Lifelong and Lifewide Learning for the Perpetual Development of Expertise in Engineering

Aditya Johri

George Mason University

Abstract

For centuries, the profession of engineering has relied on college education to train entry-level engineers for the workplace. Increasing digitization of engineering and social practices has altered this relationship between formal schooling and development of expertise for professional engineering work. What does development of expertise look like when knowledge is generated and shared at an accelerated pace due to shifts in technology? In this paper I present case studies of two early-career software engineers. Using methodological insights from digital ethnography, I trace their professional journeys over two decades as they transition from high school to college and then their first job, continuing further for another decade as they transition across positions. I empirically demonstrate how development of engineering expertise is a continuous and perpetual endeavor and engineers learn throughout their lives (lifelong) and across all the different spaces they inhabit at any given time (lifewide). I argue for extending engineering work practices research and research in engineering education more broadly to take larger timescales of learning into account to build a comprehensive understanding of engineering expertise development.

Keywords: engineering expertise, lifelong learning, lifewide learning, virtual ethnography, digital participation

Introduction

College or university based formal education is the norm for training of engineers. Consequently, engineering education research has extensively examined the transition of a student into a professional engineer. On the whole, this research clearly shows that training received as part of a formal degree program is often incongruent with expertise required as an early-stage professional engineer (Brunhaver, Korte, Barley & Sheppard, 2018; Lee, 1986). In particular, students lack professional skills such as ability to communicate and work in teams, abilities that are essential for engaging with the practice of engineering as enacted in a professional work setting (Anderson, Courter, Mcglamery, Nathans-Kelly & Nicometo, 2010; Stevens, Johri & O'Connor, 2014; Trevelyan, 2010, 2007). This dichotomy of expertise emerges largely because the knowhow required of a student engineer to perform well academically often conflicts with skills needed to succeed at tasks that are commonly found in the workplace which are ambiguous and need to be completed under severe constraints.

Given the consistency of these findings over the years, higher education institutions have responded by developing and implementing curriculum and programmatic activities, including technical communication courses and team-based activities in courses, to train new graduates with relevant skills (Stevens, Johri & O'Connor, 2014). The transfer of knowledge from an educational program to the workplace is not easy though (Eraut, 2004; Jesiek, Trellinger & Nittala, 2017) and studies continue to show that newcomer transition, socialization, and integration into the engineering workplace continues to be a challenge (Trevelyan, 2019; Korte, 2018, 2009). As educational institutions struggle to catch up with engineering practices, influx of technology in the workplace has shifted the terrain of engineering work practices even further.

Within the workplace, change in practices of engineers has been unfolding for decades (Zussman, 1985) but the past couple of decades has seen an acceleration as the use of the Internet and the digitization of organizations has become the lingua franca of engineering work (Boland, Lyytinen & Yoo, 2007). In addition to using new communication technologies, starting with email, to more recent services such as Slack™, engineers are making increased use of computational technologies, novel algorithms and software, and automated machines and tools, to model, design, and manufacture products and services (Yoo, Lyytinen, Boland & Berente, 2010). This has resulted in a concurrent shift in how organizations work – they have become more distributed, both geographically and virtually – and skills engineering need to be successful. Shifts in communicative and collaborative practices, talent and expertise recruitment, and the roles professionals have to adopt beyond providing technical know-how have further made the school-to-workplace transition challenging (Johri, 2012a; Johri, 2008).

Building a comprehensive understanding of how engineering has transformed with digitization and ways in which this might impact training of students for professional life and career is an ongoing process. What is clear though is that in conjunction with changes in workplace practices with digitization, there is also a corresponding shift in expertise required to practice engineering, especially when expertise is viewed as practice-based enactment of skills or knowledge (Clancey, 1997; Johri, 2014, 2015a). A focus on developing expertise over time, as opposed to knowledge acquisition through short term activities, is consistent with contemporary theories that conceptualize learning as engagement and a part of a process of becoming and identity development (Beach, 1999; Nasir, 2002; Wenger, 1998). In this context, what does expertise look like and how does one develop it? How is the digitization of engineering practices and the availability of highly capable tools across life settings changing the transitions between

school and workplace? What, then, might be a new way to think about expertise in the long-term over a career? These are the larger questions that guided this study.

In this paper I attempt to extend our understanding of the development of engineering expertise through the case study of two software engineers whose careers I trace, through interviews and archival material, for over two decades. I track the development of interest in working with "computers", their participation in open-source communities, internships at firms working on open-source software, and future jobs in both open-source and traditional firms. Open-source provides a fruitful context for this study since much of the information and data related to work practices is publicly available for research purposes and also because it typifies a context of engineering – open and participatory – that is becoming more prevalent with digitization of the engineering workplace. Although there is an argument to be made that the cases examined here are common in the software engineering profession due to the use of software code, which is easy to share, and not that relevant to other engineering disciplines, given the profusion of digital fabrication and devices, and the availability of digital data, tools, and sharing of expertise online, practices of software engineering will become increasingly common across other engineering disciplines. This case study can serve as a canonical view at the future of engineering practices and expertise. In the rest of the paper, I start by reviewing prior work on learning that speaks to lifelong and lifewide learning, discuss my methodological approach – a combination of trace ethnography and netnography – and then presenting the findings. I end with a discussion that draws implications for engineering education research.

