A New Look at Occupational Learning and Socialization in Contemporary Careers: The Case of Coding Bootcamps

Abstract

As the traditional employment relationship has deteriorated in the U.S. and in much of the world, the nature of careers has been changing. Workers today need to continually navigate an external labor market, construct their careers out of a series of jobs or short-term gigs, and take responsibility for training and retraining themselves to remain employable over time. Despite the burgeoning literature examining workers' efforts at navigating the new economy, we know relatively little about how workers attempt to reskill themselves in the course of their careers. This paper utilizes a unique setting—coding bootcamps—to examine how workers attempt to enter a skilled occupation without traditional organizations serving as the backdrop for their efforts. I argue that bootcamps resembled learning collectives where learning from peers and near-peers figured more prominently than expert instruction. Under conditions of minimal expert instruction and obstacles to legitimate peripheral participation, I show how aspiring software developers sought out an occupational community in virtual spaces, learned asynchronously from unknown others, developed their practice through mock-work among themselves and managed to get hired into junior developer roles.

Keywords: Employment, occupations, careers, learning, socialization

A New Look at Occupational Learning and Socialization in Contemporary Careers

Since the 1980s, with the deterioration of the traditional employment relationship in the United States and much of the Western world, firms have declined in importance as organizing structures for people's careers (Cappelli, 1999a; Osterman & Shulman, 2011; Weil, 2014; Davis, 2016a). Job tenures have shortened, advancement paths have moved outside the firm, and nonstandard work is becoming pervasive not just for low-skilled workers, but also for high-skilled workers (Lichtenstein, 2002; Hollister, 2011; Kalleberg, 2011; Weil, 2014; Katz & Krueger, 2019). As a result of changes in the nature of employment, the nature of careers is changing. Contemporary careers are more interorganizational (Arthur & Rousseau, 1996; Arthur, 2008; Smith & Neuwirth, 2008; Bidwell & Briscoe, 2010) as well as more precarious (Kalleberg, 2009, 2011). Workers today—whether they are in standard or nonstandard work arrangements—have to continuously navigate an external labor market, construct their careers out of a series of jobs or short term gigs, and take responsibility for training and retraining themselves to remain employable over time (Smith, 2001, 2010; Barley & Kunda, 2004; O'Mahony & Bechky, 2006).

Research on how workers navigate the external labor market has looked at the perils of temping (McAllister, 1998; Smith & Neuwirth, 2008), or doing gigwork (Rosenblat, 2018), the practices of temp agencies (Fernandez-Mateo, 2007; Bidwell & Fernandez-Mateo, 2008, 2010), the human and social capital building efforts of independent contractors to stay relevant in the job market (Barley & Kunda, 2004; O'Mahony & Bechky, 2006), and the difficulties of unemployment among experienced professionals (Newman, 1988; Smith, 2001; Ehrenreich, 2005; Lane, 2011), among other topics. Despite the growing number of studies examining the dynamics and challenges of contemporary careers, few scholars have looked at how workers attempt to reskill themselves to enter growing occupations in the hopes of improving their career prospects. Even though scholars note that changing occupations throughout one's career is becoming increasingly common (Tolbert, 1996; Hall, 2004; Baruch & Vardi, 2016), such transitions, especially career transitions into skilled occupations that take place outside of traditional institutional contexts are poorly understood.

Our understanding of how people become members of occupations is still largely based on studies conducted in traditional organizational contexts that facilitate occupational socialization: vocational and professional schools, and firms that coordinate the efforts of multiple occupational groups (e.g. Becker, Geer, Hughes, & Strauss, 1961; Van Maanen, 1973; Schleef, 2006; Michel, 2011; Anteby, 2013; Ranganathan, 2017). In such settings, occupational incumbents structure and facilitate the socialization of newcomers (Van Maanen & Schein, 1979; Saks & Gruman 2012). They share their expertise, initiate newcomers to the code of conduct of their occupation, and offer feedback to newcomers on their progression from novice to expert. What happens when aspiring occupational entrants pursue less institutionalized processes of occupational learning and socialization made necessary by nonstandard employment relationships (e.g. O'Mahony & Bechky, 2006), technologically mediated and distributed work (Schwartz, 2018), or the emergence of new and fleeting organizational arrangements such as incubators, bootcamps, hacker and makerspaces are ill understood. Such settings do not facilitate the kind of access between novices and occupational incumbents that is identified in the literature as being necessary for occupational entry.

Relying on a 17-month ethnography of two coding bootcamps in Silicon Valley, this paper investigates how aspiring career switchers enter a skilled occupation without following established paths of occupational entry. I show how bootcamps resembled *learning collectives* where self-learning and learning from peers and near-peers[1] figured more prominently than expert instruction. Under conditions of minimal formal instruction or mentorship from proximate experts, I show how aspiring software developers sought out an occupational community in virtual spaces, acquired expertise asynchronously from unknown others, developed their practice through 'mock work' among themselves, and managed to get hired into junior software developer roles. This study responds to recent calls to study the changing nature of work and careers (Barley & Kunda, 2001; Barley, Bechky & Miliken, 2017), specifically by focusing on "transitional periods" and "interstitial spaces" between jobs, which are becoming a prominent feature of work and employment (Bechky & Anteby, 2016: 502; see also Halpin & Smith, 2017).

BECOMING A MEMBER OF AN OCCUPATION

Several streams of research shed light on the process of becoming a member of an occupation or profession[2]. These are the literatures on socialization, situated learning and professional identity construction. While socialization and situated learning literatures examine how novices learn shared practices, behaviors and attitudes, the literature on professional identity construction focuses more specifically on the changing self-conceptions of new entrants to an occupation.

Occupational Socialization

Occupational socialization is the process by which novices learn the task domain of an occupation and develop patterns of thought and action that are shared by the occupational

[1] Near-peers are defined as other learners who are a few stages ahead of the novices in question and who can therefore teach them, even though they are not yet experts themselves.

[2] In the remainder of the paper I will use the term occupation to refer to both occupations and professions because professions are a subcategory of occupations that have more standardized training requirements, strong professional associations, and strong member identification with the occupation (Tolbert, 1996), and have to a large extent succeeded in convincing audiences that they command an esoteric body of knowledge and hence should be granted autonomy in their practice (Anteby et al., 2016).

community (Anteby, Chan, & Benigno, 2016). Incumbents act as ‘socialization agents’ (Van Maanen, 1973; Saks & Gruman 2012, Ranganathan, 2017). They transfer knowledge, model appropriate behaviors and attitudes, and provide feedback on newcomers’ progression (Becker, Geer, Hughes, & Strauss, 1961; Pratt, Rockmann, & Kaufmann, 2006). Novices are not merely passive recipients either. They observe incumbents and actively engage in information seeking behaviors to understand the demands of their new role (Morrison, 1993; Saks & Ashforth, 1997). Through the transmission of knowledge and worldviews between incumbents and newcomers, new generations of occupational entrants reproduce the expertise that is the jurisdiction of their occupation (Van Maanen, 1973; Jordan, 1989; Riemer, 1977).

Studies of entry into skilled occupations depict socialization as a long and elaborate process (Moore, 1976), a well-established pathway to membership guarded by occupational incumbents who control access to practice (Wilensky, 1964; Freidson, 1993; Abbott, 1988, 1991). Abbot (1988: 84) notes “It is common to create rigid entry standards, coupling extensive education with several levels of examination prior to formal entry into the profession. This is part of a structure of control… It protects recruitment, controls professional numbers (and consequently professional rewards), and guarantees a minimum standard of professional ability.” The occupational entry process for skilled occupations and professions thus involves a period of formal training, often coupled with a period of internship, apprenticeship or residency where learning continuous on the job and becomes more relevant to actual practice (Moore, 1976; Trice 1993; Bailey & Barley, 2011). Importantly, studies of occupational socialization all describe situations where novices have access to proximate incumbents to learn from and emulate. Whether we are talking about the socialization of a doctor in the classroom and at the hospital (Becker et al., 1961), the policeman at the academy and on patrol (Van Maanen, 1973, 1978), or the gradual indoctrination of a neophyte midwife (Jordan, 1989), the assumption is that novices and incumbents are co-located. It is assumed that only through such proximate teaching and mentoring can occupational incumbents shuffle newcomers through the cognitive and behavioral changes necessary to 'become' a member of an occupation (Anteby, Chan, & Benigno, 2016).

