Hidden Figures: Roles and Pathways of Successful OSS Contributors

BIANCA TRINKENREICH, Northern of Arizona University, USA
MARIAM GUIZANI, Oregon State University, USA
IGOR SCALIANTE WIESE, Federal University of Technology–Paraná, Brazil
ANITA SARMA, Oregon State University, USA
IGOR STEINMACHER, Northern of Arizona University, USA

Open Source Software (OSS) development is a collaborative endeavor where expert developers, distributed around the globe create software solutions. Given this characteristic, OSS communities have been studied as technical communities, where stakeholders join and evolve in their careers based on their (often voluntary) code contributions to the project. However, the OSS landscape is slowly changing with more people and companies getting involved in OSS. This means that projects now need people in non-technical roles and activities to keep the project sustainable and evolving. In this paper, we focus on understanding the roles and activities that are part of the current OSS landscape and the different career pathways in OSS. By conducting and analyzing 17 interviews with OSS contributors who are well known in the community, we provide empirical evidence of the existence and importance of community-centric roles (e.g advocate, license manager, community founder) in addition to the well-known project-centric ones (e.g maintainer, core member). However, the community-centric roles typically remain hidden, since these roles may not leave traces in software repositories typically analyzed by researchers. We found that people can build a career in OSS through different roles and activities, with different backgrounds, including those not related to writing software. Furthermore, people's career pathways are fluid, moving between project and community-centric roles. Our work highlights that communities and researchers need to take action to acknowledge the importance of these varied roles, making these roles visible and well-recognized, which can ultimately help attract and retain more people in the OSS projects.

CCS Concepts: • **Software and its engineering → Open source model**; • **Human-centered computing → Open source software**; • **Information Systems →** *Open Source software*.

Additional Key Words and Phrases: open source; career; role; collaborative development

ACM Reference Format:
Bianca Trinkenreich, Mariam Guizani, Igor Scaliante Wiese, Anita Sarma, and Igor Steinmacher. 2020. Hidden Figures: Roles and Pathways of Successful OSS Contributors. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW2, Article 180 (October 2020), 22 pages. https://doi.org/10.1145/3415251

1 INTRODUCTION

Authors' addresses: Bianca Trinkenreich, Northern of Arizona University, 1 Thørväld Circle, Flagstaff, AZ, USA, bt473@nau.edu; Mariam Guizani, Oregon State University, Corvalis, OR, USA, guizanim@oregonstate.edu; Igor Scaliante Wiese, Federal University of Technology–Paraná, Campo Mourão, PR, Brazil, igor@utfpr.edu.br; Anita Sarma, Oregon State University, Corvalis, OR, USA, anita.sarma@oregonstate.edu; Igor Steinmacher, Northern of Arizona University, Flagstaff, AZ, USA, igor.steinmacher@nau.edu.

2573-0142/2020/10-ART180 $15.00
https://doi.org/10.1145/3415251

In the last two decades we have seen the emergence and adoption of open collaboration communities [24] with a peer-production model [10, 21]. In this model, large communities, distributed across the globe, collaborate to create knowledge-intensive goods mediated by collaborative platforms [24, 42]. These communities offer new opportunities for people to form ties with others and create things together. Examples of these communities include Wikipedia, some Crowdsourcing initiatives, and Open Source Software (OSS) projects.

Specifically for OSS, collaboration is evolving from comprising a group of talented software hackers volunteering to produce high-quality software into a more mainstream and commercially viable model [23, 47, 53]. This new OSS landscape involves well known companies, which are not only using OSS, but also open-sourcing their products, and joining and managing communities [47]. Steinmacher et al. [53] recently referred to this as marking the end of OSS's teenage years.

These studies allude to a linear "joining script" [57] or an onion-model [64]—based on the Legitimate Peripheral Participation theory [34]—in which individuals sequentially achieve more central roles as they gain experience (and visibility) in the project. However, paths to becoming successful in OSS are not always linear or straightforward, especially when a project exists in a larger ecosystem (e.g., Gnome, KDE) in which contributors "short circuit" the script by leveraging prior experience [31]. Thus, there is a need to expand our understanding beyond the simplistic notion of a technical core-periphery contribution structure [7].

The term "role" can have multiple definitions. In this work, we defined roles to encompass a set of activities performed by individuals within a set of organizational processes, and following a set of expected behavior [45] that are beneficial for coordinating work [6]. While the primary focus of existing research is on the roles related to source-code activities and how developers evolve within these roles (e.g., how one becomes a committer, a repository maintainer, and a long-term contributor), OSS projects in fact rely on people playing different roles and performing different activities, without which a project could not be what it is [15]. While some of these roles are visible (e.g., coders, code-reviewers, or testers), others are not yet well-recognized or seen in the OSS context [40] (e.g., advocates, license managers, and mentors). Little is known about how these non-coding OSS contributors join, sustain (or increase) their participation, take on additional responsibilities, and build a career in OSS projects. These roles are currently poorly explored in the literature. Thus, in this study we aim to investigate the different career pathways and roles of OSS contributors from the perspective of successful and recognized stakeholders. Such a perspective is important because the journeys of these stakeholders may uncover the different roles and pathways related to OSS project success. The following Research Questions guided our study:

RQ1. What different roles can be played by successful and recognized contributors to OSS?
RQ2. How do roles evolve for such successful and recognized OSS contributors?

To answer our RQs, we conducted 17 semi-structured interviews with well-known OSS contributors who were invited speakers at the Open Source Software Conference 2019 (OSCON), along with a snowball approach to recruit more interviewees. We qualitatively analyzed our interviews and recreated the career path of each interviewee as described, in order to understand how they joined, which roles and activities they performed, and how they reached their current position in OSS. We verified these roles and pathways via member checking with five participants.

By answering our RQs, we contribute to the current literature on open collaboration communities by providing: (i) empirical evidence of diverse pathways that are followed by successful OSS contributors; (ii) findings about how coders and non-coders can follow an OSS career; and (iii) a discussion of the hidden roles of OSS contributors who are not coders.

2 RELATED WORK

In recent years, online peer production communities have been the subject of multiple studies. Identifying different roles within collaborative communities [1] has been one way to understand a community's workflow. Multiple research studies have investigated the different roles within the Wikipedia platform [5, 6, 37, 63]. For example, Arazy et al. [7] identified 12 distinct editor's roles in Wikipedia using contributors' access privileges and described their migration both to and from more advanced roles. In Piscopo and Simperl's recent study [44], the investigators looked at user's activity patterns to deduct different user roles within Wikidata. Similar inquiries explored the roles and activities undertaken in Q&A and discussion forums [33, 36, 59]. With the growing importance of user-centric design, the online co-innovation processes of designing new products where companies invite users to participate are starting to become more prevalent. Guo et al.[26] depict six different roles within these communities from the leader and active designer to the observer. In an attempt to better understand Open Government Platforms (OGP), Koch et al. [32] performed an exploratory study on the typical roles in OGP and their evolution through time.

As an online peer production ecosystem, open source encapsulates various forms of joining and participation. Prior research has described the joining process as one that follows the onion model [41]. Newcomers start as observers and passive users (outermost layer), eventually moving up the the hierarchy to become contributors and then core members (innermost layer). The inner layers the model are associated with more technical and higher reputation roles and the movement across the layers is referred to as a migration [30, 64]. Since then multiple variations of the onion model have been proposed [19, 30, 39, 48, 57]. Jergensen et al. [31] expand on the onion model by investigating large project ecosystems where contributors' overlap between projects was significant. This overlap often resulted in contributors skipping the step of building a technical identity within a community as that step was already done in a previous project and diving straight into more central roles.

Several research studies have investigated the activities and roles of developers within the onion model. Barcellini et al. [9] divided the OSS roles into a "discussion space" and an "implementation space". Yu and Ramaswamy [65] used data mining to classify developers' roles as core and associate members. Similarly, Joblin et al. classified core and peripheral developers using network modeling as an alternative to count-based models (e.g number of commits, number of code reviews etc.). Xu et al. [62] proposed a classification similar to the onion model—without considering a migration path—composed of four roles: project leaders, core developers, co-developers, and active users. Spauwen and Jansen [51] investigated developer roles within four OSS browser extensions and their prevalence within each platform. A case study of three OSS projects (Mozilla.org, Apache.org and NetBeans.org) [30] modeled the different roles present within each project, including the figure of a community manager— whose duties were more related to triaging and team coordination: "ensures that issues brought up on mailing lists are addressed fairly." Von Krogh et al. [57] categorized OSS project members into three groups (joiners, newcomers and developers). Di Bella et al. [18] used data from the software repository to classify the contributors into core developers, active developers, occasional developers, and rare developers.

The above studies provide interesting insight into examples of roles within a single project, activities present within the layers of the OSS onion model, automated ways to classify core and peripheral developers. Since the main premise of the onion model focuses on the steps from a passive user to an active core developer, these studies mainly focused on project-centric roles and activities. In fact, even the socialization activities in the outer layers (mailing list activities, bug reports, documentation) are considered a temporary initial step (a "rite of passage") where the objective is to increase ones familiarity with the project and its social norms as opposed to being a

role in itself. Duchenaut [19] identified successful trajectories based on individual OSS developers' stories, which again ranged from peripheral monitoring to being recognition by core members and gaining access to more complex architectural changes.