Lifelong Learning Pathways

A long-standing criticism of research on learning has been a disproportionate emphasis on short-term experiences whereas there is strong evidence that a longer-term focus adds to our understanding of how people learn both as an evolutionary process but also through participation in cultural practices (Cuban, 2001). Since the early 1990s a decidedly practice-based shift in the field of the learning sciences has challenged researchers to go beyond examining and articulating psychological processes that focus on near-term knowledge acquisition, including rote learning, to better understand learning as a social process of becoming that takes place across longer scales of time and, within and across the multiple spaces inhabited by a learner (Cole, 1996; Lave, 1996; Lemke, 2000, 2001; Roth, 2001, 2014). Largely this turn has been identified with the "situated paradigm" of learning (Johri & Olds, 2011; Lave & Wenger, 1991) but there is scholarship from social scientists across a spectrum of fields that shows how an understanding of historical context, timing, social networks, and access to various forms of social and human resources are necessary to explain learning over time (Elder, 1994; Saxe & Esmonde, 2005; Serpell & Hatano, 1997).

Attention to situated practices or cultural arrangements along with a renewed focus on the learner as an active contributor to his or her own development, highlights the important role of engagement and meaning making for sustained participation in learning activities (Barron, 2006). The activities leaners engaged in might be of short or long durations but together they constitute trajectories of learning (Dahlgren, Hult, Dahlgren, Segerstad & Johansson, 2006) that result in longer term, extended, learning pathways (Barron, 2010; Bricker & Bell, 2008). Practice-based participatory views of learning draw attention to opportunities for these longer-term sustained participation through membership in communities of practice that are defined by affinity groups (Gee, 2000) or interest-driven activities (Wenger, 1998). Members of affinity

groups come to develop practices and sets of experiences that position them to engage the world in ways that offer continual opportunities for learning.

In most of our current research withing engineering education, these viewpoints and longer timescale empirical work is missing. A few studies, largely panel-based research, have followed students throughout their studies (Stevens, O'Connor, Garrison, Jocuns & Amos, 2008) or as they have entered the workforce and then reported on their acculturation and socialization process in the workforce (Brown, Montfort, & Frye, 2013). But for the most part the focus has been on short-term projects: courses that last a semester or term, or occasionally, as in the case of capstone, a yearlong project. Although we have generated an extensive body of knowledge based on the current time span of our empirical studies, the downside of not looking at the longer time periods and different contexts is an inadequate view of engineering education and development of engineering expertise. In particular, we have an incomplete picture of what a student might learn from non-formal communities they participate in and other opportunities they have for developing expertise.

Lifewide Learning through Participation in Situated Practices

Lifewide learning refers to the different kinds of learning that take place in the distinct, but yet overlapping, spaces we occupy at any given time. When we are at school, we are also at home, in the playground, or, even in the workplace. With increase in the proliferation of mobile devices and ever more digitization of learning practices and ways of communicating, the spaces, and places available for learning are multiplying, further increasing the possibilities of creating new "spaces" or places to learn thereby expanding the learning ecology of learners (Barron, 2004) and diversifying their trajectories and pathways (Barron, 2010). These are not just typical

learning spaces but also people working from home, from an office, or even from coworking places. In each of these contexts, they find interaction support through email, instant messaging, social media, and related products. These spaces, even fully digital ones, are crucial to consider in mapping learning trajectories and development of expertise especially for current students and the future workforce that is largely grown up digital (Brown, 2000). In many ways, learners now inhabit a vast networked learning landscape and these networked environments themselves are nested within larger social structures extending beyond formal educational opportunities (Siemens, 2005).

Expertise, Agnew, Ford & Hayes (1997) argue, "emerges from a dynamic interaction between the individual and his physical/cultural domain (p. 221)," and "interplay between cognitive and cultural/social processes (p. 225)." Clancey (1997) further echoes this viewpoint and contends that expertise reflects knowledge that develops and has value only within enacted activities, "problems arise not in selecting facts, but in conceptualizing how we should view the activity we are currently engaged within (Clancey, 1997, p. 275)." Spaces of participation therefore, according to Clancey (1997), are essential for generating *tacit* knowledge that is central to expertise, as they guide events which provide an interpretive frame Johri, 2015a). The learning that we encounter in these spaces differs from one space to another – it is situated in the practices we enact in any given space (Johri, 2012b, 2014) – but it all comes together to create the skills, knowledge, and expertise we possess. What long-term participation in networked digital spaces thus provides, which many formal educational environments fail to do, is an opportunity to learn and develop expertise that emerges from participation across spaces and is thus qualitatively different and useful across other spaces (Weller, 2007). It is also has value as distinct spaces of participation allow for learning of skills such as leadership, task assignment,

and timely completion of work, among others, that extend beyond technical knowledge, which is often the domain of formal education.

Research Study

Research Methodology

The research approach used to develop the case studies I present can be characterized as digital ethnography that draws on two traditions within that larger paradigm: Netnography (Kozinets, 2002, 2006) and Trace Ethnography (Geiger & Ribes, 2011). In netnography, a significant proportion of the data collection and participant-observation originates in the data shared freely and publicly through the Internet (Kozinets, 2006). The method does not preclude face-to-face or place-based data collection and also includes interviews conducted via email or audio or videoconferencing (Kozinets, 2002). Netnography has emerged to as a method for capturing the transient nature of digital life, especially within communities sustained by many weak ties but the strength of the method is contextualized data and therefore developing an understanding of the overall environment is crucial for interpretive analysis.