Situated Learning

Situated learning refers to the process by which a newcomer learns the expertise of an occupational group through participating in their shared practice (Brown & Duguid, 1991). The central idea is that learning and doing cannot be separated. Newcomers to an occupation learn through participation as they work among incumbents. They initially partake in peripheral tasks and slowly acquire the necessary skills and knowledge that lead to mastery (Brown & Duguid, 2001; Jordan, 1989). Lave & Wenger (1991) proposed the term 'legitimate peripheral participation' to describe this relationship between novices and expert practice. Inspired by the model of craft apprenticeship, situated learning theory deemphasizes formal or didactic teaching, and instead highlights the importance of learning through imitation and through examples or 'occasions of use' as practitioners communicate in the course of shared practice (Jordan, 1989; Orr, 1996).

Many occupations are either learned completely through situated learning (e.g. the crafts, or arts) or the entry process involves some period of situated learning (e.g. law school internships, medical residency, or the field training period of police socialization). As in theories of socialization, access to co-located incumbents is key to situated learning theory. Communities of practice, or occupational communities, are described as being composed of novices and incumbents who are co-located and who share both tacit and highly esoteric knowledge over shared practice. In fact, Lave & Wenger (1991) talk about access as an inherent aspect of legitimate peripheral participation. They argue, "to become a full member of a community of practice requires access to a wide range of ongoing activity, old-timers, and other members of the community; and to information, resources, and opportunities for participation" (Lave and Wenger, 1991: 101). Neither socialization nor situated learning literatures have explored how novices learn and become socialized when access to incumbents and legitimate peripheral participation opportunities are constrained (Beane, 2018).

Occupational Identity Construction

In their treatment of entry to an occupation, both the socialization and situated learning literatures talk about identification (e.g. Becker & Carper, 1956), changing self-conceptions (e.g. Hughes 1958, 1971), or the development of 'identities of mastery' (Lave & Wenger, 1991) as an important aspect of becoming a member of an occupation. However, changes in identity are not the main focus of these studies (Pratt et al., 2006). It is the more recent management literature on identity work that focuses explicitly on how individuals construct their occupational identities in the course of socialization.

Identity work scholars argue that individuals entering a new role actively craft new professional identities by trying on or playing with provisional selves (Ibarra, 1999; Ibarra & Petriglieri, 2010). Identity construction involves crafting and modifying one's view of herself, as well as gaining social validation for that view from one's interaction partners (Ashforth, 2001). In both parts of this process (i.e. identity crafting and seeking validation) an occupational community with proximate incumbents is key. Occupational incumbents serve as mentors and role models for possible selves (Ibarra, 1999; Pratt et al. 2006). They offer feedback on how well the newcomer is progressing from novice to expert, and provide social validation for novices' newly forming occupational identities. While scholars note that peers and near-peers are also important in this process, studies predominantly emphasize the role played by more senior members of the occupational community. In Ibarra's (1999) study of professionals undergoing role transitions, or in Pratt and colleagues' (2006) study of the socialization of medical residents, for example, novices judge the desirability or appropriateness of possible future selves in reference to senior members of the occupation. As novices act out these new identities in the making, senior members of the occupational community offer them feedback on their role performance, which serves to validate and solidify novices' sense of self.

Taken together, the literatures on occupational socialization, situated learning and occupational identity construction all point to the importance of access to proximate occupational incumbents and role models in the learning and socialization process. What happens when newcomers pursue noncodified, under-institutionalized paths to occupational entry that may preclude or constrain such access is poorly understood. While scholars studying contemporary careers note that more and more workers today undergo noncodified transitions and discontinuities in their careers, and cross occupational boundaries in their quest for greater career fulfillment, few scholars have empirically looked at what noncodified, under-institutionalized career transitions entail in practice (Ibarra & Obodaru, 2016; Petriglieri, Petriglieri, & Wood, 2017). Ibarra's (2003) influential study of mid-life career switchers comes closest to investigating under-institutionalized processes of socialization and identity development. However, even in this study, we see that the majority of Ibarra's informants do not transition into occupations with esoteric expertise and skill requirements. Such career transitions face "real gatekeepers, such as professional groups and organizational recruiters" (Demetry, 2017: 201).

RESEARCH SETTING AND METHODS

Research Setting

Coding bootcamps offer a unique setting for studying under-institutionalized career transitions into a skilled occupation. Bootcamps emerged in 2011, during the second technology boom, as an alternative pathway for reskilling for careers in software development. Bootcamps are characterized as highly focused vocational training programs that resemble immersion-based language programs (Scott, Holzman, Ris and Biag, 2017). They offer programs in full-stack web or mobile application development[3] to aspiring occupational entrants, most of whom have no background in programming. The idea and promise behind the bootcamp model of occupational entry is to transform complete novices into hirable junior occupational entrants through a short period of intense learning and socialization.

Bootcamps represent a stark contrast to computer science degree programs which are considered by many as the traditional or desired pathway into careers in software development. A computer science degree program is based on a four year long curriculum covering basic engineering training with math and science requirements, computer science theory, programming, as well as other elective subjects. Like many traditional paths for entry into skilled occupations and professions, CS degree programs involve the transmission of abstract knowledge from experts to novices in a classroom setting. CS degree programs also have built-in components allowing legitimate peripheral participation in the form of summer internships. These internships are not a requirement for graduation, however they are one the primary ways that graduates find jobs. Moreover, the credential received through this training—a bachelor of

[3] Full-stack web application development refers to being able to develop both the server-side (backend) of an application, which is the part that holds the data and the logic of the application, as well as the client-side (fronted), which is what the user sees and interacts with through the browser. Frontend development has an aesthetic component, whereas backend development involves architecting how the web application will behave as the user interacts with it.

science—is recognized by the occupational community and employers. Even though this credential is not a requirement to be accepted into the occupation, many employers list it as a preferred qualification.

In contrast, coding bootcamps train students in a very limited and specialized body of knowledge pertaining to web application development. Computer science theory is covered only superficially. The focus of bootcamp training is to develop job relevant, practical skills. My preliminary observations and interviews also quickly revealed that bootcamps offered minimal formal instruction to the learners passing through their doors. While most bootcamps were 12 to 14 week long programs, up to half of this time did not involve instruction and was devoted to projects and job search related activities. Moreover, most of the learning at bootcamp happened not between experts and novices, but between novices and near-peers who were only a few stages ahead of the novices in question.

Besides receiving minimal formal instruction from experts, aspiring software developers following the bootcamp path of occupational entry faced obstacles to legitimate peripheral participation in the occupational community. They could not apply to internships, because internships required candidates to be enrolled in formal degree programs. Finally, in terms of credentials, bootcamp graduates did not receive any form of degree or certificate recognized by employers and the occupational community. As a result, when they entered the job market, they had neither the credentials nor the experience (and hence recommendations from previous employers) that are used in external labor markets to support one's claim to being a legitimate candidate for a job. Despite these obstacles arising from following a new and under-institutionalized mode of occupational entry, I found that the majority of my informants at my two field sites accomplished their desired career transition. Six months after finishing bootcamp, 64% of the career switchers I followed had accomplished their career transition. Within nine months, this number rose to 77%. Despite the various access challenges they faced, how did aspiring developers acquire the necessary expertise and become socialized into their desired occupation such that they were able to get hired as software developers and engineers?

Data Collection

Little was known about coding bootcamps when I embarked on this project. I employed inductive, ethnographic research methods, which are particularly valuable for building theory about novel phenomena (Glaser & Strauss, 1967; Strauss & Corbin, 1990). My goal was to develop a grounded understanding of the bootcamp path of occupational entry from the eyes of career switchers (Spradley, 1979). I was in the field for a total of 17 months, from August 2016 to December 2017. I conducted a total of 80 semi-structured interviews with bootcamp graduates, founders and staff in the San Francisco Bay Area. I spent 8 months of my fieldwork immersed in the lives of three cohorts of students attending two comparable bootcamps. I opted to have two comparable field sites so as to increase the generalizability of my findings. While most of my interview subjects were recruited from my two field sites, I also conducted ethnographic interviews with graduates of other bootcamps, which allowed me to expand my sample.