Other researchers have investigated role migration mining software repositories. Cheng and Guo [17], for example, analyzed the actions (commits, comments, pull-requests, issue reports, mentions, and admin actions) of contributors in 29 GitHub projects. By using automated clustering analysis, they identified four active roles and five supporting roles, according to their activity level related to each action. They found that, although the roles identified can be mapped to the onion model, the activities played by each role and the migration between roles are not as simplistic as those depicted by the onion model. Besides, Cheng et al. [16] applied the classication by Xu et al. [62] (project leaders, core developers, co-developers, and active users) to investigate the factors that influence the evolution of developer roles in the GNOME ecosystem. Such analysis of data from software repositories may miss (non code) contributors who do not leave traces in the version control or issue trackers.

Now that the open source community has drastically matured, as indicated by projects that involve a huge number of volunteer and paid contributors, as well as industry partnership, non project-centric roles can be vital an open source project. This paper aims to take a closer look at these "behind the scene" roles that have evolved over the years to become ingrained in the project workflow and essential to its success. To the best of our knowledge, there exists no research on the alternative pathways to success in open source—the hidden pathways that are not centered around code nor are temporary "rites of passage".

3 RESEARCH PLANNING AND EXECUTION

Here we present our study design (Section 3.1), data collection (Section 3.2), and analysis procedures (Section 3.3), as depicted in Figure 1.

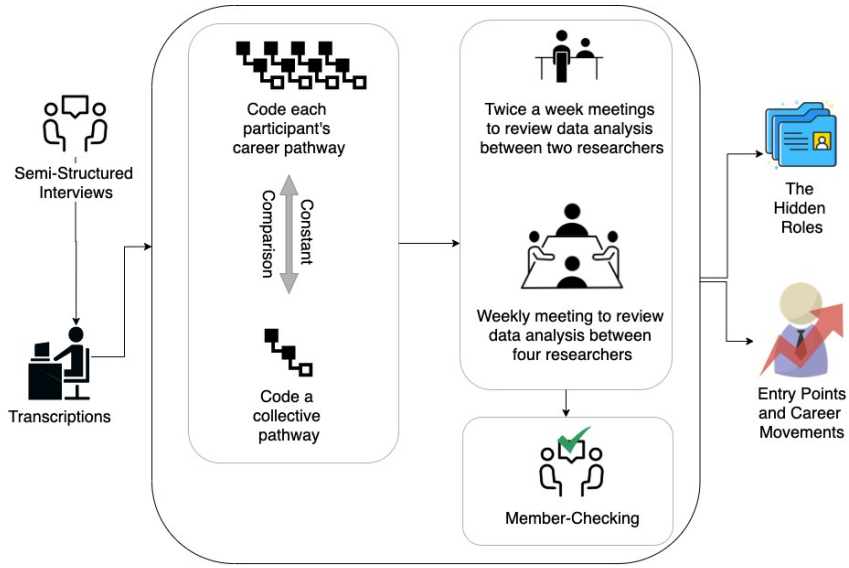

Fig. 1. Research method overview

3.1 Participants

Our goal was to investigate successful OSS career stories, therefore, we recruited OSS contributors who are experienced and well-recognized. More specifically, we recruited the invited speakers at the Open Source Software Conference 2019 (OSCON) in Portland, OR, USA, a well-recognized Open Source conference for practitioners. As these speakers were invited to give talks at the main track, they are arguably successful in OSS. We emailed and sent direct messages via Twitter to all panelists whose contact information was publicly available. We also approached some of them during the event. We conducted face-to-face interviews with 11 people who agreed to be interviewed. In addition, we used a snowball approach to recruit more interviewees. At the end of each interview we asked them to introduce us to other qualified participants for the study. We conducted 6 additional interviews. Table 1 shows the participant demographics. Participants comprised paid and volunteer contributors across 15 different OSS projects (e.g., kubernetes, drupal, R, noosfero, SPDX, envoy) across a wide range of: (1) contributors—ranging in size from 30 to 3,000 contributors, (2) product domains—including infrastructure and user-application projects, and (3) types—backed by foundations, communities, and companies (e.g., Microsoft, Linux Foundation, Google, Red Hat, IBM, Drupal, Bitergia, Apache).

Two researchers conducted the interviews, which were face-to-face during the conference and over video conference calls for the rest. For those interviewed on-line, we arranged a convenient meeting date and time and sent out a consent letter by e-mail, to collect their consent. We compensated all interviewees with a 25-dollar gift card as a token of appreciation.

Our sample—composed of 17 participants—represents a group of professionals who are successful in OSS and recognized by the community, either by being invited to give talks in well-known OSS events or for maintaining popular projects. This is in line with the anthropology literature, which mentions that a set of 10-20 knowledgeable people is sufficient to uncover and understand the core categories in any cultural domain or study of lived experience [11].

Table 1. Demographics for the Interview Participants

Participant ID	Gender	Years in OSS	Experience as Mentor	Academic Degree
P1	Woman	6	Yes	Bachelor
P2	Woman	5	No	Bachelor
P3	Woman	13	No	Post Bachelor
P4	Man	9	No	Bachelor
P5	Prefer not to disclose	7	Yes	College
P6	Man	3	Yes	Bachelor
P7	Man	12	Yes	Bachelor
P8	Woman	30	Yes	Bachelor
P9	Man	13	Yes	PhD
P10	Woman	20	Yes	Bachelor
P11	Woman	20	Yes	Master
P12	Woman	20	Yes	Bachelor
P13	Woman	7	Yes	Master
P14	Woman	20	Yes	Master
P15	Woman	15	Yes	Bachelor
P16	Woman	10	Yes	Bachelor
P17	Woman	5	No	Bachelor

3.2 Study Execution: Data Collection

Before interviewing the participants, we conducted four pilot interviews with a professor experienced in the OSS industry to validate the script and confirm whether the interview would fit in a 40- to 60-minute time slot. The pilot participants answered all the interview questions and provided us with feedback about the flow of the script. We also analyzed the questions and answers to ensure

that they provided data that would answer our research questions. In addition, we conducted four pilot studies with Ph.D. students who had mentoring experience in industry or OSS environments.

The participant interviews were semi-structured and used a central question: *"Can you walk me through your career journey and tell me the story from when you started until now?"* We asked further questions to follow up on unanticipated information and get additional details [49] about transition between roles (when mentioned), motivation for contributing to OSS, difficulties faced, and participants' reflections on their careers. Researchers followed the interview script,[1] while using an informal (*ad-hoc*) conversation format, with initial questions used to establish rapport.

All interviews lasted about 60 minutes. With participant consent, we recorded all interviews. The first author of this paper transcribed the interviews for the analysis.

3.3 Study Execution: Data Analysis

We qualitatively analyzed the transcripts by inductively applying open coding, whereby we identified the pathway that each participant followed, including the main roles and activities they reported. We built post-formed codes as the analysis progressed and associated them to respective parts of the transcribed interview text, as we aimed to code the roles and activities according to the participants' discourse, and not according to a preconceived set of roles.

While we consider "role" as a term that includes activities performed by individuals according to the organizational processes and following expected behaviors [45], it is possible for individuals to switch roles within a project or when moving to a different project. Individuals can also occupy multiple roles—sometimes in different projects—in parallel (see Section 4.2)

We coded each interview transcript and created a graph for the career story as explained by the participant. We then compared this graph with the career story of the following participants, using the constant comparison technique. When we found the same meaning for a concept that had been coded differently for more than one excerpt, we discussed until we found the appropriate concept that represented all the coded excerpts. The outcome was a set of higher-level categories created according to their properties, cataloged in a code-book [2]. We then created an integrated graph including all pathways, while annotating each edge of the path with the participant number to facilitate discussing the categories. Two researchers met twice a week to work on the analysis. The researchers discussed the codes and categorization until they reached a consensus about the meaning of and relationships between codes. In addition, we held a weekly meeting with the whole group (including two more experienced researchers) to discuss and adjust codes and categories until we reached agreement.

In addition to the OSS activities and roles mentioned by interviewees, we classified the activities according to their relationship with OSS: Non-related to OSS, OSS Consumer, and OSS Contributor, as explained in Table 2.

3.4 Study Execution: Member Checking

After analyzing the pathways of all interviews, we performed member checking to evaluate the validity of participant's individual pathways. To do so, we contacted 9 participants who had agreed for a follow-up meeting. We contacted these participants via email which included an editable visual representation of their pathway and description of each role. Participants could give feedback by email, annotating the pathway visualization directly, or through a online meeting.