Trace ethnography is an approach that focuses on several locations or vantage points (Marcus, 1995) to build an understanding in contrast to traditional ethnography that relies on a single site and perspective. For Geiger & Ribes (2011), trace ethnography "combines the richness of participant-observation with the wealth of data in logs so as to reconstruct patterns and practices of users in distributed sociotechnical systems" (p. 1). They emphasize the advantage of the approach since traces of different sorts are omnipresent in today's technological systems and because in many instances these traces are the primary mechanism for interaction within a distributed community. Hasu (2005) refers to trace ethnography as an "ethnography of

change" (p. 90) that serves to understand, for instance, the "invisible work" that takes place in organizations. The combined netnography and trace ethnography approach is suitable for the study I conducted as both participants contributed to many online platforms, especially as they also developed open-source software, and left a trace of digital artifacts online in addition to maintaining detailed online profiles that depicted their life transitions and experiences.

Data Collection

I first met both case study participants, whom I will call Roberto and Bruno, in-person in 2008 while I was conducting a large-scale study of professional global engineering work practices at a field site in Europe. The firm they worked at was a large technology multinational headquartered in the United States. At that time, I conducted interviews with both participants, in addition to observations at their workplace. The interviews lasted a little over an hour and I also chatted with the participants informally multiple times during the week I spent at the company. During my interview and conversations with them, both participants provided me with information about their online presence – which included their blogs and their contributions to open-source software communities. In the open-source community, blogging and using microblogs, such as Twitter are common ways for sharing knowledge within the organization and also with the larger community (Johri, 2011; Johri, 2015b). I also connected with them on LinkedIn through which I was able to get updates about their professional moves and intermittently followed their Twitter account to learn more about their professional life. Both participants are active online although one of them is more so than the others. The final interview with one of the participants took place in July 2020 using Zoom.

This research was covered by the Institutional Review Board (IRB) and all consent procedures were followed. To protect participants' identity, I am not disclosing precise information about the data and am suppressing some data but the overall online data I used for this study include Blog posts (200 printed pages), Twitter messages (over 65,000 Tweets), LinkedIn profiles, GitHub profiles and contributions, Mailing Lists Postings (public), presentations and talks at conferences (public). To demonstrate the kind of data that was available to me, beyond interview transcripts and observations, I present two tables (Table 1 and Table 2 below) that contain blog postings and Tweets I have culled from the data corpus available to me. I have combined data from both participants in the table, once again, to maintain their privacy. The case study data collection and analysis are unorthodox and presented many challenges. More than anything, it required interpretive judgements that had to be supported by knowing a lot more about the context of work of the research participants. The future of field study is going to be increasingly digital and it is important for the field of engineering education to think of ways to collect data and to gain access and utilize new methods for research (Jesiek, Johri, Brozina & Korte, 2020).

Blog Postings (Selected sample from 2006 - 2020)

Job Transitions

2020

Changes are risky, taking on a new role on a new company with people you never worked before, growing a whole org from scratch is hard work that comes with a lot of uncertainties. When I decided that I wanted to try something new and join a large company to work on product upgrade in a new city I knew that it was a leap of faith.

2019

I would also like to make a special mention to my manager, who has supported me all the way in several ways both professionally and personally. There is this thing people say: "people do not leave companies, they leave managers," well this is certainly not the case for me. I have found not only a great manager but a true friend. As for my experience at this company, I have never lasted more than 2 years in the same spot before, I truly found my place there, deep in my heart I know I will always be attached to this company, but there are some things I want to try and learn elsewhere. This job switch has been the hardest departure I ever had and in many ways it breaks my heart to leave. If you are considering joining this company, do not hesitate, there is no better place to write and advocate for open source.

2018

A year and a couple of months ago, I was asked to pivot a little bit from working full time on a product to manage a new team we were building to work on client hardware enablement with an emphasis on upstream. The idea was to fill the gap in the organization where nobody really owned the problem of bringing up new client hardware features vertically across the stack (from shell down to the kernel), or rather, ensure both products work great on modern laptops. Part of that deal was to take over the bootloader and start working closer to customers and hardware manufacturing partners. At first I hesitated as I wasn't sure I could do a good job, y'all know how the impostor syndrome works specially outside of the comfort zone, also had very little engineering experience on the kernel or hardware related fields outside the hardware

design I did at university. However, after some thinking I thought this was a terribly exciting prospect and I had some ideas as to how to go about it and do a decent job. Fast forward 16 months and I'm loving it, in a relatively short period of time we've been able to build an amazing team that have been able to execute quite a few important highlights to make both products work better on laptops.

2014
I just moved back to Europe, this time to a foggy town, to join the Open-Source at [electronics company] where I will be contributing upstream to different packages during the day and ironically mocking the local hipsters at night.

2009
In three weeks, I will be moving to [a new country], to work with [name] and the rest of the [company name] team! How cool is that? Exciting times ahead!

2007
This is the first time that I'm going to leave home (and country), so, is a really serious change for me: no more mom's food, no more quite weather, no more my mother tongue. On the other hand, this is the first time that I have a real opportunity to work on a big company (and doing free software!), so I won't waste it, I want to give my best and learn as much as possible (even Java!), probably, the most exciting thing on my life since I passed my first college exam.