Nonparticipant Observation. I initially conducted observations at over 20 learn-to-code meetups organized by bootcamps and nonprofit learn-to-code organizations in the San Francisco Bay Area. Through connections I made at these meetups, I first secured access to one of the most well known bootcamps in the Bay Area—which I call CodeCamp. I spent on average three days a week at the San Francisco campus of CodeCamp for five months (December 2016 to April

2017) following two cohorts of students. To enhance the validity of my results, I then secured access to a second field site, another prominent bootcamp which I call DevHouse, and followed one cohort of students there for three months (June 2017 to August 2017).

Both CodeCamp and DevHouse were full-time, 12 to 13 week long programs considered ‘immersive’ learning experiences. They were very similar in terms of how long they had been in operation, the duration of the program, and the content of the curricula. Both bootcamps had several ‘campuses’ across the United States. The main difference between DevHouse and CodeCamp was their selectivity. Both bootcamps had an admissions process in place that assessed the fit and aptitude of applicants. DevHouse’s admissions process was known in the industry and recognized among aspirants as being rigorous and selective, whereas CodeCamp’s selectivity was considered to be low.

At both CodeCamp and DevHouse, formal hours were from 9:00am to 6:00pm, when everyone was expected to be on the premises. Many students also stayed into the late hours of the evening finishing up the day’s work, preparing for the next day, or attending an evening meetup at the bootcamp. Some preferred to come in on weekends, especially if group work was required. Once the curriculum was over, students were encouraged to continue coming to the bootcamp and use the ‘alumni space’ for their job search and interview preparation. This coworking space function that bootcamps offered their alumni was very useful for my research purposes, as it gave me access to my informants during their job search journey, as well as meeting rooms where I could conduct interviews. I tried to limit my observations to 5-hour episodes a day so as to be able to accurately record my observations. When days ran longer due to evening meetups or events, I recorded my observations the next day.

At both bootcamps I introduced myself as a researcher and PhD student. I had full access to the student experience at both bootcamps. I could come and go as I pleased and attend any events I wanted. I opted to be a nonparticipant observer because I realized early on that trying to learn to code myself would consume all of my attention and not allow for detailed observations and note taking. As an observer, I sat between rows of learners and observed aspirants working alone, working in pairs, or working in teams. I also observed aspirants during the brief lectures they had each day. I joined them for lunch time activities, as well as evening meetups. I attended panels where alumni came to give talks to current students and advise them on their job search. I attended job search lectures offered by bootcamp staff. I observed aspirants practice technical interviewing with each other, or help improve each other's resumes.

I hung out in the alumni spaces at CodeCamp and DevHouse and engaged in countless informal conversations with job seekers. I attended several alumni events where job seekers broke into groups to commiserate over the job search process, share anecdotes as well as leads. I observed bootcamp graduates work on new projects together to improve their portfolios. Finally, I followed my informants to tech industry networking events, job fairs, and two hackathons.

Ethnographic Interviews. I conducted a total of 80 semi-structured interviews in the San Francisco Bay Area.[4] I conducted eleven semi-structured interviews with the founders and directors of different bootcamps and nonprofit learn-to-code organizations. These interviews served as background information, gave me an understanding of the industry dynamics, as well as the challenges facing bootcamps and their graduates in the job market. I then conducted 69 ethnographic interviews with aspiring software developers who completed bootcamps. I

[4] I conducted an additional 32 interviews in the Midwest to validate that similar processes were observed outside of Silicon Valley. While I do not use my data from the Midwest in the presentation of my findings in this dissertation, they informed my theorization.

recruited the majority of my interview participants (count=58) from my main field sites, CodeCamp and DevHouse. I interviewed 31 CodeCamp graduates and 27 DevHouse graduates roughly 6 months after graduation. My goal in these interviews was to develop an in-depth understanding of the career transition process of my informants. I asked them about their backgrounds, how they decided to switch careers, how they decided to pursue a bootcamp training, their experience at bootcamp, as well as their job search experience afterwards, and (if they were already employed) their adjustment to their first jobs. I expanded my sample with 11 interviews with graduates of five other bootcamps in Silicon Valley. I recruited these interviewees via LinkedIn or through the technology networking events I attended.[5]

Student Outcomes I gathered job placement data on two cohorts of students who graduated around February-March 2017 from my two field sites. This was an important set of data to gather in order to understand if I were observing a learning and socialization process that was actually successful in initiating newcomers into the occupation. I collected a comprehensive list of students who graduated in February and March 2017 from CodeCamp (count=28), and the closest cohort from DevHouse, who graduated in March 2017 (count=56)[6]. I found that six months after graduation, 73% of DevHouse graduates and 46% of CodeCamp graduates who sought employment in software development had accomplished their career transition. The job placement rate across the two bootcamp cohorts six months after graduation was 64%. By nine months after graduation, 86% of DevHouse graduates and 57% of CodeCamp graduates had found programming related employment. The job placement rate nine months after graduation

[5] These eleven interviews were conducted anywhere between 1 month to 2 years after the informant had finished bootcamp.

[6] I combined February and March graduates from CodeCamp, because otherwise my sample from CodeCamp would have been too small compared to DevHouse.

across the two bootcamps was 77%. In total, 17% of the 84 aspiring developers in my sample did not accomplish their desired career transition within a year after finishing bootcamp.

Data Analysis

I analyzed my data using ATLAS.ti software, following the principles of inductive analysis and grounded theory building (Glaser and Strauss, 1967; Strauss and Corbin, 1990). I initially open-coded my field notes and interview transcripts. This process revealed the recurring actions of my informants and recurring themes in my data. I then developed a set of higher order codes to organized my data. For example, first-order codes such as 'self-learning through documentations/tutorials/forums', 'Googling', and 'asking near-peers' were all aggregated under 'seeking expertise'. Similarly, 'dealing with being stuck' and 'getting code to work' were subsumed under 'relating to the object of practice'.

FINDINGS

The Reproduction of Occupational Expertise at Bootcamps

Bootcamps did not resemble formal educational institutions, which are characterized by extensive classroom teaching as experts transmit abstract knowledge to novices (Becker, et al., 1961; Schleef, 2006). They did not resemble—or necessarily lead to—internships or apprenticeships, where newcomers work side by side with incumbents and learn within an occupational community through legitimate peripheral participation (Lave & Wenger, 1991). The two bootcamps I studied housed few, if any, experts and offered minimal formal instruction to learners. Instead, learning happened through self-study coupled with working at the edge of one's capacity (Beane, 2018) among peers and near-peers in what I'm calling a *learning* *collective*. I will first talk about the role of peers and near-peers in these learning collectives, before turning to the role of the virtual occupational community in expertise reproduction.

The Role of Peers and Near-Peers: Bootcamp as Learning Collective

Format of training. The bootcamp curriculum was divided into two parts: an instructional curriculum (the first nine weeks), and time for projects and interview preparation (the remaining three to four weeks of bootcamp). During the first phase, students followed a curriculum that walked them through learning basic algorithms, how to store and retrieve data, how to set up a development environment, how to use several backend and frontend languages, frameworks and libraries to build full-stack web applications from scratch, and how to deploy these applications on the Internet.

Bootcamps utilized a flipped classroom approach whereby students were given tutorials to watch, and readings and exercises to complete on their own the night before. Classroom instruction time was spent answering questions, clarifying key concepts, and live coding part of an assignment that was particularly important for the work of that day. After the brief lecture, students spent the rest of the day at their computer terminals working on the day's coding challenges.

I found that bootcamps often utilized near-peers for teaching. These former students were aspiring developers themselves who, after bootcamp, opted not to go immediately into the labor market and instead took full-time or part-time jobs at the bootcamp. This allowed them to start earning income while continuing to improve their programming knowledge for a few more months before embarking on the job search. Hiring their own graduates for teaching and mentorship roles was a common practice in the bootcamp industry, partly to reduce costs. At DevHouse—which had significantly better student outcomes than CodeCamp, and some of the most successful student outcomes in the industry overall—there were no experts. All the instructors and teaching assistants were near-peers. At CodeCamp, there were two full-time instructors and two part-time mentors who had worked as professional programmers before. However, they were outnumbered by near-peers.

In the course of the nine weeks of instructional curriculum, lectures became more brief over time. Finally, during the project phase of bootcamp, there were no lectures and students worked in teams to devise project ideas and build their own web applications. For these projects, aspirants were expected to learn and utilize technologies beyond the bootcamp curriculum in order to make themselves more employable upon graduation. While they were learning these technologies, it was often the case that the teaching staff could not assist them because the technologies fell beyond their level of expertise. Instead, aspiring developers learned from the multitude of knowledge artifacts made available online by the vast occupational community of software developers.

