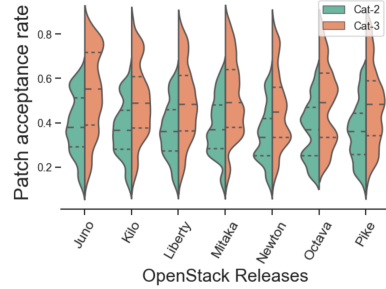


Fig. 9. Comparison of patch acceptance rate between Cat-2 and Cat-3

719 contexts, or could be due to technology transfer between  
 720 ecosystem companies. For example, one given contributor  
 721 started contributing to the Nova project in the Pike release  
 722 cycle with *IBM*, switched to *Huawei* and later to *Futurewei*,  
 723 all between February 20<sup>th</sup> – August 1<sup>st</sup> (2017).

724 *C. RQ2: Does onboarding correlate with new contributors’  
 725 productivity?*

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

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

747 3) *Retention rate: Onboarding correlates with a 16%  
 748 longer average retention rate across the three genders in  
 749 the SECO, i.e., Cat-3 contributors are active much longer  
 750 than Cat-2 (and Cat-1) contributors, which is beneficial for*

751 the sustainability and cohesion within a community. We ob-  
 752 served from the survival analysis chart (Fig. 8) that 95% of  
 753 contributors were active for 450 days in Cat-2 and 750 days  
 754 (four SECO release cycles) in Cat-3. While there is a 50%  
 755 probability of Cat-1 contributors (either gender) abandoning  
 756 the SECO/project within 1,000 days (not shown), this retention  
 757 period is 1,100 (1,000) days for Cat-3 (Cat-2) non-binary  
 758 contributors, 1,150 (1,100) days for males, and 1,290 (950)  
 759 days for females. Therefore, contributors, on average, were  
 760 productive for a significantly longer time in Cat-3 than in both  
 761 Cat-1 and Cat-2 (large effect sizes), with self-declared female  
 762 contributors with onboarding experience persisting longer than  
 763 any other declared gender.

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

773 *D. RQ3: Does onboarding correlate with new contributors’  
 774 code quality?*

775 1) *Effort: Cat-3 contributors require less effort to have  
 776 their commit accepted.* Based on our observation and results  
 777 in Fig. 10, contributors who joined the ecosystem without  
 778 an onboarding training (Cat-1 & 2) on average require sig-  
 779 nificantly more attempts to get their contributions accepted  
 780 than those who were onboarded (Cat-3), with  $\rho$ -value of  
 781  $6.621x10^{-77}$  and large effect size.

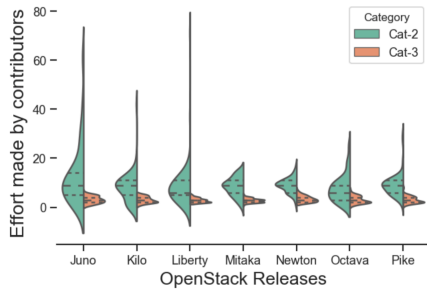


Fig. 10. An Overview of effort needed by Cat-2/3 contributors.

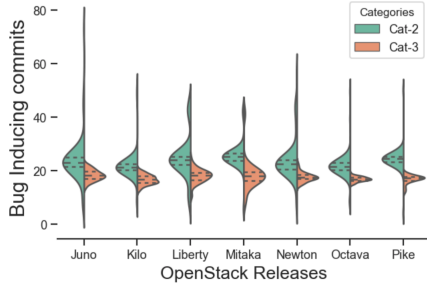


Fig. 11. Likelihood of bug-inducing commits across Cat-2/3.