Community Participation
2013
I just returned from the Open-Source Conference in Europe and I'm still processing the extremely interesting conversations we had there. Once again the conference was a great success, fascinating and great fun. Big kudos to the awesome organizers. [Lengthy description of the event with details about the keynote, demos, and technical updates, and recap of hallway conversations.]

2006
I've been very busy last weeks, so I'll summarize what have been happening around. A couple of weeks ago went to attend the Open-Source where I gave a talk about software integration and the Web 2.0. There were some interesting talks, among them, [name] spoke about the new stuff that will come soon.

Tutoring/Teaching
2020
So, this is my progress thus far, I think once I get to a point where I can iterate over the concept, it would be easier to start sketching the mapping. I would love feedback or help, so if you are interested please reach out by filing an issue on the Gitlab project!

2019
In [a programming language] there are two standard ways to interact with strings, the String type, a dynamic utf-8 string that can be modified and resized, and &str, which basically is a bare pointer to an existing String. It took me a while to realize that internally a String is not null terminated and can contain many null characters. This means that the internal representation of String is not compatible with C strings.

2017
I spend most of my time writing and reading C code, but every once in a while I get to play with a C++ project and find myself doing frequent reference checks to cppreference.com. I wrote myself the most concise cheat sheet I could that still shaved off the majority of those quick checks. Maybe it helps other fellow programmers who occasionally dabble with C++.

2016
A few months ago, I spent some time to learn some basic [a programing language], I was interested in getting an informed view of the language, specifically about the safety and concurrency idioms as well as its compatibility with the C. I must say I was pleasantly surprised with the tools. More recently though I've been investing time to actually understand the memory ownership model and how it plays with channels/concurrency. I must say that what these guys have achieved is really clever with a language that once you get the grasp of things feels actually really nice to use.

2013
So how do you use temporary URLs? Glad you asked. Let's assume you have the following installation:
The first thing we need to do is add temporary URL secret keys to the account.. At this stage you must be happy to see how easy and convenient this is but wondering how you can integrate it with your code. You can replicate it with the following block of Python code:...That's all for now. What else would you like to learn regarding it?

2006
I'll explain how to create multithread applications showing a simple example. First we'll write the basic code, we need the window, the progress bar and the exit callback that will be used to destroy the threads. The position of the progress bar is set with the set_fraction() method, which argument is a float between 0.0 to 1.0.

Recommendations
Thanks to [name]'s brilliant answer in cstheory.stackexchange.com I have been reading some of the famous data structures and algorithms used in the Linux Kernel. And so can you.

Happy Birthday Grace Hopper
By the way, for anybody wondering why the COBOL in the Google Doodle for her today: "SUBTRACT CurrentYear from BirthYear GIVING Age Display Age" doesn't return a negative value, it is because age is defined unsigned in the data division. Since ages can't be negative, and COBOL handles this negative return to an unsigned value Gracefully.

Work Practices
2014 Leadership
You want to make things better? Why don't you start by learning how to work with others and contributing yourself? You think we need better leadership? Why don't you learn what it takes to become a leader? (hint: your blog post doesn't help)

2008 Skill Transference
I feel quite lucky for having such opportunity, and I would like to thanks [name] for recommending me in the first place, without him this turn in my life wouldn't be possible. I would also like to thanks to the whole GNOME community for being such a good software development school, during this year I've realized how much I've learned since my involvement back in 2003 I now realize that I've wouldn't been as valuable for my first company as I am without all the skills I learned inside such a great community.

2007 Team Building
This weekend I've been in Europe for team building, the first of the team ever. The team is 6-year- old already (although I've been here for one month), and they've never had the opportunity to stay together without thinking about work for a couple of days. For me also, it's been a perfect introduction, since I've been able to meet the people on the German side of the team. The idea of being out of work for a weekend it's been around way far before I started working at here, but the activity itself was kept in secret until the last week.

Tweets (Selected sample from 2016-2020)

Apr 2016
Proud to have my first bug fix in a new system I am learning accepted #Linux #filesystems #hacking.

Apr 2016
"For those who don't know what a while loop is, it's the 'mommy are we there yet?' loop." - liking my statistics class at work today.

Apr 2016
learning the internals of filesystems. Filesystems are super interesting yo!

Apr 2016
some subsystems have bugs in kernel. Reading the code is a great way to learn and find small things.

May 2016
"In thirty years' time, as technology moves forward even further, people are going to look back and wonder why offices ever existed."
Working across distributed teams: Assume no malice. When in doubt, reach out.

Jun 2016
What are you doing in the kernel? Upstreaming it? what do you guys use it for? any secret features I should know about? Recommended usage guides?

Jul 2016
I did. I have 2 domains at my current job. Most people have 2, it's fun to learn new things.

Jun 2017
Last day at my current job. Very grateful for the amazing personal growth and fun times. Sad to leave, many cool people here.

Jun 2017
Day One at my new job. Very excited for the new challenge! So much to learn and play with.

Jul 2017
Week 2 of turning into a morning person. Great life decision. Days feel longer, more hours and tons more productive. 9/10 would do again.

Jul 2017
The colleague seating besides me has a copy of Effective Modern C++ on his desk. Many "effective" books for C++. Read the C++ Primer but it was mostly about the standard libraries. Dry. Listing all API interfaces. What's your recommended book for a C programmer to become fluent in C++? Mostly syntaxis since I'm comfortable with Python, Rust and GObject

Sep 2017
Having tons of fun learning the basics of 3D Graphics with https://learnopengl.com! and a friend at work tempted me to write it in.