My data suggest that throughout the course of bootcamp, aspirants developed most of their expertise through self-learning, learning with their peers, and learning from and learning with their near-peers, who were themselves trying to progress from aspirant to hirable junior developer. In interviews, my informants emphasized the collective and informal nature of learning at bootcamps, and the lack of intentional formal teaching:

> What a bootcamp offers you is the environment. You get to work with other people on problems. [At DevHouse] there were very short lectures, then we split into pairs and worked through problems on our own. You could ask questions to the teaching assistants, that's all. It was mostly you learning with your peers. All the resources are out there for free. (Nathan, Aspirant, DevHouse, Personal Interview)

> Everyone was helping each other out. It was a very inclusive community of people trying to learn together. (George, Aspirant, CoderSpace, Personal interview)

My data from other bootcamps indicate that the informal, collective nature of learning applied not just to CodeCamp and DevHouse, but more broadly to the bootcamp model of occupational learning and socialization. Laura, who had attended Digital Academy and who was working as a Software Engineer when I interviewed her explained,

> You're learning from other people who are learning themselves. There is not a lot of hands on instruction from people who know what they're doing. If you need help you can get help, but you're getting help from people who went through the course three months ago. It really helped solidify the idea that this isn't unobtainable knowledge. This is stuff you can figure out on you own. We'll give you the tools that you need to figure out how to do this on your own. And that is so empowering. (Laura, Aspirant, Digital Academy, Personal Interview)

What made the learning collective model of occupational socialization possible was the organization of expertise in the broader occupational community of software developers. Aspirants were able to engage with and learn from a broader, virtual occupational community that shared knowledge amongst each other, as well as with amateurs and enthusiasts.

The Role of the Broader Occupational Community

For aspiring developers trying to progress from neophyte to hirable occupational entrant, learning involved a constant search for expertise online. Every technical task they faced involved many small steps, each of which they had to figure out how to accomplish. Aspiring developers, therefore, spent a large portion of the day searching for information online, sifting through a multitude of knowledge artifacts created by other developers, selecting appropriate solutions to the problems they faced, and applying these solutions through trial and error. This self-learning process became especially salient when they ran into errors—which happened constantly—and when the instructional curriculum was over and aspirants started building their own projects and preparing for technical interviews.

The knowledge artifacts that aspirants consulted as they engaged in self-learning included documentation, tutorials, Q&A forums, blog posts, and code repositories, among others. Documentation refers to users' manuals for software tools, typically produced by the creators or maintainers of the tools, and modified and expanded through the contribution of a user community. Aspirants supplemented this knowledge source with tutorials and forums. Aspiring developers also heavily utilized StackOverflow, a question and answer forum that is the most popular online platform for accessing a crowdsourced body of knowledge about programming (Vasilescu, Filkov, & Serebrenik, 2013; Bosu, Corley, Heaton, Chatterji, Carver & Kraft, 2013). In most cases, aspiring developers could access the occupational expertise they needed by searching through existing question and answer threads on StackOverflow, instead of posting new questions themselves.

Aspiring developers constantly reached out to the expertise of this broader occupational community for their learning needs. Each time an aspiring developer faced a new learning need, she consulted multiple knowledge artifacts available online until she found a solution that worked. Through this process, aspirants cultivated a reliance on themselves as self-learners, and a reliance on the broader occupational community as the source of the knowledge they sought to amass. For example, as a short project, Adam built an application that shows a user at any point in time what restaurants are open for the next two hours within a certain radius. For this project, Adam needed to learn how to implement Google Maps API (application programming interface), which would integrate his application with Google Maps. He first watched a video tutorial created by a developer advocate from Google to understand the basics of using the API. As he was working, he started searching StackOverflow and other sources to find information on how to implement various features to the map such as dropping a pin on the location of the user. By the time he completed his Google Maps integration, he had accessed the expertise of multiple developers and learned from them virtually and asynchronously in order to accomplish his task at hand.

Another aspirant explained how he went about learning Python for a team project, and also how to use Raspberry Pi, which is an inexpensive, single board computer for teaching/learning programming. He said,

> I think the biggest part was learning what to look for. Learning how to ask and how to get an answer. Python generally translates pretty well with Ruby [*which was the first coding language he learned at bootcamp*]… I actually started late with Python because I was trying to get the Raspberry Pi working. So when I started learning Python, the other guys had already been grappling with it, so I asked them. They showed me what they learnt. I did a tutorial… This project was mostly researching. I didn't know Python or Raspberry Pi until this project. (Phil, Aspirant, CodeCamp, Field notes)

All of my informants described the project phase of bootcamp as a process of self-learning and learning as a team, using online resources:

> You're learning on your own and you are just looking up documentation a lot. You're looking up StackOverflow and researching yourself. (Oliver, Aspirant, DevHouse, Personal Interview)

Some of my informants referred to the Google search engine as their "friend" or "team member," to emphasize the importance of searching for information and self-learning in the course of building their team projects. For example, below is an excerpt from my field notes where a team of aspirants were talking about the game they built and referred to Google as “the fifth member of (their) team.”

> Matt: We wanted to try our hands at a game. It’s a retro game with Pixel art.
> Sharon: We had quite a few challenges. None of us had built a game before.
> Matt: Google in general was our friend. Probably the fifth member of the team. (CodeCamp, Field notes)

Learning new technologies in the course of everyday work and self-reliance in the process of learning were the most salient features of the occupational socialization process of aspiring developers at coding bootcamps. All of my informants interpreted their training as a process of “learning how to learn new technologies” through reaching out to the expertise of the broader occupational community. While some students expressed discomfort with the lack of teaching at bootcamps, many internalized self-learning as a core skill they needed to develop in their desired occupation.

> The biggest skill I learned [at bootcamp] is how to learn other coding languages. I’ve gotten really good at Googling problems. When you run into a problem, 99 times out of a 100 someone else has run into the same problem. So when you Google your error message, you find a StackOverflow entry or a tutorial... It’s amazing how many resources are available for free in tech. I am way less scared about striking out on my own [after bootcamp]. (Matt, Aspirant, CodeCamp, Fieldnotes)

Like Matt, aspirants viewed bootcamp as the beginning of their learning experience. While striking out on one’s own after bootcamp implied self-reliance, this was made possible by cultivating a reliance on the broader occupational community for just-in-time expertise. Aspiring developers recognized this body of fellow developers with whom they interacted virtually, and referred to “the developer community” in their conversations:

> It is an incredible community. There are about 25 million people in the world who consider themselves to be in software development one way or the other. And these people learned through each other. For example StackOverflow – it's all people helping each other figure things out. (Jim, Aspirant, CodeCamp, Personal interview)

Aspirants talked both about a developer community at large, as well as the smaller user communities formed around particular technologies. These community members were the ones responding to questions on forums like StackOverflow around their particular area(s) of specialization. For example, when talking about the community of developers around one coding language that they focused on, a near-peer mentor at Hacker School said: "It is a great community. There are people out there who'll respond to your StackOverflow comment in the middle of the night" (Field notes). My data thus indicate that the learning process at bootcamps was premised on the distributed organization of expertise in the occupational community, as well as the culture of expertise sharing among practitioners as well as enthusiasts.

Occupational Socialization and Identity Development

At both of my field sites, I observed aspiring occupational entrants form an intense relationship to the object of their practice—computer software. They constantly talked about the programs they were writing; formed and expressed preferences about which technologies (languages, text editors, frameworks, etc.) were superior to others; they obsessed about errors and shared stories of how they solved particular problems they faced. Over time I observed aspirants gain self-confidence as they started bending software to their will and built web applications that "others could see and interact with". As these accomplishments accumulated, I observed aspirants talk about themselves as becoming software developers. These changes correspond to the encounter stage of socialization when novices first come in contact with professional practice (Van Maanen, 1973). In the case of the skilled occupations and professions, encounter usually happens in the course of an internship, apprenticeship or residency, or a new graduate recruitment program. Such social arrangements allow novices to work alongside incumbents who not only show them 'the ropes', but also provide novices with feedback on their progression and validation for their newly forming professional identities as junior members of the occupation (Ashforth, 2001; Pratt et al., 2006). In fact, Wenger (1998) referred to these apprenticeship style arrangements for working and learning as "generational encounters" (p.99) between newcomers and old-timers.