Five participants provided their feedback. P2, P9, P13, and P14 scheduled virtual meetings, whereas P4 gave their feedback over email. Each call lasted about 30 minutes. During the call, we

[1] https://figshare.com/s/abebcfc04eb93fc3eca0
[2] https://figshare.com/s/abebcfc04eb93fc3eca0

Table 2. Classification for each activity according to relation to OSS projects

Category name	Rationale
Not-Related to OSS	Any activity not involving OSS (General recruiting, coding, managing project or providing technical support to proprietary software)
OSS Consumer	Activities using OSS, passively or actively, but not giving any contribution to OSS product or community (Lurking, Coder who uses OSS pieces to produce proprietary software, Software Startup Founder, System Admin or Project Manager who uses OSS)
OSS Contributor	Activities that use but also produce software, data or support back to the product or the community (OSS Coder, OSS Researcher, OSS System Admin, OSS Project or Community Founder, OSS Advocate or Community Manager)

first described the participants' career pathway using the visualization and then solicited feedback or correction. Four participants (P2, P9, P4, P13) verified that the pathways we had generated reflected their career path. P14 felt that our "project manager" role did not encompass software architecting, which she felt was a prestigious activity that should be recognized. We addressed this feedback by specifically calling out this activity in the project manager role.

4 RESULTS

Figure 2 shows the overview of our results, summarizing the roles and transitions between roles in the career pathways reported by participants. We used an * to represent people who are now simultaneously working in more than one role. In the rest of the section we present the results grouped per research question.

4.1 The prevalence of community-centric roles (RQ1)

There are different ways to contribute to OSS and it can be in different forms, as aptly mentioned by P3: *"all of the team members there brought this insatiable curiosity about what the others around the table had to contribute [...] contribution can take three forms: time, talent, and treasure ... "*. However, currently there is a lack of research about OSS roles and career opportunities unrelated to source code. Since successful OSS projects rely not only on programmers, but also on a community of actors [2, 40], it is imperative to consider different types of roles and activities—the focus of RQ1.

Figure 2 shows that the OSS roles reported by our participants are diverse and are played during different stages of contributors' careers. In addition to the common project-centric—mostly code-related—roles (*e.g.*, coder, system admin and project manager), our analysis identified the emergence of a set of community-centric—including non-code-related—roles and activities (*e.g.*, advocates and mentors). Although some of these roles are common in the software industry, they were not common (and many are still not formally recognized) in OSS until recently. In this section, we will present these roles, as reported by our participants, starting with the community-centric ones, and depict their importance to the current landscape of OSS.

4.1.1 Community-Centric Roles. This category includes roles related to: (a) the creation of the community (a.k.a OSS Community Founders) and (b) the management of the community (a.k.a OSS Community Managers). These roles, although known and important for the projects' sustainability and community evolution, are often not formally recognized in OSS communities. We present a summary of the roles we identified in Figure 3 and discuss them as follows.

OSS Community Founder

Creating a new product or a new project is the real beginning of OSS communities and this role can bring relevant experience to a career in OSS. This role is related to entrepreneurship, but in the OSS landscape, which can benefit both the founder and the community around the product. P4 for

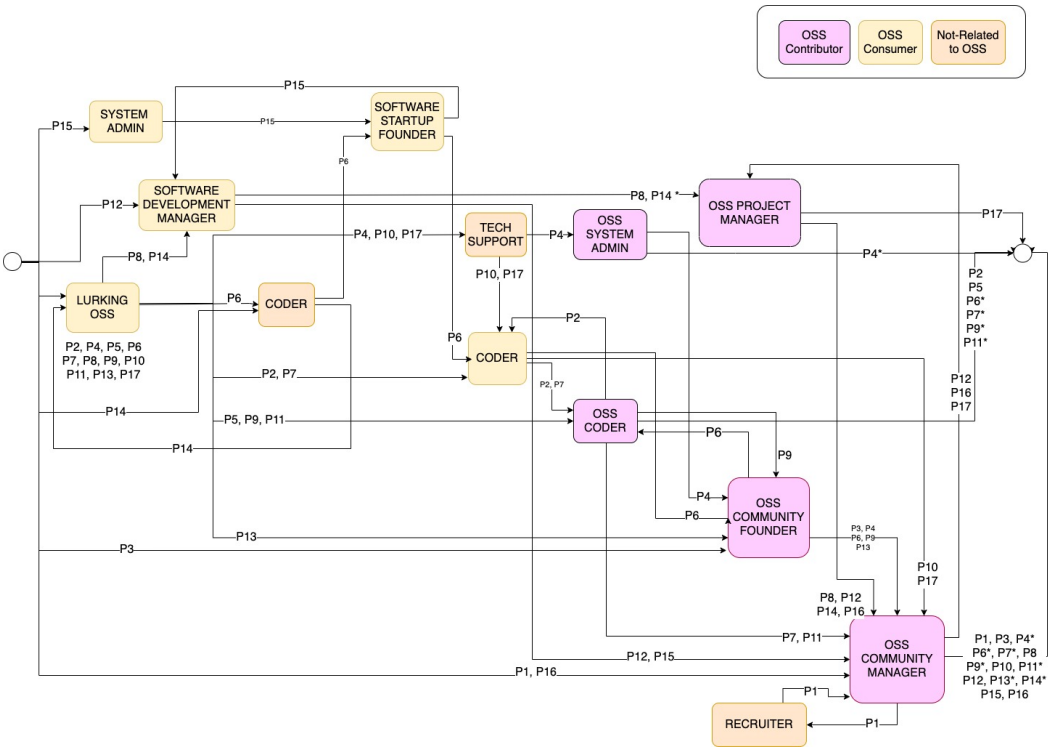

Fig. 2. Overview of career pathways reported by interviewees. The roles are classified according to its relation to OSS projects: OSS contributor (pink), OSS Consumer (yellow) and Not-Related to OSS (orange). The labels presented in the arrows, represent the participants who took that path (those decorated with '*' parallel roles).

example, mentioned that *"the bulk of my experience from working with open source communities comes from launching open source community."* We also noticed that "OSS entrepreneurship" became important for developers aspiring for professional development—according to P6 it was *"a way of advancing my own career."* He mentioned that it was important to add this OSS experience to his portfolio. Some interviewees mentioned that founding a community served to showcase their careers (P3, P9). P3 created an OSS content management system and was responsible for providing support services for their solution and for other similar OSS solutions. P9 took advantage of the knowledge acquired during his PhD to create an OSS startup that provides quality assessment services for OSS projects.

OSS Community Manager

Simply launching a community or project is not enough to create a thriving community. Most community founders, when they created their projects, did not have the suitable experience to manage the "business" and nurture it enough for it to become a an active and evolving entity [35]. In this sense, community managers play an important role in sustaining and helping a community mature. The "Community Manager" category groups different roles focused on a number of activities that are required to sustain and build a community and includes: OSS Advocate, OSS Mentor, OSS Treasurer, OSS Strategist, OSS Writer, and OSS License Manager.

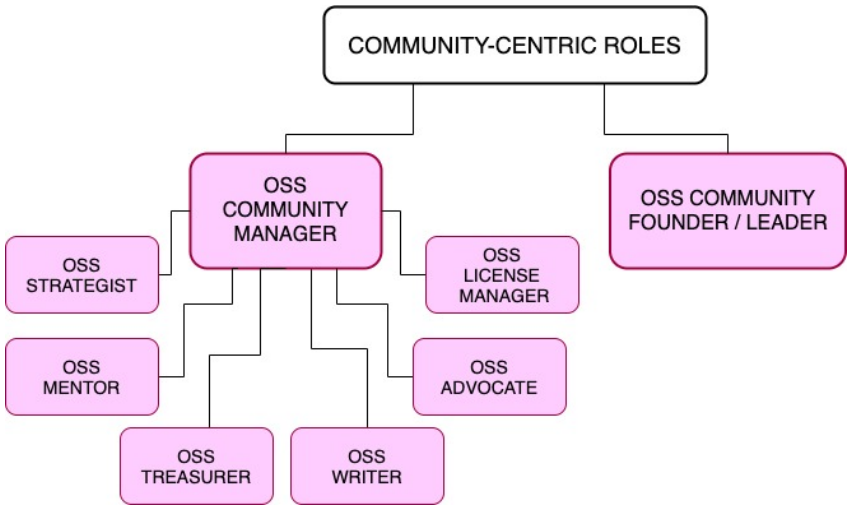

Fig. 3. The Community-Centric Roles: OSS Community Founder and OSS Community Manager (and the roles that are part of it)

OSS Advocate: Following the idea of *"why would we not do this together because we are better together"* (P1), this role embraces evangelism, developing plans for *"bringing new people in"* (P10), increasing contributions to the OSS project, and making the community inclusive, welcoming and safe. It involves activities about *"large mass engagement" (P16)*, how to *"connect and inspire people"*(P1 and P13), find and recognize talents, and build a trustful and friendly connection between the industry and the community. Activities in this role also include encouraging the community to learn the coding skills currently required by the market. One of the practical ways to operationalize advocacy is through events that bring people together. Since we interviewed people who were giving talks at an annual OSS conference, most of them mentioned this as a way to contribute to OSS. However, besides giving talks, some interviewees mentioned that they organized events by selecting *topics (projects, changes, histories, industry evolution), creating meetups to institute some open source flares in there" (P1)*, planning learning groups *"because we're better together" (P1)*, and running podcasts when there is a new software release to *"get people on the show to talk about their specialties" (P15)*.