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

787 2) *Bug-inducing Changes: Contributors who did on-*  
 788 **boarding produce code that is 14% less likely to introduce**  
 789 **bugs.** Using the PyDriller [51] implementation of the SZZ al-  
 790 gorithm [52]–[56], our results show that the median probability  
 791 of a commit introducing a bug is 25% for Cat-2 compared to  
 792 14% for Cat-3 (Fig. 11). In other words, accepted patches are  
 793 less buggy for Cat-3, even though Cat-3 contributors submit a  
 794 higher quantity (with more complexity) of code changes than  
 795 contributors from the other categories (as previously discussed  
 796 in RQ1 for TD). These differences are significant with a  $p$ -  
 797 value of  $4.290 \times 10^{-57}$  and a large effect size. Not only are  
 798 patches of Cat-3 contributors less buggy, they also required  
 799 less attempts to be accepted (see previous metric).

## 800 VI. DISCUSSION

801 Based on the observational study findings (Fig. 1), we  
 802 notice how the themes in the affinity diagram form a  
 803 holistic set of socio-technical activities relevant to onboard-  
 804 ing in a complex SECO. Such onboarding is more than  
 805 giving a tutorial on creating a feature branch or run-  
 806 ning a test suite. Mentors spent substantial effort explain-  
 807 ing the interactions and differences between the Open-  
 808 Stack SECO and the individual projects inside the SECO.  
 809 Knowledge on community concerns is another es-  
 810 sential pillar of the teaching content, as well as activities to  
 811 train participants’ active communication skills and  
 812 rational/analytical reasoning. Combining such  
 813 topics with the more technical hand-on activities requires (i)  
 814 the use of a host of engaging teaching strategies, as well as

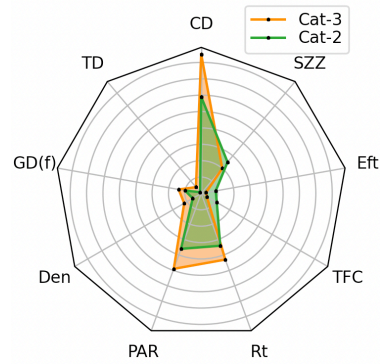


Fig. 12. Radar chart of the studied metrics showing that onboarding (Cat-3) has significant differences and improvements over Cat-2. The metrics are those of Table I: Bug-inducing-commits (SZZ), Effort (Eft), Time to first commit (TFC), Retention (Rt), Patch Acceptance Rate (PAR), Density (Den), Diversity: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

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

## 863 VII. THREATS TO VALIDITY

864 **Construct validity.** This study uses existing diversity, pro-  
865 ductivity, and quality metrics from the literature [2], [35], [57],  
866 [58] and open source communities such as CHAOSS [34].  
867 However, concerning gender, we relied on the self-declared  
868 gender available in OpenStack’s internal profiles [20]. Further-  
869 more, we observed an onboarding event and mined readily  
870 available data from version control, issue reports, and code  
871 review repositories but did not have access to the private  
872 online communication between mentors and mentees after the  
873 onboarding event.

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

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

891 Another threat concerns the effect of unreported bugs on  
892 the result of the SZZ bug-inducing commit analysis, which  
893 uses an implementation of the original SZZ algorithm [52]. To  
894 mitigate this, we run SZZ on the entire history of OpenStack’s  
895 official issue tracking systems (Launchpad/Storyboard). Also,  
896 our study window spans 14 releases (7 for Cat-1 and 7 for Cat-  
897 2/3), which gives ample time for contributors to make active  
898 contributions. We base our study on the assumption that par-  
899 ticipants/contributors had no prior experience with any SECO.  
900 However, since some educational institutions introduce their  
901 students to open-source development concepts and practices as  
902 part of their learning path, this could be a confounding factor  
903 that could affect our results. Since “generic” development  
904 concepts form only a minor part of the onboarding process,  
905 we believe this threat is minimal.