Aspiring developers at bootcamps largely lacked access to such generational encounters because they were operating on the fringes of the occupational community. Despite these access challenges, I found that aspiring developers at bootcamps became socialized into their new occupational roles in the course of engaging in mock-work in the form of projects. These projects facilitated their socialization as aspirants built a relationship to their object of practice, which not only taught them the natural rhythm of software development work, but also over time provided validation for their new occupational identity as developers.

Developing a Relationship to the Object of Practice

At both CodeCamp and DevHouse, I found that aspirants adopted a rhythm of work that structured their relationship to the object of their practice—computer software. This rhythm, which repeated itself throughout the day, involved "dealing with being stuck" until you managed to "get your code to work". These were two separate states of work between which aspirants oscillated constantly.

Aspiring developers expressed being "stuck" repeatedly in the course of working. Being stuck described a cognitive and emotive state where the aspirant faced a problem she did not understand and could not figure out how to resolve with her first few attempts at finding a solution. One could be stuck trying to understand a new concept, trying to implement a new feature, decipher a team member's code, or fix errors in a program (i.e. debugging). When an aspirant was stuck trying to fix errors in a program, getting code to work involved careful scrutiny of the code, adding or deleting code until the error was identified and fixed. Sometimes errors were as simple as a semicolon where it didn't belong and yet debugging was still a time consuming task. If the aspirant was trying to solve a more complex problem in her code, getting code to work involved researching online and asking peers or near-peers for help before starting the trial and error process of applying various solutions to figure out what worked. Being stuck elicited frustration, which was sometimes expressed with the idiom, "banging one's head against the wall."

"Getting code to work" was the goal of everyday practice for novice developers. Whereas an expert developer might direct her practice towards improving the performance of code or making code more elegant (i.e. written in a more clever way, often consisting of fewer lines of code), novices focused exclusively on getting code to work. Sennett (2008) notes that as skills progress, work becomes more problem-attuned, trying to improve technique and quality, whereas when skill level is still primitive, the main goal is to get things to work. For aspirants trying to progress from amateur to hirable occupational entrant, this was the case.

When code ran, the moment was marked by a feeling of elation. Being able to manipulate software, and being able to see the outcome immediately on the computer screen was a cause for joy that was often expressed out loud. The below excerpt from my field notes shows the pride and elation that aspirants derived from getting code to work. I was observing several aspirants sitting together one day when Bill figured out what was wrong with his code and managed to get his program to work:

> Bill lifts his fist in the air and says, "I did it! I can go home now!" He holds up his computer, showing his team the rendering of a Google maps on his screen
> Aaron: Congrats man. What was it?
> Bill: I wasn't giving it a height, because I wasn't styling it or anything. So I gave it a height and it worked!
> Aaron: Oh yeah man, Google maps doesn't work if you don't give it a height and width.
> Bill [smiling, turns to me]: This is the reason to write code. This moment.
> Aaron: Yeah, that high is totally worth it.
> Bill: Today is the best day right now. I worked through all my errors too. So now it's all good. Until the next thing.
> (DevHouse, Field notes)

By "the next thing," Bill was referring to the next time he would get stuck, which he expected would be soon. The elation that he expressed ("Today is the best day right now"), or the "high" as Aaron put it, was a common sentiment among aspiring developers and accompanied moments of success when code worked. Another aspirant, when describing how he got his code to work on a particularly difficult project said:

> I'm really cautious [while making changes to my code] right now. I've already broken it a couple of times and it was so hard to fix it. You know how it is; the lows are really low and the highs are really high. (Dave, Aspirant, DevHouse, Field notes)

Cycling between being stuck (and feeling frustration) and getting code to work (and experiencing elation) structured aspiring developers' relationship to their object of practice—software—and the technical complexity it represented. This rhythm and the success moments it embodied also facilitated the development of their new occupational identities.

Stepping into a New Occupational Identity

Aspiring developers began to think of themselves as builders and developers as they experienced success moments in their work, ranging in complexity from writing simple programs that produced desired outputs to building a clone of a popular web application from scratch to building their own projects and deploying these applications on the Internet for others to see and interact with. In the course of their learning journey, aspirants' sense of self as builders and developers evolved with their technical ability. Scholars studying work and identity development processes have noted that learning about identity and learning about work go hand in hand (Hughes, 1958; Fine, 1996; Pratt et al., 2006). Especially when work involves building a specific skill, there is a gradual buildup of knowledge such that

> At the end, persons will 'discover' themselves to be considerably different than when they began. One sees this effect most clearly in the acquisition of complex skill or in the formation of a complete 'professional' perspective' (Van Maanen & Schein, 1979: 51).

The literature however, also points to the importance of an audience, because our identities are social and require validation from others (Ashforth, 2001), particularly knowledgeable others consisting of role models and occupational incumbents (Ibarra, 1999, 2003). Such role models and knowledgeable audiences were largely absent at CodeCamp and DevHouse. Audiences instead consisted mostly of peers and near-peers who were also outsiders to the professional community that aspirants were trying to break into. My data suggest that in the absence of proximate incumbents to provide feedback on their performance and validation for their newly forming identities, aspiring developers derived feedback and validation primarily from the object of their practice.

Aspiring developers formed an emotion-laden relationship to the object of their practice—software—which was visibly responsive to their actions. When code didn't work, red error messages appeared. When code worked, it produced a desired output. Seeing the fruits of their labor on the computer screen served as an undeniable manifestation of technical accomplishment, even in the absence of validation from an expert audience. Over time, as aspiring developers experienced more and more significant success moments in their work, they began to think of themselves as 'real developers' and not just amateurs.

For example, I was observing Roy and Alice from DevHouse pair programming one day when they achieved a new success moment that made them feel like software developers for the first time. Roy and Alice were in the early weeks of bootcamp and were slowly progressing from working on small coding challenges to actually building web applications similar to the ones they regularly used in their daily lives. In this case, they were trying to build a simple to-do list application where the user can add, delete and check off to-do items on her list. I was observing them struggle to figure out how to connect the 'backend' or server side of the application, which held the database of to-do items, with the 'frontend', which is the part that the user sees in the browser. After multiple trial and error attempts, when they finally managed to get their code to work, Alice shouted in joy:

> Alice: We connected our frontend and backend!
> Roy: Which had never happened before!
> Alice: I'm pretty sure we *just* became software developers. Like this moment was it! (DevHouse, Field Notes)

In this instance, and in many other such success moments I observed in the course of my fieldwork, aspirants derived validation for their newly forming occupational identities as developers from the object of their practice. Validation did not come from their teachers or mentors, but from software itself.

Different success moments could provide validation for aspirants' new occupational identities. Jim from CodeCamp described the time he started feeling like a developer:

> I started feeling like a developer when we started dealing with databases and routing, and learning about concepts like RESTful routing as a best practice and the whole HTTP handshake and how to chain together web pages and then start to do that programmatically so you can write little programs that will generate your whole website for you and build out all these different pages. That's when it really clicked and I thought, 'Oh wow, *I can do this!*' All those little pieces that I've been learning now fit together into something that I can say, 'Look, I built this. *This is something I can do now!*' (Jim, Aspirant, CodeCamp, Personal interview, emphasis added)

Being able to "build" things, to generate a whole website, validated Jim's sense of self as a developer.

The mock-work that aspiring developers completed at bootcamp in the form of several small-scale projects was particularly conducive to feeling like a 'builder' and 'developer' because aspirants built these applications from scratch. Whereas in a professional setting a developer might work on a large codebase, trying to maintain an existing software system and making only small modifications to it, the bottom-up project work that bootcamps asked from their students allowed aspirants to execute one or more application ideas of their own from the ground up. Building an application idea from the ground up allowed aspirants to feel like developers, albeit junior ones. Their projects were visible, tangible, interactive web applications that came alive on their computer screens. Sam, who loved playing video games, felt this pride when he was able to build a simple JavaScript game for the first time:

> I guess that the biggest thing [about the bootcamp experience] was being like ‘wow I can't believe I actually can do these things!' With coding, there are these small pieces that you learn how to do, and then there is also the larger idea that I can take these small pieces and build them into something big and amazing. I'm a PC gamer. Now I could actually build – not those huge millions of dollar budget games but the smaller games, indie games – I could actually code those now! (Sam, Aspirant, DevHouse, Personal Interview)

In each of these instances, aspiring developers felt that they finally had a tentative grasp on the expert practice they were trying to break into. Having technologies respond to them in the way they intended was a huge accomplishment *(“I could actually code those now!”)*. These technical feats that accumulated over time served as sources of validation for their new self-concept as developers. I call this validation that aspirants derived from engaging with the object of their practice *objectual validation.* After just a few months of learning and working on projects at bootcamp, aspirants reported identifying with the labels ‘web developer’ and ‘software developer’.