OSS Mentor: In OSS, like any project, newcomers need to be trained in different aspects of the project [8], such as becoming acquainted to the project architecture, implementation and feature details, development guidelines, and organizational rules [14]. Although an important activity [20], it is often not recognized as a formal role in OSS, and those who perform the activities related to it are not formally trained, recognized, or assigned to play that role [8]. The perception of our interviewees matched that of prior evidence. Participants mentioned that mentoring is related to activities of *"teaching how to forage" (P10)* and it is usually a *"one-on-one engagement" (P10)* activity to understand why the person wants to contribute, present the tasks available, and give task recommendations based on newcomer's motivations.

OSS Strategist: "When a project starts to grow (or evolves) [it] requires governance" (P7), it is critical to deeply understand and control the quality of involved processes. This role is usually related to innersourcing initiatives, which is an emerging topic with growing interest from commercial projects that aim to replicate the success of popular OSS projects internally [54]. The OSS strategist

is responsible for fostering the adoption (of OSS technology) or for improving its processes (to match that of OSS) and improve transparency in organizations or communities, making *"strategic decisions for business" (P8)* and *"moving the company in OSS direction" (P12)*. This role is new in the software industry, and adds to the OSS landscape by fostering the involvement of software industry in OSS, and defining conditions for releasing (in house) projects under OSS licenses. This is an important role in the current age of OSS in which companies are central stakeholders.

OSS Treasurer: Treasuring is an important financial activity performed by someone who leads strategical budgetary decisions for the OSS project or foundation, and who creates a *"sustainable model" (P3)* for the community to support the projects. "*Treasure is literally money, literally money…It's not about technical direction at all, it's literally about making sure that this ecosystem can continue forward*" (P3). The increasing involvement of companies in OSS projects and the involvement of paid developers in the development process makes this role important to communities. Bringing and managing donations and income cannot be done in an *ad-hoc* manner, so having someone trained and focused on the activities is necessary.

OSS Writer: Well-written documentation is important not only for the everyday working of the project, but it can help onboard newcomers, create inclusive communities, and represents a meaningful way to contribute back to OSS. Nearly 25% of OSS community are not able to read and write English very well and need careful, clear, and accessible language in documentation [66]. Writing contributions can be technical documentation about the OSS product, like a software installation guide. But, it can also encompass educational materials, books, presentations and publications about OSS. Although not related to coding activities, the outcome of this activity can be visible in project repositories—*"I made contributions to Ubuntu documentation, I got involved with the docs team"*(P10)— and in publications— *"I wrote my book, which was great. And I'm glad that happened"*(P8), " *I've written a few books, the most recent and the most complete one is a case study book, it has some theoretical stuff at the beginning. And at the end, it has practical advice about how you can do it in your company*" (P12). The increasing importance of this role is evident; Google started a program called Google Season of Docs.[3] This program focuses on enrolling people who are technical writers to OSS projects to improve project documentation. Unfortunately, this role is not well recognized and documentation writing activities are done by people playing other roles.

OSS License Manager: Licenses are an important type of documentation for both users and contributors [66], since they are the legal means used to regulate how a software can be copied, changed, or redistributed. The central importance of licenses, and the specific knowledge needed to make the right decision, justifies the existence of someone who is responsible for licensing, including overseeing the compatibility and compliance of software licenses. Currently, this role is under-recognized, and licenses are applied by the maintainers, which may lead to inconsistencies and incompatibility [56, 60, 61]. The importance of this role is clear based on how long P12 spent on this activity: *"[I] was on the board of the OSI for 10 years, helping to run the licensing question."* P14 started working for an OSS foundation because the board knew she *"could handle certain issues, licensing and compliance because of the [projectname]"*, an experience she had accrued before as a Community Founder.

4.1.2 Project-Centric Roles. In this category we grouped those roles that relate to the project deliverable (the product of the OSS community's labor). These are mostly related to creating the product and maintaining the project, and not the community around it. Three roles emerged that fit into this category: OSS Coders, OSS Project Managers, and OSS System Admins.

[3] https://developers.google.com/season-of-docs

OSS Coders: This role is well-known and is the driving force in OSS product development. Since the beginning of OSS, this role has been considered the core piece of the model. This role includes activities related to developing new code, maintaining existing code, and writing tests. Our data shows contributors involved with core activities who regularly contributed to project and those who were occasionally involved with the project (casual contributors). In the former case, these contributors made it their career and reached "success" as coders—becoming long term contributors or maintainers. In the latter case, the casual [43] contributors eventually offered new patches or suggested fixes or new features. Often times, these contributions were a side product of another activity (*"hey, here's some code that I wrote"* (P7)) or a reason to stay active in the community (e.g., *"producing certain support product"* (P9)).

OSS Project Managers: People in this role perform two broad types of activities. First, management-centric activities, such as *"responsible for [...] effectively project managing, [in] budget, on time and on schedule, [with] appropriate features and keeping the quality levels high" (P14)* with the goal on *"producing open source projects" (P17)*. Second, product-centric activities which include, managing releases and project deliverables as per requirements and schedule constraints and/or being responsible for the product architecture. The architecture piece is "*a technical and complex activity that represents a career advance, as the architect makes core and strategic decisions about the product and reports to the organization's CTO*" (P14). When involved with large technical changes, an architect can also be a negotiator, having to to guide, detect and harmonize issues between a community of project stakeholders and avoid negative influence to system development [55]. While project management is less explored in OSS literature, it was central to Ye and Kishida's model [64]. However, they associated this role with software maintainers. Our participants reflected a more nuanced concept of project/product management.

OSS System Admin: The system admin supports the base operational systems, selects, configures, connects, and finetunes the subsystems that are components of a robust and efficient larger part [52]. This role includes activities related to providing technical support and system administration for OSS tools. A system admin provides tech support for OSS tools, and *"helps customers, walking them through registration, system setup, and all those different types of tasks that they need, they were trying to get done" (P4)*. This role is currently played by code contributors, although this is role well-defined in many company-sponsored projects.

RQ1. What different roles can be played by successful and recognized contributors to OSS?
Roles in OSS projects, as reported by our participants, extend beyond the code-related, project-centric roles. Community-centric roles that include advocates, strategists, community managers, community founders, mentors, license managers, writers, and treasurers are key. While these roles do not produce code, they are important for the growth and sustenance of OSS, especially in the new OSS landscape.

4.2 Entry points and career movements (RQ2)

In addition to identifying the roles played by OSS participants, we analyzed how they evolved in their career pathway. The first point we analyzed was the entry point in OSS. Most of our interviewees (11 out of 17) explicitly mentioned that they started by lurking in OSS, either by tinkering with code, installing and running OSS tools, or familiarizing themselves with OSS by reading or participating in events. P14's lurking was non-traditional since *"OSS didn't even exist at that time"*; We consider her as lurking, when she transitioning from a coder to product manager

in a commercial project, as the project needed some of the Linux components that were being developed. After this lurking stage, different participants followed quite different pathways. The other five interviewees did not mention lurking as their first activity related to OSS. P1, P3, P11, and P16 mentioned that they started as OSS advocates or project founders. P15 started directly in a job where he was a tech support admin using OSS.

After joining OSS, the interviewees reported their career evolution, which was fluid with no fixed paths. Participants changed their pathways in different ways, moving from coding to non-coding-related roles, across companies and projects, and performing parallel OSS and non-OSS activities. To start understanding the movements, we first analyzed the interviewees' careers from the perspective of the relationship of their roles with OSS (Table 2). This showed that most of the participants (15 out of 17) moved from roles not directly related to OSS contribution (lurking or consumers of OSS) to becoming OSS contributors. Three others started their careers directly as OSS contributors. These different movements are summarized in Figure 4.

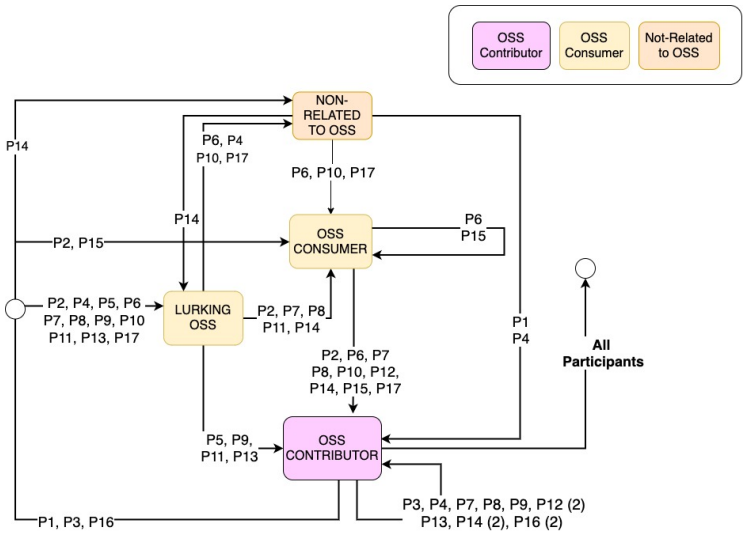

Fig. 4. Career Movements according to the relationship to OSS. The labels presented in the arrows, represent the participants who took that path.