Mar 2018
The best way to realize your progress... look at code you wrote a few years back. Ignorance was bliss back then.

Mar 2018
Since last year. He knows how to be a digital nomad properly.

Sep 2018
Spending the week in the Amalfi Coast disconnecting from bugs, sprints, launches, tickets, and deadlines.

Sep 2018
Learn by contributing. The hidden benefit of giving back to #OpenSource software. Especially for companies which have tech in their core competency.

Jul 2019
“The chase is better than the catch,” Motörhead. In programming, the result is rewarding, but the journey is where we learn.

Jul 2019
Start with just using it to make collapsible tree-based notes: Headings, subheadings, section content. Then slowly when used to it try out features: to-do tags, calendar events, references, etc. I use 10% of org-mode, but I use it everyday.

Jul 2019
Interns should do normal team dev work. Nothing critical path that could block others, but the only way to learn is to do the real thing. Just with patience and tons of knowledge sharing/pairing. Bonus: they had fun and will want to come back and you know they are great.

Aug 2019
Very excited to have my first patch land in the new software I am learning! Small UI papercut fix, but you have to start somewhere. Smiling face with open mouth.

Aug 2019
Open Source is awesome! Feeling very welcomed to the new community after @Name gave me a shout-out in the for my first patch. Awesome community Smiling face with open mouth.

Nov 2019
Why do all programming language books start by teaching you the basics of programming? Surely there is a market for developers wanting to learn a new language. For example: I'm happy in C, Python and Rust. But I need to understand Java better for the occasional project at work.

Jan 2020
My first week at my first graduate job I was advised “never admit you know anything about Printing or Email. It’s all you’ll ever do from that point onwards.” That person was a kernel engineer working on the sendmail implementation.

Jan 2020
Today was one of those days where, after trying to code for 4 hours, you convince yourself that your whole career was just sheer luck and you don't know anything.

Feb 2020
Where I am from, non-PC platforms were an extreme rarity for consumers. The only exposure to computers I ever had were x86 based PCs. My dad had UNIX at the office but was hard to understand what was going on in those machines outside of the DOS/Win311 emulator.

Feb 2020
Oh, yes. Absolutely. IRC was my internet wow moment. I ended up using Linux because I was told it was harder to hack your PC through your public IP (which at the time was advertised in IRC Hispano and I had several BODs a week from pranksters)

Replying to: Let me tell you my first Linux install was not fun. The lovely people at open source helped me lots. That was the WOW moment for me, finding a community of people that were as fascinated as I was about the PC as a platform and Linux as a tinkering OS.

Feb 2020
Shout out to the person who coined the term “nerd immunity.” I knew our time would eventually arrive. “Coronavirus: Glasses wearers less likely to get COVID, study says.”

Feb 2020
Then closed source is a philharmonic where only the privileged can attend and the super privileged can play at. Reply to “I like to think about OpenSource projects as bars where musicians are welcome to join and take part of Jam sessions.”

Feb 2020
So true it hurts. “Open source infrastructure is like 75% maintained by people like me who tried to build an adventure game in their teens but

found out that there was no good memory allocator for that (or whatever), and instead ended up holding the hot potato of malloc (or whatever) for 25 years."

Feb 2020
So, I occasionally worry about where our next generation of volunteers is going to come from. One day all the maintainers of all the current pillars of the internet are just going to retire, or die, and everyone left will find that their castles are built on sand.

Mar 2020
The ones that are stiff tend to have a biased understanding of what the role is and tend to have a handicap about allowing people that add value in different dimensions other than deliver code or resolve X number of tickets per month.
Replying: I think this is a topic that changes greatly depending on the company. Some have a huge flexibility around headcount and titles. Others have very specific expectations about each role and doing things outside of the box that are valuable is detrimental for career prospects.

Mar 2020
To my Linux graphics peers, is there a distributed way to list available graphics cards that does not require root privileges?

Mar 2020
People who have never worked with Rust or Go tend to underestimate the degree to which dealing with C/C++ feels like banging rocks together.

Mar 2020
They need to stick around long enough though... and keeping the right culture without the right people in leadership positions to keep you in check is really easy to derail culture. I've seen then often enough sadly :(
On diversity, yes junior positions are a bigger pool, but you also need to commit enough mid and senior diversity hires so that junior developers don't feel tokenized and have someone they may identify with to look up to and engage with
Replying to @ In ideal conditions you are right, but a junior with poor performance can be quite a headache for everyone, it is a riskier hire also, I struggle A LOT mentoring junior developers effectively in distributed teams.

Mar 2020 Retweeted
I'm helping build a team to work on edge computing and automation at my current company! Who wants in? We're hiring for multiple positions, junior up through senior candidates. Hit me up!

Mar 2020 Retweeted
If you're a first-time contributor to curl and you want to tell me what we did right and wrong when your work as received and handled, I will be thrilled to listen to everything you have to say. And then do better for the next first-timer in line.

Mar 2020
Crazy Linux kernel feature I learnt about today: when naming network interfaces from userspace, kernel resolves format strings.

Jul 2020
Nice! This is what my team has built to support our product in the cloud. Smiling face with open mouth.