906 **External validity.** While OpenStack is a representative  
907 modern SECO, our results may not generalize to other ecosys-  
908 tems. That said, the methods that we use in our observational

and quantitative studies are ecosystem-agnostic. Hence, prac- 909  
titioners and researchers could use our methods to identify and 910  
evaluate the impact of any ecosystem’s onboarding program. 911  
As a side note, the post-Covid-19 era fosters a culture of online 912  
collaboration that could disrupt the dynamics of in-person [59] 913  
events. Even though Rodeghero et al. [60] studied onboarding 914  
during the Covid-19 pandemic at the project-level, it is still too 915  
early to understand the impact of this on the top-level SECO 916  
training events or the SECO onboarding process as a whole. 917  
For example, the recent OUI training event on October 22-23, 918  
2020, was virtual, yet the turnout was much lower (8 mentors 919  
and 11 participants) than previous events. Future research is 920  
necessary for the new reality of in-person vs. virtual training 921  
events in OSS communities. 922

923 **Reliability validity.** Except for confidential participant in-  
924 formation, we provide the necessary description and resources  
925 (OSS tools and dataset) [28] needed to replicate our research.

## 926 VIII. CONCLUSION

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

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

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

- [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 Proceedings*, 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](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://mpr.aub.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](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*

- 1108 *Software Testing, Verification and Validation (ICST)*, March 2017, pp. 1109 219–229.
- 1110 [41] C. Jergensen, A. Sarma, and P. Wagstrom, “The onion patch: Migration 1111 in open source ecosystems,” in *Proceedings of the 19th ACM SIGSOFT 1112 Symposium and the 13th European Conference on Foundations of 1113 Software Engineering*, ser. ESEC/FSE ’11. New York, NY, USA: 1114 Association for Computing Machinery, 2011, p. 70–80. [Online]. 1115 Available: <https://doi.org/10.1145/2025113.2025127>
- 1116 [42] S. Hönel, M. Ericsson, W. Löwe, and A. Wingkvist, “Importance 1117 and aptitude of source code density for commit classification into 1118 maintenance activities,” in *2019 IEEE 19th International Conference 1119 on Software Quality, Reliability and Security (QRS)*. IEEE, 2019, pp. 1120 109–120.
- 1121 [43] E. C. Neto, D. A. da Costa, and U. Kulesza, “Revisiting and improving 1122 szz implementations,” in *2019 ACM/IEEE International Symposium on 1123 Empirical Software Engineering and Measurement (ESEM)*. IEEE, 1124 2019, pp. 1–12.
- 1125 [44] C. Mendez, H. S. Pedala, Z. Steine-Hanson, C. Hilderbrand, A. Horvath, 1126 C. Hill, L. Simpson, N. Patil, A. Sarma, and M. Burnett, “Open source 1127 barriers to entry, revisited: A tools perspective,” 2017.
- 1128 [45] G. Avelino, E. Constantinou, M. T. Valente, and A. Serebrenik, “On 1129 the abandonment and survival of open source projects: An empirical in- 1130 vestigation,” in *2019 ACM/IEEE International Symposium on Empirical 1131 Software Engineering and Measurement (ESEM)*, Sep. 2019, pp. 1–12.
- 1132 [46] M. Valiev, B. Vasilescu, and J. Herbsleb, “Ecosystem-level determinants 1133 of sustained activity in open-source projects: A case study of the 1134 pypi ecosystem,” in *Proceedings of the 2018 26th ACM Joint Meeting 1135 on European Software Engineering Conference and Symposium on the 1136 Foundations of Software Engineering*, ser. ESEC/FSE 2018. New 1137 York, NY, USA: Association for Computing Machinery, 2018, p. 1138 644–655. [Online]. Available: <https://doi.org/10.1145/3236024.3236062>