We analyzed the career movements by labeling activities as coding (those dealing with the source code) and non-coding (when not related to the source code). Doing so revealed that some careers moved from coding to non-coding, while others remained exclusively in coding or non-coding activities (see Figure 6). Figure 5 shows some examples of individual pathways. We explore further these movements next.

4.2.1 From Coding to Non-Coding. This career path includes those participants who started in project-centric roles and then later moved to community-centric roles. This was the case of 7 out of 17 interviewees. Examples of this movement are the career pathways of P4, P6, P9, and P14 who created their own projects after observing the needs and the opportunities for the product. P9's path stands out. P9 studied OSS communities during his PhD and occasionally contributed to some projects. However, instead of becoming an active developer, he founded an OSS community that became a well-known company in OSS. This experience led him not only to work in his company, but also to become a board member in another community. Another interesting case is P6's, who

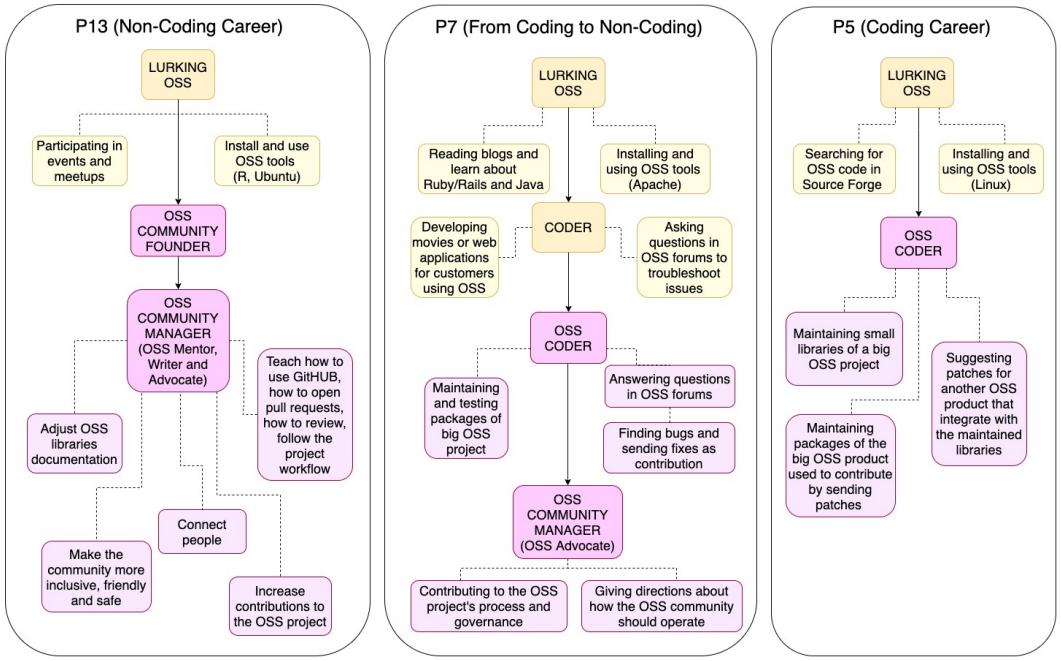

Fig. 5. Example of individual pathways for of three interviewees representing their fluid movements. P13 who had always been a non-coder contributor; P7 who moved from a coder to a non-coder role; and P5 who followed a coding career. The Inflection Points represent the events that motivated the role changes.

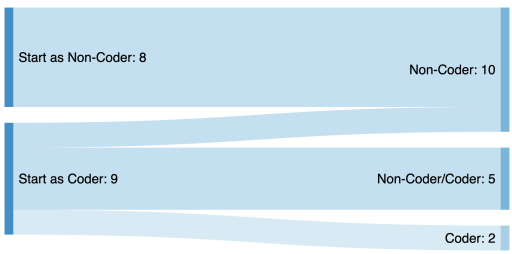

Fig. 6. Career Movements by activity type: Coders vs. Non-coders.

started by lurking: *"downloading some code from somewhere and compiling it, but mostly just like a hobby, like Middle School teenager type of thing"*. Meanwhile, P6 worked as a coder for a software company and then created a software startup based on OSS infrastructure and products to build solutions for customers (OSS consumer). P6 then quit the startup and went to work for big software companies as a developer, but still consumed OSS products for his work. He then created a new OSS product and started to contribute to it in parallel to his work as a coder in industry.

In other cases, a career movement occurred after the participant worked on code-related activities in OSS projects for a while and progressed to using their expertise to support the communities. P7, P10, and P17, are examples of such movement. P7, for example, started his journey lurking, then became a peripheral contributor and persisted until becoming a maintainer of an OSS project, which was his main occupation. The expertise he acquired through multiple code contributions

helped him not only continue as an active coder, but also to keep *"contributing more than code, contributing documentation, contributing processes, starting to form the governance of the project"*.

P11 followed a different path. After using OSS for many years, where she *"could never figure out how to actually contribute to open source"* but a friend of her invited her to work at a nonprofit community to help newcomers to OSS. She realized that *"documentation, community stuff is still really important, getting people to be able to install your software, whatever, is critical."* So, she engaged in writing documentation and organizing OSS events to support community building.

4.2.2 Non-coding career. 8 out of 17 interviewees built their careers on non-coding activities. People who traversed this path came from areas other than CS and were hired into OSS management positions either in large global tech companies or OSS foundations.

P1, P13, P15, and P16 did not have previous background in technology, yet had successful careers in OSS. P1 is a good example for someone successful who came from a different background. She mentioned that this was possible because she does not *"think you need to be a coder to understand community needs and norms"*. She started by eventually organizing some meetups and working as a general recruiter in a job not related to OSS. She did not like the overall job, but *"loved the advocacy pieces that came with it"*. This motivated her to change her career into OSS advocacy, innersourcing and community building. Although she was not trained in CS, she ended up learning the technical side so she could *"at least understand the development environment enough to suggest a change and to advocate for it appropriately"*.

On the other end, even with a very technical background, P4 never performed coding-related activities in his OSS career. He started providing Linux support to customers/users. His motivation was from joining a *"marketing program to work with their Linux user groups"*, where he realized that *"OSS is going to change the world"*. Later on P4 founded and still leads an OSS community, doing community building, and advocacy work.

P12 and P15, have different stories, they started in closed source software industry as program managers, where they became aware of OSS concepts and decided to move to OSS. P12 *"created a completely transparent project to the whole company"* to communicate how they were producing a new software. Then while at dinner with a friend a recruiter overheard her complaints about the pushback from engineers to OSS at the proprietary software company, and invited her to work as the project manager for another company, where she was *"the person who was assigned to open source and to move the whole company in that [Open Source] direction"*.

4.2.3 Coding career:

A couple of interviewees (P5 and P2) followed the onion model and performed source code-related activities during their entire career.

P5's story follows the classic "joining script" [57]. He started lurking by using code from projects in Source Forge for his pet projects. He *"kind of dabbled in Linux a bit, and then started running Linux kind of more heavily"* when he got his first laptop. After that, he started occasionally contributing to OSS with some patches until he became a regular code contributor, contributing *"more seriously to open source"* until he was hired by the project/company he was contributing to.

P2 was introduced to OSS in college, when she needed to *"change a small piece of the Linux kernel during the Operating System course"*). This experience opened her eyes to OSS and she applied for an internship as part of an Outreach program, which, according to her *"was one of the most motivating things that made me keep working with Open Source."* Through her career, she had different development roles in different companies moving from OSS contributor to OSS consumer and back again. When asked about her aspirations, she discarded the possibility of joining a community-centric career given her (technical) skills.

While coding-only career path is the most visible path (because of our current focus on source code activities), only two participants followed this traditional path. The rest followed markedly different pathways, especially those who came from non-tech backgrounds.

RQ2. How do roles evolve for such successful and recognized OSS contributors? Non-coding activities are an integral part of a flexible and fluid OSS career path, which encompasses multiple roles. Different participants from different backgrounds built their OSS careers through different roles and activities. The career evolution shows people with code-related expertise typically moving to non-coding activities and community-centric roles.

5 DISCUSSION

In this section, we review our results and briefly discuss our findings from different perspectives.

OSS has matured. Having left its teen years behind, OSS has "grown up"; OSS projects have become software powerhouses and an important driving force in today's software development environment. Even startups and commercial projects are increasingly relying on and contributing to OSS [4, 12]. With this maturity OSS has a new set of roles.

The subset of the roles and activities reported by our participants are similar to those in established software companies. These roles, in fact, are increasingly being sought out in the OSS projects in addition to more traditional project-centric and code-related roles.