Case Studies

When I met them, both Roberto and Bruno were working in Ireland with a large engineering firm in computing hardware and associated software. Neither of them were from Ireland but from a small island in Southern Europe and had moved to Ireland for their jobs. I interviewed them as part of an effort to understand how their team worked with its counterparts in the USA, UK, and China. During the interview I realized that neither of them had taken a traditional – formal education followed by a fulltime job –to their current professional position.

They had started contributing to open-source projects when they were in high school and built a reputation among other contributors that led to their current positions. This helped them land an internship while they were still in their undergraduate programs. Open-source software, which depends on contributions from volunteers, emerges as a strong through line in their narrative. By its nature, participation in open-source projects provides opportunity to learn many of the skills that are traditionally associated with work practices and which are critical for the development and demonstration of expertise. Furthermore, as software and engineering knowledge changes, in open-source you are both an expert at something but also a novice and newcomer in another area (Johri, 2018; Von Hippel, 2001). Subsequently, they joined the firm full time while still completing their degree. Although both Roberto and Bruno followed a similar initial trajectory, their paths started to diverge after their first job together. Roberto continued to work on open-source projects as part of a large open-source company, while Bruno ended up using his technical expertise to work on similar projects but as part of a larger technology company that was not solely open-source but had some open-source projects. They both ended up working in different subunits of the same organization for a while before Roberto moved to work for another open-source company.

Case Study 1: Roberto

I first interviewed Roberto in November 2008. At that time, he had just started working in a large technology multinational as a full-time employee after interning there for a year. He had moved to Ireland for the internship and was hired largely because of his open-source related experiences. In his current project he was assisting the company create an open-source version of their propriety product. As part of his day-to-day activities, he worked on webpage development

for that product, creating a code repository, and creating a community of developers to assist with the product. Aside from his day job, he was also deeply involved with another open-source product called GNOME which is desktop software like Microsoft Windows ™. At the time of the interview, he was co-organizing the next developer summit for the product and was also working on improving the overall platform to attract more developers. He characterized his participation in GNOME as "active" and stated that for him that implied within that community "people know me and they know what I've contributed." He started out as a lurker, just observing the product and the conversations around it, and slowly moved to a helping position, improving his English all the while.

Roberto started working on open-source software when he was fifteen years old and a student in high school. His father had an old UNIX machine that he wanted ported to Linux and he could not find any local expertise in their small town to help him. So, Roberto took on the task and began his journey in open-source software. After high school he joined the local university and in his second year joined the open-source office of the university. The university was actively promoting the integration of open-source software across the university and Roberto started working on this project. As part of his job, he organized talks by open-source experts. When the programming language Java was released as an open-source language in 2006, a Java contributor from his country who worked in Dublin was invited to give a talk. After the talk Roberto had lunch with the speaker and the speaker informed him about internship opportunities with his company in Ireland. This led to his current position.

Roberto outlined the following process related to his open-source participation. He started out by reading the software code related to the project he was interested in and began monitoring messages on the online discussion forums associated with the project. He realized

that to have an impact and build his reputation it was important to make a visible contribution to the software and bring something new to the community. He started making small contributions and blogged about them and then posted that information on mailing lists. Slowly he started to get credit for his work and became recognized with the community. In his interview he mentioned that contributing code is the most important thing in open-source software. In addition to his contributions, he also started becoming active locally and organized several events.

Roberto reported learning how to work across time zone as a skill he developed, especially the use of different communication media. He collaborated with developers across the globe including France, UK, US, Germany, and Canada. To work with them he used email, IRC, and many other tools that he picked up as he worked on code and problems. He picked up many nuances such as a lack of use of audio technology due to language issues and a preference for written communication. Although Skype was used as an audioconferencing tool, he had to pay constant attention when using audio or voice since language was an issue as most project contributors did not speak English as their first language. Informal communication was often preferred over formal communication and Internet Relay Chat (IRC), a text-based communication medium, was used commonly. Furthermore, he stated that phone demands immediate attention whereas with IRC you can finish what you are doing/check and get back, it is easier to switch and multi-task. Also, IRC leaves a track of everything making it easier to follow-up.

Roberto's participation in the open-source community and thereafter in a traditional software role within a large company provided him with the opportunity to learn about different work environments. In his interview he commented on the different levels of hierarchy in different environments and on roles taken within the firm. He observed that comparing his open-

source participation with working in a traditional large technology company he was amazed at the power differentials and authority issues. Specifically, he brought up the approval process that one had to go through to get anything done at his current company. He also said that compared to open source, he felt as if he was involved very late in the decision-making process. He said that through his participation and conversations with other developers he had realized that different communities have different norms. He stated that GDK development was technical as well as organizational participation and he had learned a lot about how a community thrives and barriers to reaching that by his current organization.

Over a 15-year period, he had worked in four countries, four companies, multiple open-source projects in addition to these formal positions and developed skills both technical and professional. Throughout his career, Roberto received help from others and in his blog there are many references to the people who helped him on different projects and even assisted him in finding jobs when he wanted to move on for professional or person reasons. In spite of his stints with private technical companies, Roberto always felt more comfortable with open-source philosophy and moved back to his roots at the time of this writing.