DISCUSSION

Traditionally two types of institutions in society train and socialize newcomers into occupations and professions: schools where students are formally trained through classroom instruction, and workplaces where newcomers learn the expertise and behavioral demands of their occupations on the job, surrounded by incumbents (Becker et al., 1961; Moore, 1976; Fine, 1985; Schleef, 2006). Coding bootcamps did not resemble these established institutions that facilitate occupational entry. What made bootcamps an interesting field site was specifically this unorthodoxy and their popularity despite it.

I found that bootcamps resembled learning collectives more so than schools. Aspirants learned through self-study, which was possible as a result of the distributed organization of expertise in a virtually connected occupational community. They learned from their peers in the course of mock-work, and they learned from their near-peers who were empowered in formal roles such as instructor, teaching assistant and mentor. Finally, the object-centered nature of their occupational practice enabled aspirants to derive validation for their new occupational identities as their skills improved.

The role of near-peers. Our understanding of occupational communities largely rests on a conceptualization of relations between experienced practitioners and newcomers (Lave & Wenger, 1991; Wenger, 1998). In fact, Lave and Wenger (1991: 29) propose that the concept of legitimate peripheral participation "provides a way to speak about the *relationship* between newcomers and old-timers". While the role of near-peers is recognized, it is not seen as central in the socialization of newcomers. At bootcamps, however, due to the high cost of hiring expert instructors, near-peers played a key role in scaling instruction. While learning from those who were trying to learn themselves could be frustrating when questions could go unanswered, most of the time it reinforced a sense that the expertise that aspirants were trying to master was "obtainable". Aspirants reported feeling empowered as they sought answers to problems within a their bootcamp community, and later being put in a position to mentor more junior students.

The Role of Occupational Community. Since expertise was limited within these learning collectives, bootcamps encouraged aspirants to learn how to learn on their own by reaching out to the expertise of a broader occupational community that was organized online and shared knowledge openly, not just among practitioners but also with amateurs. Software developers, more so than most occupational groups, share an ethos of open access to knowledge (Levy, 1994; Rheingold, 2000; O'Mahony & Ferraro, 2007). This ethos can be interpreted as the main reason for the abundance of tutorials, or the existence of discussion boards and knowledge sharing platforms such as StackOverFlow (Lakhani & von Hippel, 2003; Wang, Lo, & Jiang, 2013). Thus, although the processes of occupational expertise reproduction at bootcamps were under-institutionalized and unorthodox compared to our traditional understanding of occupational expertise reproduction, they drew on existing systems of communication and collaboration in a wider occupational community of software developers organized irrespective of organizational membership. Forming a relationship to this virtually organized occupational community, and learning how to learn from the multitude of knowledge artifacts that the community produced and shared was key to the socialization process of aspiring software developers.

*Developing a Relationship to the Object of Practice***.** Socialization into an occupational role involves, among other things, learning the system of relations in which a practitioner is embedded (Emirbayer, 1997; Anteby et al., 2016). Software developers working in professional settings are embedded in a system of relations with product managers, user experience designers, quality assurance engineers and other occupations with whom they must collaborate, as well as the users of their products whose feedback they incorporate into their work. The bootcamp experience was not conducive to socializing aspirants into the entirety of this system of relations. However, I found that the bootcamp experience socialized aspirants into two important relationships that lay at the heart of software development practice: The relationship of the practitioner to the occupational community from whom she will continue to learn throughout her career, which enabled them to attain expertise in the absence of proximate experts to learn from, and secondly, the relationship of the practitioner to the object of her practice—software. This second relationship was important for the socialization of aspiring developers because software development work is an object-centered practice. In the case of many occupations, the objects of practice are other people or institutions. For example, "the neophyte policeman learns how to deal with the objects of his occupation—'the traffic violator, the hippie, the drunk, the brass, and the criminal justice complex itself" (Van Maanen, 1973: 412). For a doctor, the objects of her practice could be patients, families of patients, as well as the institutional order of the hospital which the doctor must learn how to work with. However, certain occupations such as engineers, scientists, craftsmen and artists have inanimate things as their primary object of practice. As a result, relating to these objects by way of exploring, taming, and learning to give shape to them is key to both skill and identity development.

Aspirants at bootcamps were able to develop a relationship to their object of practice through the projects they did, which resembled real life work scenarios. In the absence of opportunities for legitimate peripheral participation, engaging in such mock-work facilitated aspirants' socialization as well as identity development. Hughes (1958: 122) notes that, "in any occupation, people perform a variety of tasks, some of them approaching more closely the ideal or symbolic work of the profession than others." The standalone projects that aspiring developers built most closely approximated the ideal or symbolic work of web developers, and facilitated aspirants' identification with the occupation. Aspirants derived what I call *objectual validation* from their object-oriented practice as they experienced various technical feats throughout their learning journey that culminated in this ideal type of work.

Most organizational research focuses on how people construct a sense of self in activities and relationships with others. The role of technologies or inanimate objects in this process is rarely considered. I propose the concept of objectual validation, which is inspired by Knorr-

Cetina’s (2001) concept of ‘objectual practice’, which she used to refer to nonroutine, object-centered knowledge work. Knorr-Cetina’s (2001) argument was that object-centered nonroutine knowledge work is filled with tension and emotion, and is generative as a result. I build on this insight and suggest that such object-centered, nonroutine knowledge work can generate validation for professional identities. The nonroutine feature of work is critical in this process, because routine work does not lend itself to the success moments I identified in my data. It is through these success moments and the pride they elicited that aspiring developers at bootcamps were able to derive validation for their newly forming professional identities. By putting forth the concept of objectual validation, I bring a central insight into organization studies from studies of science and technology—that people don't just use technologies but interact with them. People can develop emotion-laden relationships to objects that respond to their actions and that reveal new possibilities in the course of nonroutine practice (Pickering, 1993; Knorr-Cetina, 2001). As such, objects can provide validation for identities in the course of practice.

Finally, this paper aims to contribute to the fledgling literature on under-institutionalized learning and socialization processes in contemporary careers (e.g. Ibarra and Obodaru, 2016; Schwartz, 2018; Beane, 2018). Contemporary careers unfold in a world of work that is characterized by fewer formal organizing structures (e.g. Davis, 2016a, 2016b). New organizing practices coupled with new technologies are not only changing the nature of work (Barley, Bechky, & Miliken, 2017), they are also changing how people develop new skills, take on occupational roles and participate in the world of work. Crowdsourced workers or algorithmically managed freelancers, for example, need to ‘understand the lay of the land’ or ‘learn the ropes’ (VanMaanen & Schein, 1979) without the benefit of proximate supervisors or peers (Gray, Suri, Ali, & Kulkarni, 2016; Schwartz, 2018). In these contemporary work arrangements, novices need to take responsibility for their own learning and socialization, seek out an occupational community on their own, and pick up skills however they can to become employable and remain employable.

For example, in his study of creative freelancers producing digital goods for a video game company, Schwartz (2018) shows how freelancers who never met face to face formed a virtual occupational community where they shared expertise, helped each other progress from novice to expert, and collaborated on projects to smooth out an uncertain income stream caused by the precarity of their relationship to the client. Other studies suggest that even for people who pursue more traditional careers, changing technologies are altering the landscape of working and learning such that people need to be more self-directed in their learning, socialization and career advancement. Beane (2018), for example, in his study of medical residents trying to learning robotic surgery skills shows how changing technologies can hinder access to expert practice such that novices fail to learn necessary skills. The few successful learners, Beane (2018) finds, were those who engaged in 'shadow learning' practices; who developed skills outside of their residency program, rehearsed their skills watching videos on YouTube, and created opportunities for themselves where they could work at the edge of their capacity with little oversight. These examples suggest that self-directed and under-institutionalized processes of learning and socialization are becoming more common across a range of occupations and work arrangements, and our fields needs more such ethnographic studies in order to revise and expand our understanding of occupational learning and socialization in the age of digital technologies, distributed work, and fleeting employment arrangements.