The community-centric roles reported by our participants (including advocates, license managers, writers, strategists and mentors) are not usually well-recognized or easily identifiable from OSS project archives. Repository platforms (version control systems, issue trackers, project hosting sites) do not contain traces of these activities [15]. These activities, while crucial to the survival of the OSS community, are currently performed by *hidden figures* in the community. They (or their contributions) are not visible when navigating the project repository data. As a result they are not seen as the central figures of the project. This not only is demotivating for the hidden figures because of the lack of recognition, it also can keep away individuals who are not from CS background or interested in coding-related activities.

Parallel activities are not uncommon. According to the Open Source Survey of 2017, 70% of contributors are employed full- or part-time, and 85% of them are OSS coders, either at a regular or occasional level in their main job [66]. When analyzing our participants' pathways, we found that all of them were being paid to contribute to OSS at some point—either by big companies or by working for their own companies—to provide support and consulting on OSS. Even while being financially compensated, they mentioned that they performed parallel activities throughout their career, including volunteer (non-paid) activities in OSS projects. In our sample, P4, P6, P7, P9, P11, P13 and P14 shared stories about having a paid job in parallel with unpaid work in OSS.

Paid jobs were a starting point for an OSS career in some cases (P14, P4, P7). These participants mentioned working in industry while simultaneously working on OSS-related activities at the same time (e.g., casually contributing, founding communities). The participants reported that this led them to becoming closer to OSS projects, and then to getting OSS jobs based on the experience they had gained. P6, for example, founded a project that became his portfolio for joining OSS as his career. Additionally, some people continued being volunteers (e.g, committee membership, advocacy, mentorship) outside of their main occupation. For example, P13, who is a project manager in a big company, still performs community management activities for the OSS community she founded. In summary, volunteering to OSS helped some participants start careers that were in closed-source projects, but, even then many continued volunteering to OSS as a way to give back, or to maintain their reputation in the community.

Legitimate Peripheral Participation. The theory of Legitimate Peripheral Participation (LPP) has been widely adopted to describe how a person joins an online, open collaboration communities [22, 27, 28]. It mainly describes how participation, situated learning, and identity construction are interrelated, and can evolve. Participation is the key element in this theory, as it aids both situated learning and the construction of social identity within communities of practice. [34]. The theory of LPP applies to the majority of our participants' pathways too; 12 participants "lurked" while they were consumers of OSS products before learning enough about the product to start to participate and create their identities. Some participants did not mention lurking as their first steps (e.g., P12 and P15), but situated learning still occurred in practice while they worked and consumed OSS (P12 while managing software projects that included pieces of OSS and P15 while supporting tools based on OSS).

However, LPP fails to reflect some careers in OSS (e.g., P1, P3, P16), which did not follow the pattern of contributors starting in peripheral roles and building social identity. Instead, they started in central positions, by either creating communities or being brought into the community because of their expertise in other areas (e.g., Human Resources, Company Management). Such pathways are becoming more common in the current OSS landscape and need to be highlighted. Further, LPP also fails to match the pathways of participants who contribute to different projects in different roles. Our findings suggest that theory of LPP needs to be broadened to accommodate the different pathways in OSS that have emerged because of its changing landscape.

What about the Onion Model? The onion model–one of the best known joining model–is similar to LPP where contributors go through a set of roles ranging from being consumers of the project to becoming core contributors [41]. Current literature discussing this model has focused largely on source code activities within a single project. In our study, only two (P7 and P2) participants, followed the typical onion model progression, starting with tinkering, then performing peripheral contributions, until reaching the core level in an OSS project. Moreover, their careers were not limited to a single project and they performed different roles in different projects. In the rest of the cases—15 out of the 17 interviewees—the careers cannot be explained by the onion model as it is.

A detailed investigation of the career pathways reveals that only three of the roles we identified can be explained by the onion model (see Table 3). While most of the project-centric roles (2 out of 3) can be mapped to the onion model, only one community-centric role (community founder) fits into its roles descriptions. This is likely because the onion model focuses mainly on developers and project-centric activities. In the early 2000s, when the model was conceived, most OSS activities indeed revolved around code, and the model successfully captured the contribution process. But now that OSS has matured and involves large ecosystems, often encompassing multiple related projects, it needs a variety of roles to help sustain the contribution process. While project-centric roles exist and involve a core set of contributors, the emergence of the new community-centric roles adds to the onion model by creating a symbiosis between the project and its community. This is even more visible in OSS projects where industry is actively involved. We argue that OSS contribution models need to expand their view from a single project (onion) lens to the ecosystem (onion patch) lens, which in addition to including other projects (onions) also should encompass other community-centric activities (akin to maintaining the onion patch with soil amendments and water) that are critical to maintaining the health, growth, and sustenance of the entire ecosystem (onion patch).

A call for action: Projects and Researchers. The new direction that the OSS model is experiencing has meant a changed project landscape and more involvement from industry, and thus a new way of doing business and a new job market [23, 47, 53]. As part of this new OSS landscape, we identified a set of new roles and activities that are now formally or informally needed for the

Table 3. The mapping between the roles presented in this paper and the onion model

Roles	Relationship	Onion Model Roles
Community-Centric		
Community Founder/Leader	*encompasses*	Project leader
OSS Advocate		None
OSS Mentor		None
OSS Strategist		None
OSS Treasurer		None
OSS Writer		None
OSS License Manager		None
Project-Centric		
OSS Coder	*encompasses*	Bug fixer, peripheral developer, active developer, core member
OSS Project Manager	*encompasses*	Project leader
OSS System Admin	*encompasses*	None

growth and sustainability of OSS projects. As a result OSS can be more open than before—people with different backgrounds are now key to helping communities recruit and engage people, deal with technical writing, seek funding, dealing with external requests, and so on.

While bigger OSS projects backed or supported by big corporations or foundations have already filled many of these roles, smaller projects are still lagging and must adapt to this new landscape. A first step towards this is to find ways to recognize and reward these *hidden roles* and *hidden figures*. One way is to make explicit the existence of such roles and recognizing them as formal ways to contribute to the project. Formally recognizing the people working on these "less technical" roles and the types of activities they do can help attract people from diverse background to OSS projects.

In general, by having characterized the roles and pathways in OSS domain, CSCW community can benefit from our research. Multiple roles presented here are not related to the technical (coding) side of the projects, and may potentially exist in other domains. Therefore, future research can focus on understanding the existence of these hidden roles and pathways, or the impact that they may have in other open collaboration communities such as, Wikipedia, crowdsourcing initiatives (e.g., citizen science), as well as identifying commonalities and differences among roles in different types of communities. This may help to build more generalized models and theories about roles and pathways in open collaboration communities.

In summary, as presented in Section 2, most existing work focuses on the technical side of roles, pathways and success, both in Open Source [17, 30, 31, 41], and in Wikipedia [6, 7]). Such focus on the technical side of the communities, leaves behind those contributors who work "behind the scenes" to drive and sustain the community. It is therefore necessary to broaden our focus and start analyzing Open Collaboration Communities as larger endeavors, considering every contribution and role as worthy of investigation. Specifically for OSS, we can observe some efforts to understand how developers make use of Twitter [13, 50, 58] and News Aggregators [3, 29] can be found in the literature. This could be a time to re-target the efforts and investigate OSS contributors like any other professionals. An informal analysis of LinkedIn showed all interviewees have a a LinkedIn profile and report their experiences in OSS as part of their background. Thus, we see an opportunity to find new ways to mine information about project contributors through different channels, starting from professional networks (like LinkedIn), and exploring other platforms.

6 THREATS TO VALIDITY

We discuss validity and reliability of our results from the perspective proposed by Merriam [38] as our analysis is qualitative.

Construct validity in qualitative research is related to the precise definition of constructs. To control this limitation, we compared and contrasted our construct definitions with the literature, with the interviewees' LinkedIn profiles and OSS renowned gray literature like Open Source Initiative (OSI) and Free Software Foundation (FSF) webites. During the coding phase, we used the constant comparison technique [25] whereby each interpretation and finding is compared with existing findings as it emerges from the data analysis to increase the construct validity in our findings.

Internal validity is related to the credibility that researchers were able to capture the reality as close as possible. One can argue that our interviewees all had successful OSS careers (and walked successful pathways) as they were speakers at an important OSS conference. Thus, our findings do not identify those pathways that ended in failure or disengagement. Moreover, a majority of our interviewees identified as women, which is a different distribution from typical OSS gender demographics (e.g. 11.2% [46]). The roles uncovered in our study are not meant to be exhaustive and further research into different ecosystems, domains, and types of contributors will likely uncover other roles and pathways. During our recruitment process we reached out to OSCON invited speakers regardless of their gender. In our initial sample, out of the 11 interviewees, 6 identified as women. The sample reached via snowballing resulted in 6 other women (and no men). This can be because of two possible reasons. Firstly, it could be a self-selection bias given that some participants knew previous work from the researchers and identified themselves with the study. Secondly, when recruiting via snowballing, the women interviewees recommended more women contributors who ended up participating. While these limitations exist, we minimized their effects by selecting interviewees with different backgrounds and expertise. Additionally, we compared our findings with the literature on software engineering and OSS to sharpen construct definitions and increase internal validity.