Case Study 2: Bruno

Bruno was working as an intern and had been at the firm for six months when I first interviewed him in November 2008. Bruno studied in an American school in Spain and then spent his sophomore year of High School in the United States. He was from a small town in Southern Europe and after starting at a local university he moved to Madrid, Spain, after his first couple of years to finish college. He started out in open-source by working on the GNOME project and then became a regular contributor to Ubuntu. He had been working on open-source

projects for the past five years. He started looking at Debian packing 3/3.5 years ago and for the last two years he was an official Ubuntu developer. He had significant experience on packaging and integrating software and porting software and experience with both GPL and BSD. In his current position he was working on a team that was open sourcing a product and his job was to figure out ways in which to make contributions to code repositories easier for outside contributors. He had been offered a fulltime position by another open-source company but he preferred his current firm as he wanted to gain experience working face-to-face rather than in a virtual firm.

Bruno liked to develop multimedia related applications. When he found out that there was a new project in Ubuntu looking for multimedia integration he started making contributions. When the project leader stepped down due to other commitments, Bruno took over the project leadership. His project has 10-12 people spread all over the world and they communicate through IRC and email but prefer Skype. He has met face-to-face with most of his colleagues as there is an Ubuntu developer summit every six months. He had learned that making a useful contribution to open-source was hard. He said that in Ubuntu people grab the easy and more visible stuff and not the hard stuff. He said that many people get started but cannot stick round for long. He gave an analogy of playing guitar. The cost of participation is low but very few people stick around; people must be self-motivated and learn the basics such as reading music. He got started by reading documentation about the project and the examples and codes. Once he contributed, if he ran into trouble, he posted a question on the list and got help easily.

He said that he learned how communities work and how one can participate in them and he was using his knowledge to support newcomers to the community. He made sure that the first contribution for a newcomer is relatively easy as newcomers need to be able to make that first

contribution as it is a great motivational exercise. He also said that he ensured that there is a balanced learning whereby archival resources are supported by firsthand feedback. There must be effort from both sides – learner and mentor – and the learner must be given tasks and opportunities to provide themselves. The mentor needs to model good effort and then that leads to efforts from others.

In open-source the other important thing is meritocracy and visibility of that "you've to show-off what you are doing as well." He also commented that it is important to demonstrate that you can finish what you've started. He said that small prototypes are important but the effort needs to go beyond that and result in a meaningful contribution the community.

After working with the large company for some time, Bruno worked fulltime in an open-source company that made collaboration software and then in the open-source research lab of an electronics company. Now he works for a large technology firm on their video projects. He still contributes to open-source software projects and in his online profiles he described himself as open-source developer, Linux hacker, and a curious geek. He maintains an active GitHub page that list his projects as well as shared code.

Bruno commented on many of the same organizational practices that Roberto did and he also found the current company a lot more bureaucratic than Ubuntu. He also commented that he found the decision making in Ubuntu a lot more democratic as compared to his present company. He said that he had learned at both places that working with 'coders' is not easy as they are not very social and friendly people. He said that his experience had made him realize the benefit of non-technical skills and he had realized that they go hand in hand if the goal is to achieve something like completing a project.

He said that in his internship he was applying many things he learned working on Ubuntu as an open-source participant. He had realized that it is important to keep things interesting for contributors to keep them motivated and he did so by providing a mix of tasks to contributors – what people want to do with what needs to get done. He made sure that if contributors got a boring task to do, they also got a "really cool" task to go with it. He said that he admired the current horizontal structure and liked the fact that in the company, similar to open-source, employees were not judged by "what they have in their CVs, like a PhD or 10 years with the company, but your expertise and ability to get the job done".

Discussion

In this paper I present case studies of two software engineers to highlight the complex nature of expertise development across the life span. Learning is lifelong and lifewide – engineers learn throughout their lives and they learn in all spaces they inhabit at any given time, and over time. Developing expertise, therefore, is a constant endeavor and it has become more complex in the digital knowledge economy with shifts in how engineers work. Over a period of almost two decades, Roberto and Bruno consistently updated their expertise and even though they would not be considered newcomers in the sense of moving from formal education to the workplace for the first time, their acculturation to new work opportunities had that same spirit.

In their careers, Roberto and Bruno moved across half-a-dozen countries and interacted with people in dozens of locations. In addition to the physical movement, they participated digitally across communities and learned the ropes of software development through their online participation. Their digital participation had supported many of the core socio-cognitive factors necessary for learning such as collaboration, knowledge sharing, and meaning-making by joint

activity. It helped in the development of expertise that was situated and contextual but also transferable. For Bruno and Roberto, participation in affinity-based online communities for interest-driven learning was a core feature of developing expertise. Whether it was the domain of audio software or an operating system which they found interesting, wherever they could contribute they dd, and in return they got credit and a sense of belonging that were critical in expertise development. The online ecosystem now available for developing expertise constitutes a new form of apprenticeship where the guidance is through a networked form of participation. Although the ties are not as strong as in traditional place-based apprenticeship, but expertise is available from many more sources and of a more diverse nature. Expertise development is aided through the ability to be mentored and mentor others.

What expertise did Roberto and Bruno develop through their participation? It is clear from the cases that their expertise can analytically be divided into two categories – technical and non-technical, although in practice expertise developed along these two lines in an intertwined manner. For instance, the knowledge to work with open-source software developed through practice, trial-and-error, problem-solving, reading material online; concurrently, the expertise to learn from others and work with others was also acquired. The bigger their transitions from one company to another, the greater was their need to develop new expertise of both technical subjects and also of how to work in the new context. Expertise development was a perpetual undertaking and they continued to learn new technical skills as well as new ways of collaborating, managing people, and other skills that are required of a professional engineer, throughout their career. When it comes to engineering practice, the literature often talks of boundary-crossing and the ability to work across boundaries of all types – disciplinary, temporal, language, among others. The case studies presented here illustrate that boundaries are emergent;

therefore, the ability to keep learning and being able to increase one's repertoire – a marker of one's expertise – is crucial for long term success. Bruno and Roberto often faced obstacles in their work but they crossed these boundaries thereby leveraging new opportunities for learning and working. In many instances, they became boundary-spanners who were able to navigate new boundaries not just for themselves but even for others.