Generalizability

The aspirants I studied were able to progress from amateur to acceptable occupational entrant outside of traditional paths of occupational entry as a result of several specific features of the object of their practice, and the way that expertise is organized in the occupational community. These features of the object of practice were that it could be inexpensively set up outside of an industrial context, and learned through trial and error. The features of the organization of expertise in the occupational community that made bootcamps viable were that expertise was distributed and openly shared among practitioners and amateurs, and could be communicated digitally. Bootcamps, as an assemblage of people, technologies and processes, were overlaid on this broader network of people, technologies, and processes for expertise sharing. As a result, even though the bootcamp path of occupational entry itself was under-institutionalized, it drew on existing processes of knowledge sharing and reproduction in the broader occupational community. Any expertise domain with similar characteristics of the object of practice, and similar organization of expertise in the broader occupational community, we can assume, could be learned in a bootcamp type of social arrangement.

Moreover, software development has historically prided itself in being a meritocratic occupation where skills are more important than credentials (Ensmenger, 2010). Even though computer science degrees are highly valued in many labor markets, and even though the hiring process in Silicon Valley favored candidates with theoretical knowledge of computer science, pursuing an noncodified path of entry into software development is much more acceptable than doing the same for law, medicine or civil engineering where the lack of a degree could disqualify an aspirant from consideration completely (Abbott, 1988). Therefore, these findings would not generalize to job areas where credentials or licensing are a prerequisite for practice.

Bibliography

Abbott, A. 1988. *The system of professions: An essay on the division of expert labor*. Chicago: University of Chicago Press.

Abbott, A. 1991. The Future of Professions: Occupation and Expertise in the Age of Organization. In P. S. Tolbert and S. R. Barley (Eds.), *Research in the Sociology of Organizations,* 8:17-42. JAI Press Inc.

Anteby, M. 2013. *Manufacturing Morals: The Values of Silence in Business School Education*. Chicago, IL: University of Chicago Press

Anteby, M., Chan, C. K., & DiBenigno, J. 2016. Three Lenses on Occupations and Professions in Organizations: Becoming, Doing, and Relating. *The Academy of Management Annals*, 10,(): 183-244.

Arbeit, C. A., Bentz, A., Cataldi, E. F., and Sanders, H. 2019. *Alternative and Independent: The Universe of Technology-Related Bootcamps*. RTI Press. https://doi.org/10.3768/rtipress.2018.rr.0033.1902

Arthur, M. B. 2008. Examining contemporary careers: A call for interdisciplinary inquiry. *Human Relations*, 61(2): 163-186

Arthur, M. B., & Rousseau, D. M. 1996. *The boundaryless career: A new employment principle for a new organizational era*. New York: Oxford University Press.

Ashforth, B. E. 2001. *Role transitions in organizational life: An identity-based perspective*. Mahwah, N.J: Lawrence Erlbaum Associates.

Bailey, D. E. & Barley, S. R. 2011. Teaching-learning ecologies: Mapping the environment to structure through action. *Organization Science*, 22: 262-285.

Barley, S. R. 2008. Coalface institutionalism. In R. Greenwood (Eds.) *Sage Handbook of Organizational Institutionalism*: 491-518. Los Angeles: Sage

Barley, S.R., Bechky, B. A., and Miliken, F. J. 2017. The Changing Nature of Work: Careers, Identities and Work Lives in the 21st Century. *Academy of Management Discoveries*, 3(2): 111-115

Barley, S. R. & Kunda, G. 2001. Bringing work back in. *Organization Science*, 12(1): 76–95

Barley, S. R. & Kunda, G. 2004. *Gurus, Hired Guns, and Warm Bodies: Itinerant Experts in a Knowledge Economy.* Princeton: Princeton University Press.

Baruch, Y. & Vardi, Y. 2016. A Fresh Look at the Dark Side of Contemporary Careers: Toward a Realistic Discourse. *British Journal of Management*, 27: 355–372

Beane, M. 2018. Shadow Learning: Building Robotic Surgical Skill When Approved Means Fail. *Administrative Science Quarterly*. 64(1): 87-123

Bechky, B. A. & Anteby, M. 2016. How workplace ethnographies can inform the study of work and employment relations. *ILR Review*, 69(2): 501-514.

Becker, H. S., Geer, B., Hughes, E. C., & Strauss, A. L. 1961. *Boys in White: Student Culture in Medical School*. Piscataway, NJ: Transaction.

Bidwell, M.J., & Fernandez-Mateo, I. 2008. Three's a crowd? Understanding triadic employment relationships. In P. Cappelli (Ed.), Employment relationships: New models of white collar work (pp. 142–178). New York: Cambridge University Press.

Bidwell, M.J., & Briscoe. F., 2010. The Dynamics of Interorganizational Careers. *Organization Science*. 21(5): 1034-1053

Bosu, A., Corley, C. S., Heaton, D., Chatterji, D., Carver, J. C., & Kraft, N. A. 2013. Building reputation in StackOverflow: An empirical investigation. IEEE Working Conference on Mining Software Repositories (MSR 2013). 89-92.

Cappelli, P. 1999b. Career jobs are dead. *California Management Review*, 42(1):146–167

Croidieu G., & Kim, P. H. 2017. Labor of Love: Amateurs and Lay-expertise Legitimation in the Early U.S. Radio Field. *Administrative Science Quarterly*. 63(1): 1-42.

Davis, G. F. 2016a. *The vanishing American corporation: Navigating the hazards of a new economy*. San Francisco: Berrett-Koehler Publishers.

Davis G. F. 2016b. Can an Economy Survive Without Corporations? Technology and Robust Organizational Alternatives. *Academy of Management Perspectives*. 30(2): 129–40

Demetry, D. 2017. Pop-Up to Professional: Emerging Entrepreneurial Identity and Evolving Vocabularies of Motive. *Academy of Management Discoveries*, 3(2): 187-207.

Ehrenreich, B. 2005. *Bait and switch: The (futile) pursuit of the American dream*. New York: Henry Holt.

Emirbayer, M. 1997. Manifesto for a Relational Sociology. *American Journal of Sociology*, 103(2): 281-317

Ensmenger, N. 2010. *The Computer Boys Take Over: Computers, Programmers, and the Politics of Technical Expertise*. Cambridge, MA: MIT Press

Fernandez-Mateo, 2007. Who pays the price of brokerage? Transferring constraint through price-setting in the staffing sector. *American Sociological Review*. 72(2): 291-317.

Fine, G. A. 1985. Occupational aesthetics: How trade school students learn to cook. *Urban Life*. 14: 3-32

Fine, G. A. 1996. Justifying work: Occupational rhetorics as resources in restaurant kitchens. *Administrative Science Quarterly*. 41: 90-115

Freidson, E. 1993. *Professional powers: A study of the institutionalization of formal knowledge*. Chicago, Illinois: Univ. of Chicago Press.

Glaser, B. G., & Strauss, A. L. 1967. *The discovery of grounded theory: Strategies for qualitative research*. New York, NY: Aldine de Gruyter.

Gray, M.L., Suri, S., Ali, S.S., & Kulkarni, D. 2016. The Crowd is a Collaborative Network. In P. Bjorn and D. Konstan (Eds), Proceedings of Computer-Supported Cooperative Work, New York, NY: ACM Press.

Hall, D. T. 2004. Protean Career: A quarter-century journey. *Journal of Vocational Behavior*. 65: 1-13

Halpin, B. W. & Smith, V. 2017. Employment Management Work: A Case Study of Theoretical Framework. *Work and Occupations*. 44(4): 339-375

Hollister, M. 2011. Employment Stability in the U.S. Labor Market: Rhetoric versus Reality. *Annual Review of Sociology*, 37, 305-324

Hughes, E. C. 1958. *Men and their work*. London: Forgotten Books.

Hughes, E. C. 1971. *The sociological eye: Selected papers.* Chicago: Aldine-Atherton

Ibarra, H. 1999. Provisional selves: Experimenting with image and identity in professional adaptation. *Administrative Science Quarterly*, 44: 764–791.

Ibarra, H. 2003. *Working Identity: Unconventional Strategies for Reinventing Your Career*. Cambridge, MA: Harvard Business School Press.