Reliability refers to the extent that the results can be replicated. In short, it is difficult to replicate qualitative research since human behaviors, feelings, and perceptions change over time. Thus, in the reliability thread, Merriam [38] suggest checking the consistency of the results and inferences. According to Merriam [38], consistency refers to ensuring that the results consistently follow from the data and there is no inference that cannot be supported after the data analysis. To increase consistency, we performed data analysis in pairs, which was consistently revised by two experienced researchers.We had weekly meetings to discuss and adjust codes and categories until we reached an agreement. In the meetings, we also checked the consistency of our interpretations, continually discussing our results based on the previous literature. We also performed member checking with five participants and they confirmed our interpretation with minor changes.

Finally, the results presented in this paper are related to Open Source communities, thus, we do not expect that all our findings will be applicable to other contexts. However, to allow replication of our study, we carefully describe our research method steps.

Theoretical saturation. A potential limitation in qualitative studies is not reaching theoretical saturation. The quality, rather than the size, of the sample of participants is essential to increase our confidence on the results. In this study, we interviewed 17 participants with different perspectives and perceptions about the studied phenomenon. Our participants were diverse in terms of the number of years with OSS, experience in mentoring, roles, and highest academic qualifications. Further, these participants represent 15 different OSS projects that differ in size, domain, and ownership (company, community, and foundation). They went through many projects during their careers, changing roles within and across companies, as well as playing multiple roles in parallel. Moreover, as mentioned previously, the number of interviewed participants was adequate to uncover and understand the core categories in any all-defined cultural domain or study of lived experience [11]. While we cannot claim saturation, our population has helped us uncover

a consistent and comprehensive account of the nature of OSS contributions, which included uncovering several novel roles and pathways that.

7 CONCLUSION

In this paper we investigated the different roles performed by of successful and recognized OSS contributors during their career in OSS communities. We conducted 17 semi-structured interviews with recognized OSS contributors and qualitatively analyzed data to understand their career paths. We found that the roles that are available in OSS projects go beyond those related to the source code. We identified several activities that are performed as part of community-centric roles, which focus on creating and managing communities to make projects more welcoming, inclusive, and sustainable (RQ1). The stakeholders that play these roles are usually "hidden figures" who are not visible when analyzing the data from projects' repositories or their coding platform websites.

These community-centric activities are part of a flexible and fluid career path, that encompass multiple roles (RQ2). OSS projects include individuals with non-technical background who built their career in OSS without moving to roles that require programming. In addition, most of our interviewees (15 out of 17) evolved in their career and, at some point, played community-centric roles, even when they had technical background and expertise in code-related skills.

As future steps, we plan in-depth analysis of community-centric roles, to understand what projects currently do to foster them, and what are the needs for these roles to flourish. In addition, as a long term plan, we will work on proposing ways to identify and measure the evolution of stakeholders performing non-technical roles in OSS. Our goal with this research is to increase diversity in OSS, by helping currently hidden roles to be recognized and part of the core of OSS projects. This awareness will help attract people with different backgrounds and expertise to OSS by showing them the multitude trajectories to success.

ACKNOWLEDGEMENTS

We thank all the interviewees for their time and the insights. This work is partially supported by the National Science Foundation under Grant Numbers 1815486, 1815503, 1900903, and 1901031.

REFERENCES

[1] Ezgi Akar, Sona Mardikyan, and Tevfik Dalgic. 2019. User roles in online communities and their moderating effect on online community usage intention: An integrated approach. *International Journal of Human–Computer Interaction* 35, 6 (2019), 495–509.

[2] Stephanos Androutsellis-Theotokis, Diomidis Spinellis, Maria Kechagia, Georgios Gousios, et al. 2011. Open source software: A survey from 10,000 feet. *Foundations and Trends® in Technology, Information and Operations Management* 4, 3–4 (2011), 187–347.

[3] Maurício Aniche, Christoph Treude, Igor Steinmacher, Igor Wiese, Gustavo Pinto, Margaret-Anne Storey, and Marco Aurélio Gerosa. 2018. How Modern News Aggregators Help Development Communities Shape and Share Knowledge. In *Proceedings of the 40th International Conference on Software Engineering (ICSE '18)*. Association for Computing Machinery, New York, NY, USA, 499–510. https://doi.org/10.1145/3180155.3180180

[4] Gary Anthes. 2016. Open source software no longer optional. *Commun. ACM* 59, 8 (2016), 15–17.

[5] Ofer Arazy, Johannes Daxenberger, Hila Lifshitz-Assaf, Oded Nov, and Iryna Gurevych. 2016. Turbulent stability of emergent roles: The dualistic nature of self-organizing knowledge coproduction. *Information Systems Research* 27, 4 (2016), 792–812.

[6] Ofer Arazy, Hila Liifshitz-Assaf, Oded Nov, Johannes Daxenberger, Martina Balestra, and Coye Cheshire. 2017. On the How and Why of Emergent Role Behaviors in Wikipedia. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. ACM, 2039–2051.

[7] Ofer Arazy, Felipe Ortega, Oded Nov, Lisa Yeo, and Adam Balila. 2015. Functional roles and career paths in Wikipedia. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. ACM, 1092–1105.

[8] Sogol Balali, Igor Steinmacher, Umayal Annamalai, Anita Sarma, and Marco Aurelio Gerosa. 2018. Newcomers' barriers... is that all? an analysis of mentors' and newcomers' barriers in OSS projects. *Computer Supported Cooperative*

Work (CSCW) 27, 3-6 (2018), 679–714.

[9] Flore Barcellini, Françoise Détienne, and Jean-Marie Burkhardt. 2014. A situated approach of roles and participation in Open Source Software Communities. *Human–Computer Interaction* 29, 3 (2014), 205–255.

[10] Yochai Benkler and Helen Nissenbaum. 2006. Commons-based peer production and virtue. *Journal of political philosophy* 14, 4 (2006), 394–419.

[11] H Russell Bernard. 2017. *Research methods in anthropology: Qualitative and quantitative approaches*. Rowman & Littlefield.

[12] BlackDuck. 2016. The Tenth Annual Future of Open Source Survey. Available at: https://www.blackducksoftware.com/2016-future-of-open-source. Accessed: 2016-11-08.

[13] Gargi Bougie, Jamie Starke, Margaret-Anne Storey, and Daniel M. German. 2011. Towards Understanding Twitter Use in Software Engineering: Preliminary Findings, Ongoing Challenges and Future Questions. In *Proceedings of the 2nd International Workshop on Web 2.0 for Software Engineering*. ACM, New York, NY, USA, 31–36.

[14] Gerardo Canfora, Massimiliano Di Penta, Rocco Oliveto, and Sebastiano Panichella. 2012. Who is going to mentor newcomers in open source projects?. In *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*. ACM, 44.

[15] Kevin Carillo, Sid Huff, and Brenda Chawner. 2017. What makes a good contributor? Understanding contributor behavior within large Free/Open Source Software projects–A socialization perspective. *The Journal of Strategic Information Systems* 26, 4 (2017), 322–359.

[16] Can Cheng, Bing Li, Zeng-Yang Li, Yu-Qi Zhao, and Feng-Ling Liao. 2017. Developer role evolution in open source software ecosystem: An explanatory study on GNOME. *Journal of computer science and technology* 32, 2 (2017), 396–414.

[17] Jinghui Cheng and Jin LC Guo. 2019. Activity-based analysis of open source software contributors: roles and dynamics. In *2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*. IEEE, 11–18.

[18] Enrico Di Bella, Alberto Sillitti, and Giancarlo Succi. 2013. A multivariate classification of open source developers. *Information Sciences* 221 (2013), 72–83.

[19] Nicolas Ducheneaut. 2005. Socialization in an open source software community: A socio-technical analysis. *Computer Supported Cooperative Work (CSCW)* 14, 4 (2005), 323–368.

[20] Fabian Fagerholm, Alejandro S Guinea, Jürgen Münch, and Jay Borenstein. 2014. 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, 55.

[21] Joseph Feller, Patrick Finnegan, Björn Lundell, and Olof Nilsson. 2008. The servicitization of peer production: Reflections on the open source software experience. In *Information technology in the service economy: Challenges and possibilities for the 21st century*. Springer, 353–355.

[22] Casey Fiesler, Shannon Morrison, R Benjamin Shapiro, and Amy S Bruckman. 2017. Growing their own: Legitimate peripheral participation for computational learning in an online fandom community. In *Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing*. ACM, 1375–1386.

[23] Brian Fitzgerald. 2006. The transformation of open source software. *MIS quarterly* 30, 3 (2006), 587–598.

[24] Andrea Forte and Cliff Lampe. 2013. Defining, understanding, and supporting open collaboration: Lessons from the literature. *American Behavioral Scientist* 57, 5 (2013), 535–547.