- 1139 [47] V. Cosentino, J. L. C. Izquierdo, and J. Cabot, “Assessing the bus factor 1140 of git repositories,” in *2015 IEEE 22nd International Conference on 1141 Software Analysis, Evolution, and Reengineering (SANER)*. IEEE, 2015, 1142 pp. 499–503.
- 1143 [48] I. Steinmacher, I. Wiese, A. P. Chaves, and M. A. Gerosa, “Why 1144 do newcomers abandon open source software projects?” in *2013 6th 1145 International Workshop on Cooperative and Human Aspects of Software 1146 Engineering (CHASE)*, May 2013, pp. 25–32.
- 1147 [49] J. Li, G. Sun, G. Zhao, and H. L. Li-wei, “Robust low-rank discovery 1148 of data-driven partial differential equations,” in *Proceedings of the AAAI 1149 Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 767–774.
- 1150 [50] H. S. Qiu, A. Nolte, A. Brown, A. Serebrenik, and B. Vasilescu, “Going 1151 farther together: The impact of social capital on sustained participation 1152 in open source,” in *Proceedings of the 41st International Conference on 1153 Software Engineering (ICSE) 2019, Montreal, Canada*. IEEE, 2019, to 1154 appear.
- 1155 [51] D. Spadini, M. Aniche, and A. Bacchelli, *PyDriller: Python Framework 1156 for Mining Software Repositories*, 2018.
- 1157 [52] J. Śliwerski, T. Zimmermann, and A. Zeller, “When do changes induce 1158 fixes?” *SIGSOFT Softw. Eng. Notes*, vol. 30, no. 4, pp. 1–5, May 2005. 1159 [Online]. Available: <http://doi.acm.org/10.1145/1082983.1083147>
- 1160 [53] S. Kim, T. Zimmermann, K. Pan, E. James Jr *et al.*, “Automatic iden- 1161 tification of bug-introducing changes,” in *21st IEEE/ACM International 1162 Conference on Automated Software Engineering (ASE’06)*. IEEE, 2006, 1163 pp. 81–90.
- 1164 [54] C. Sadowski, C. Lewis, Z. Lin, X. Zhu, and E. J. Whitehead Jr, “An 1165 empirical analysis of the fixcache algorithm,” in *Proceedings of the 8th 1166 Working Conference on Mining Software Repositories*. ACM, 2011, 1167 pp. 219–222.
- 1168 [55] E. C. Neto, D. A. da Costa, and U. Kulesza, “The impact of refactoring 1169 changes on the szz algorithm: An empirical study,” in *2018 IEEE 1170 25th International Conference on Software Analysis, Evolution and 1171 Reengineering (SANER)*. IEEE, 2018, pp. 380–390.
- 1172 [56] G. Rodríguez-Pérez, G. Robles, and J. M. González-Barahona, “Repro- 1173 ducibility and credibility in empirical software engineering: A case study 1174 based on a systematic literature review of the use of the szz algorithm,” 1175 *Information and Software Technology*, vol. 99, pp. 164–176, 2018.
- 1176 [57] M. Hilton and A. Begel, “A study of the organizational dynamics of 1177 software teams,” in *2018 IEEE/ACM 40th International Conference on 1178 Software Engineering: Software Engineering in Practice Track (ICSE- 1179 SEIP)*, May 2017, pp. 191–200.
- 1180 [58] M. Borg, O. Svensson, K. Berg, and D. Hansson. (2019, 03) Szz 1181 unleashed: An open implementation of the szz algorithm - featuring 1182 example usage in a study of just-in-time bug prediction for the jenkins 1183 project.
- 1184 [59] S.-F. Wen, “Learning secure programming in open source software 1185 communities: A socio-technical view,” in *Proceedings of the 6th 1186 International Conference on Information and Education Technology*, 1187 ser. ICIET ’18. New York, NY, USA: Association for Computing 1188 Machinery, 2018, p. 25–32. [Online]. Available: <https://doi.org/10.1145/3178158.3178202>
- 1189 [60] P. Rodeghero, T. Zimmermann, B. Houck, and D. Ford, “Please turn 1190 your cameras on: Remote onboarding of software developers during a 1191 pandemic,” *arXiv preprint arXiv:2011.08130*, 2020. 1192