Ibarra, H. & Obodaru, O. 2016. Betwixt and between identities: Liminal experience in contemporary careers. *Research in Organizational Behavior*. 36 (2016) 47–64

Ibarra, H., & Lineback, K. 2005. What's your story? *Harvard Business Review*, 83(1): 64–71.

Ibarra, H., & Barbulescu, R. 2010. Identity as narrative: Prevalence, effectiveness and consequences of narrative identity work in macro work role transitions. *Academy of Management Review*, 35: 135–155.

Ibarra, H., & Petriglieri, J. L. 2010. Identity Work and Play. *Journal of Organizational Change Management*. 23(1): 10-25

Inkson, K., Gunz, H., Ganesh, S., & Roper, J. 2012. Boundaryless careers: Bringing back boundaries. *Organization Studies*, 33: 323–340

Jordan, B. 1989. Cosmopolitical obstetrics: Some insights from the training of traditional midwives. *Social Science and Medicine*, 28: 925–937

Kalleberg, A. L. 2009. Precarious work, insecure workers: Employment relations in transition. *American Sociological Review*, 74(1): 1–22

Kalleberg, A. L. 2011. Good Jobs, Bad Jobs: The Rise of Polarized and Precarious Employment Systems in the United States *1970s to 2000s*. New York: Russell Sage

Katz, L. F., & Krueger, A. B. 2016. The Rise and Nature of Alternative Work Arrangements in the United States, 1995–2015. *ILR Review*, 72(2): 382-416.

Katz, L. F., & Krueger, A. B. 2019. Understanding Trends in Alternative Work Arrangements in the United States. Working paper. National Bureaue of Economic Research. https://www.nber.org/papers/w25425.pdf

Knorr-Cetina, K. 2001. Objectual Practice. In T. R. Schatzki, and K. K. Cetina (Eds.) *The Practice Turn in Contemporary Theory*: 175-188. London: Routledge

Lakhani, K. R., and von Hippel, E. 2003. How Open Source Software Works: "Free" User-to-User Assistance. *Research Policy,* 32(6): 923–943

Lave, J., & Wenger, E. 1991. *Situated Learning: Legitimate Peripheral Participation*. Cambridge: Cambridge University Press.

Levy, S. 1994. *Hackers: Heroes of the Computer Revolution*. New York: Dell Publications

Marshall, H. M. 1972. Structural constraints on learning: Butchers' apprentices. *American Behavioral Scientist*, 16: 35–44.

Michel, A. 2011. Transcending Socialization: A Nine-Year Ethnography of the Body's Role in Organizational Control and Knowledge Workers' Transformation. *Administrative Science Quarterly*, 56 (3): 325-368

Moore, W. E. 1976. *The Professions: Roles and Rules*. New York: Russell Sage Foundation

Morrison, E. W. 1993. Newcomer Information Seeking: Exploring types, modes, sources and outcomes. *Academy of Management Journal*, 36(3): 557-589.

Newman, K. A. 1988. *Falling from Grace: The Experience of Downward Mobility in the*

American Middle Class, New York: The Free Press.

O'Mahony, S., & Bechky, B. 2006. Stretchwork: Managing the career progression paradox in external labor markets. *Academy of Management Journal*, 49: 918–941

Orr, J. E. 2016. *Talking about Machines: An Ethnography of a Modern Job.* Ithaca: Cornell University Press

Osnowitz, D. 2010. *Freelancing Expertise: Contract Professionals in the New Economy*. Ithaca, N.Y. : Cornell University Press

Osterman, P., & Shulman, B. 2011. *Good jobs America: Making work better for everyone*. New York: Russell Sage Foundation.

Petriglieri, G., Petriglieri, J.L., & Wood, J.D. 2017. Fast tracks and inner journeys: Crafting portable selves for contemporary careers. *Administrative Science Quarterly*. 63(3): 479-525.

Pickering, A. 1993. The mangle of practice, agency and emergence in the sociology of science. *American Journal of Sociology*, 99(3), 559–589.

Pratt, M. G., Rockmann, K. W., & Kaufmann, J. B. 2006. Constructing professional identity: The role of work and identity learning cycles in the customization of identity among medical residents. *Academy of Management Journal*, 49: 235–262.

Ranganathan, A. 2017. Train Them to Retain Them: Work Readiness and the Retention of First-time Women Workers in India. *Administrative Science Quarterly*, 63(4): 879-909.

Rheingold, H. 2000. *The virtual community: Home steading on the electronic frontier*. MIT Press

Riemer, J. W. 1977. Becoming a Journeyman Electrician: Some Implicit Indicators in the Apprenticeship Process. *Sociology of Work and Occupations*, 4(1): 87-98.

Rosenblat, A. 2018. *Uberland: How Algorithms Are Rewriting the Rules of Work*. University of California Press

Saks, A. M., & Ashforth, B. E. 1997. Organizational Socialization: Making Sense of the Past and Present as a Prologue for the Future. *Journal of Vocational Behavior*, 51(2): 234-279.

Saks, A. M., & Gruman, J. A. 2012. Getting newcomers on board: A review of socialization practices and introduction to socialization resources theory. In S. Schmidt (Ed.), *The Oxford Handbook of Organizational Socialization*: 27-55. New York : Oxford University Press

Schleef, D. J. 2006. *Managing elites: Professional socialization in law and business schools*. Lanham: Rowman and Littlefield.

Schwartz, D. 2018. Embedded in the Crowd: Creative Freelancers, Crowdsourced Work, and Occupational Community. *Work and Occupations*. 45(3): 247-282.

Scott, W. R., Holzman, B., Ris, E. & Biag, M. 2017. The Changing Ecology of Higher Education in the San Francisco Bay Area. In W. R. Scott, and M. W. Kirst (Eds.), *Higher Education and Silicon Valley: Connected but Conflicted*. Baltimore: Johns Hopkins University Press

Smith, V. 2001. *Crossing the Great Divide: Worker Risk and Opportunity in the New Economy*. Ithaca: Cornell University Press.

Smith, V. 2010. Enhancing employability: Human, cultural, and social capital in an era of turbulent unpredictability. *Human Relations*, 63(2), 279–300

Smith, V., & Neuwirth, E. B. 2008. *The Good Temp*. Ithaca: Cornell

Spradley, J. P. 1979. *The Ethnographic Interview*. New York: Holt, Rinehart and Winston

Strauss, A., & Corbin, J. 1990. *Basics of qualitative research: Grounded theory procedures and techniques*. Newbury Park, CA: Sage Publications

Susskind, R., & Susskind, D. 2015. *The future of the professions: How technology will transform the work of human experts*. Oxford, UK: Oxford University Press.

Tolbert, P. S. 1996. Occupations, organizations, and boundaryless careers. In M. B. Arthur and D. M. Rousseau (Eds.), *The boundaryless career*: 331-349. NewYork, NY: Oxford University Press

Trice, H. M. 1993. *Occupational subcultures in the workplace*. Ithaca, NY: ILR Press.

Turkle, S. 1995. *Life on the Screen: Identity in the Age of the Internet*. New York: Simon and Schuster

Van Maanen, J. 1973. Observations on the making of policemen. *Human Organization*, 32(4), 407-418.

Van Maanen, J., & Barley, S. R. 1984. Occupational communities: Culture and control in organizations. *Research In Organizational Behavior*, 6, 287-365.

Van Maanen, J., and Schein, E. H. 1979. Toward a theory of organizational socialization. In B.M. Staw (Ed.), Research in organizational behavior (Vol. 1, pp. 209-264). Greenwich, CT: JAI Press.

Vasilescu, B., Filkov, V., & Serebrenik, A. 2013. StackOverflow and GitHub: Associations between software development and crowdsourced knowledge . Proceedings of the 2013 International Conference on Social Computing, Sept., pp. 188–195.

Wang, S., Lo, D., & Jiang, L. 2013. An Empirical study on developer interactions in StackOverflow. SAC '13 Proceedings of the 28th Annual ACM Symposium on Applied Computing, p. 1019-1024

Weil, D. 2014. *The Fissured Workplace: Why work became so bad for so many and what can be done to improve it*. Harvard University Press.

Wenger, E. 1998. *Communities of practice: Learning, meaning, and identity*. Cambridge, U.K: Cambridge University Press.

Wilensky, H. L. 1964. The Professionalization of Everyone? *American Journal of Sociology*, 70(2): 137-158.