[25] Barney G Glaser and Anselm L Strauss. 2017. *Discovery of grounded theory: Strategies for qualitative research*. Routledge.

[26] Wei Guo, Qing Zheng, Weijin An, and Wei Peng. 2017. User roles and contributions during the new product development process in collaborative innovation communities. *Applied ergonomics* 63 (2017), 106–114.

[27] Aaron Halfaker, Os Keyes, and Dario Taraborelli. 2013. Making peripheral participation legitimate: reader engagement experiments in wikipedia. In *Proceedings of the 2013 conference on Computer supported cooperative work*. ACM, 849–860.

[28] Karen Handley, Andrew Sturdy, Robin Fincham, and Timothy Clark. 2006. Within and beyond communities of practice: Making sense of learning through participation, identity and practice. *Journal of management studies* 43, 3 (2006), 641–653.

[29] Carolin D. Hardin and Matthew Berland. 2016. Learning to Program Using Online Forums: A Comparison of Links Posted on Reddit and Stack Overflow (Abstract Only). In *Proceedings of the 47th Technical Symposium on Computing Science Education*. ACM, New York, NY, USA, 723–723.

[30] Chris Jensen and Walt Scacchi. 2007. Role migration and advancement processes in OSSD projects: A comparative case study. In *Proceedings of the 29th international conference on Software Engineering*. IEEE Computer Society, 364–374.

[31] Corey Jergensen, Anita Sarma, and Patrick Wagstrom. 2011. 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*. ACM, 70–80.

[32] Giordano Koch, Katja Hutter, Peter Decarli, Dennis Hilgers, and Johann Füller. 2013. Identifying participants' roles in open government platforms and its impact on community growth. In *2013 46th Hawaii International Conference on System Sciences*. IEEE, 1900–1910.
[33] Yubo Kou, Colin Gray, Austin Toombs, and Robin Adams. 2018. Knowledge production and social roles in an online community of emerging occupation: A study of user experience practitioners on Reddit. In *Proceedings of the 51st Hawaii International Conference on System Sciences*. 2068–2077.
[34] Jean Lave, Etienne Wenger, et al. 1991. *Situated learning: Legitimate peripheral participation*. Cambridge university press.
[35] Zhuoxuan Li, Warren Seering, Tiffany Tao, and Shengnan Cao. 2019. Understanding Community Behaviors in For-Profit Open Source Hardware Projects. In *Proceedings of the Design Society: International Conference on Engineering Design*. Cambridge University Press, 2397–2406.
[36] Yuyang Liang and Joshua Introne. 2019. Social Roles, Interactions and Community Sustainability in Social Q&A Sites: A Resource-based Perspective. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*. 2802–2811.
[37] Jun Liu and Sudha Ram. 2011. Who does what: Collaboration patterns in the wikipedia and their impact on article quality. *ACM Transactions on Management Information Systems (TMIS)* 2, 2 (2011), 11.
[38] Sharan B Merriam and Elizabeth J Tisdell. 2015. *Qualitative research: A guide to design and implementation*. John Wiley & Sons.
[39] Audris Mockus, Roy T Fielding, and James D Herbsleb. 2002. Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 11, 3 (2002), 309–346.
[40] Dawn Nafus. 2012. 'Patches don't have gender': What is not open in open source software. *New Media & Society* 14, 4 (2012), 669–683.
[41] Kumiyo Nakakoji, Yasuhiro Yamamoto, Yoshiyuki Nishinaka, Kouichi Kishida, and Yunwen Ye. 2002. Evolution patterns of open-source software systems and communities. In *Proceedings of the international workshop on Principles of software evolution*. ACM, 76–85.
[42] Shaul Oreg and Oded Nov. 2008. Exploring motivations for contributing to open source initiatives: The roles of contribution context and personal values. *Computers in human behavior* 24, 5 (2008), 2055–2073.
[43] Gustavo Pinto, Igor Steinmacher, and Marco Aurélio Gerosa. 2016. More common than you think: An in-depth study of casual contributors. In *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, Vol. 1. IEEE, 112–123.
[44] Alessandro Piscopo and Elena Simperl. 2018. Who Models the World?: Collaborative Ontology Creation and User Roles in Wikidata. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 141.
[45] Wolfgang Reinhardt, Benedikt Schmidt, Peter Sloep, and Hendrik Drachsler. 2011. Knowledge worker roles and actions—results of two empirical studies. *Knowledge and Process Management* 18, 3 (2011), 150–174.
[46] Gregorio Robles, Laura Arjona Reina, Jesús M González-Barahona, and Santiago Dueñas Domínguez. 2016. Women in Free/Libre/Open Source Software: The Situation in the 2010s. In *IFIP International Conference on Open Source Systems*. Springer, 163–173.
[47] Gregorio Robles, Igor Steinmacher, Paul Adams, and Christoph Treude. 2019. Twenty Years of Open Source Software: From Skepticism to Mainstream. *IEEE Software* 36, 6 (2019), 12–15.
[48] Walt Scacchi. 2007. Free/open source software development: recent research results and emerging opportunities. In *The 6th Joint Meeting on European software engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering: companion papers*. ACM, 459–468.
[49] Carolyn B. Seaman. 1999. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on software engineering* 25, 4 (1999), 557–572.
[50] Leif Singer, Fernando Figueira Filho, and Margaret-Anne Storey. 2014. Software Engineering at the Speed of Light: How Developers Stay Current Using Twitter. In *Proceedings of the 36th International Conference on Software Engineering*. ACM, New York, NY, USA, 211–221.
[51] Ruvar Spauwen and Slinger Jansen. 2013. Towards the roles and motives of open source software developers. In *5th International Workshop Software Ecosystems (IWSECO)*. Citeseer, 62–73.
[52] Diomidis Spinellis. 2006. Open source and professional advancement. *IEEE Software* 23, 5 (2006), 70–71.
[53] Igor Steinmacher, Gregorio Robles, Brian Fitzgerald, and Anthony Wasserman. 2017. Free and open source software development: the end of the teenage years. *Journal of Internet Services and Applications* 8, 1 (05 Dec 2017), 17. https://doi.org/10.1186/s13174-017-0069-9
[54] Klaas-Jan Stol and Brian Fitzgerald. 2014. Inner source–adopting open source development practices in organizations: a tutorial. *IEEE Software* 32, 4 (2014), 60–67.
[55] Damian A Tamburri, Rick Kazman, and Hamed Fahimi. 2016. The architect's role in community shepherding. *IEEE Software* 33, 6 (2016), 70–79.

[56] Christopher Vendome, Mario Linares-Vasquez, Gabriele Bavota, Massimiliano Di Penta, Daniel M German, and Denys Poshyvanyk. 2015. When and why developers adopt and change software licenses. In *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 31–40.
[57] Georg Von Krogh, Sebastian Spaeth, and Karim R Lakhani. 2003. Community, joining, and specialization in open source software innovation: a case study. *Research policy* 32, 7 (2003), 1217–1241.
[58] Xiaofeng Wang, Ilona Kuzmickaja, Klaas-Jan Stol, Pekka Abrahamsson, and Brian Fitzgerald. 2014. Microblogging in Open Source Software Development: The Case of Drupal and Twitter. *IEEE Software* 31, 4 (2014), 72–80.
[59] Howard T Welser, Eric Gleave, Danyel Fisher, and Marc Smith. 2007. Visualizing the signatures of social roles in online discussion groups. *Journal of social structure* 8, 2 (2007), 1–32.
[60] Yuhao Wu, Yuki Manabe, Tetsuya Kanda, Daniel M German, and Katsuro Inoue. 2015. A method to detect license inconsistencies in large-scale open source projects. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*. IEEE, 324–333.
[61] Yuhao Wu, Yuki Manabe, Tetsuya Kanda, Daniel M German, and Katsuro Inoue. 2017. Analysis of license inconsistency in large collections of open source projects. *Empirical Software Engineering* 22, 3 (2017), 1194–1222.
[62] Jin Xu, Yongqin Gao, Scott Christley, and Gregory Madey. 2005. A topological analysis of the open souce software development community. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. IEEE, 198a–198a.
[63] Diyi Yang, Aaron Halfaker, Robert Kraut, and Eduard Hovy. 2016. Who did what: Editor role identification in Wikipedia. In *Tenth International AAAI Conference on Web and Social Media*. 446–455.
[64] Yunwen Ye and Kouichi Kishida. 2003. Toward an understanding of the motivation Open Source Software developers. In *Proceedings of the 25th international conference on software engineering*. IEEE, 419–429.
[65] Liguo Yu and Srini Ramaswamy. 2007. Mining cvs repositories to understand open-source project developer roles. In *Fourth International Workshop on Mining Software Repositories (MSR'07: ICSE Workshops 2007)*. IEEE, 8–8.
[66] Frances Zlotnick. 2017. GitHub Open Source Survey 2017. http://opensourcesurvey.org/2017/. https://doi.org/10.5281/zenodo.806811

Received January 2020; revised June 2020; accepted July 2020