What does this study contribute to our current understanding of engineering practice and the development of expertise by early-career engineers? Although at some point an engineer is "early-career", as in, in the workforce after completing their degree, but this depiction is not true for all engineers who are in the workforce or entering the workforce now. Especially in computing related fields, participation in non-school professional practices often starts in conjunction with a formal program. Consequently, the relationship between "taught" engineering practice through formal engineering programs and engineering practice in the workplace has become more complicated. There is an additional layer of complexity as engineering practices are changing rapidly with technology and reinvention is needed to keep up; things are different than they used to be and jobs that last an entire career are rare. This does not mean that school-learned knowledge is redundant, as Roberto indicated when he mentioned that he had found most courses he took in college useful at some point or another, but in terms of developing expertise they are insufficient. It is difficult to see how a college or university can assist with gaining expertise beyond a certain level and type. Faculty members who prepare students for the workplace, especially in the U.S. but in other places as well, often do not have professional work experience as they are hired largely for research competence and lack expertise of professional practices.

This study provides a cautionary tale if we think about all those who are unable to participate in the new knowledge economy and are unable to build the kinds of expertise, networks, and mobility that is needed to participate. Both participants in this study were lucky that they not only had access to technology but also that they could physically move across borders and legally participate in opportunities. This is often not the case. When we think about equity and access, we need to think about how these opportunities can be created. Language was an issue for both participants but they managed to work through their issues and leverage opportunities that helped them overcome their problems with the English language. On the other hand, the symbolic power of access cannot be denied and new ways of turning that into learning opportunities need to be designed and implemented. Not every is capable of or interested in participating in open-source communities and in many ways there is criticism of open-source in terms of its lack of diversity. Therefore, for the future, it is critical to examine what lessons can be learned here that are applicable to other learning and working ecosystems. For instance, makers and making is one community where the work is more relevant to traditional engineering fields like mechanical but it is a community that has adopted many principles from open-source software development.

Conclusion

In this paper I present a case study of two software engineers. I follow their life trajectory for over two decades and highlight the complex socio-temporal nature of learning in an era of digital connectivity and digital tools. I also demonstrate the value of ecosystems such as the open-source software communities that thrive on non-traditional modes of collaboration and partnerships among its members for expertise development. The interplay between time

(lifelong) and the spaces occupied by the study participants (lifewide) resulted in them being able to learn continuously and develop and enhance their expertise.

Acknowledgements

I am grateful to the study participants for their time and for sharing their experiences with me openly. This work is partly supported by U.S. National Science Foundation Awards#EEC-1939105 and DUE-1712129. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Agnew, N.M., Ford, K.M. & Hayes, P. J. (1997), Expertise in context: Personally constructed, socially selected and reality-relevant? In Feltovich, P. J., Ford, K. M. & Hoffman, R. R. (Eds). Expertise in Context: Human and Machine. MIT/AAAI Press. pp. 219-244.

Anderson, K., Courter, S., Mcglamery, T., Nathans-Kelly, T. & Nicometo, C. (2010). Understanding engineering work and identity: A cross-case analysis of engineers within six firms. Engineering Studies, 2(3): 153-174.

Barron, B. (2010). Conceptualizing and tracing learning pathways over time and setting. NSSE Yearbook, 109(1).

Barron, B. (2004). Learning ecologies for technological fluency in a technology-rich community. Journal of Educational Computing Research, 31, 1–37.

Barron, B. (2006). Interest and self-sustained learning. Human Development, 49, 193–224.

Beach, K. D. (1999). Consequential transitions: A sociocultural expedition beyond transfer in education. Review of Research in Education, 24, 124–149.

Boland, R., Lyytinen, K., & Yoo, Y. (2007). Wakes of innovation in project networks: The Case of digital 3-D representations in architecture, engineering, and construction. Organization Science, 18(4): 631-647.

Bricker, L. A., & Bell, P. (2008). Mapping the learning pathways and processes associated with the development of expertise and learner identities. In P. A. Kirschner, J. van Merriënboer, & T. deJong (Eds.), Proceedings of the Eighth International Conference of the Learning Sciences (Vol. 3, pp. 206–211).

Brown, J. S. (2000). Growing up digital: How the web changes work, education, and the ways people learn. Change: The Magazine of Higher Learning, 32(2): 11-20.

Brown, J. & Duguid, P. (2001). Knowledge and organization: a social-practice perspective. Organization Science, Vol. 12 No. 2, pp. 198-215.

Brown, J.S. and Collins, A. and Duguid, P. (1989) Situated Cognition and the Culture of Learning. Educational Researcher, 18, 1, pg. 32-42.

Brown, S. A., Montfort, D., & Frye, N. L. (2013). What is engineering knowledge: A longitudinal study of conceptual change and epistemology of engineering students and practitioners. Proceedings of the ASEE Annual Conference and Exposition, Atlanta, GA, June 23-26.

Brunhaver, S., Korte, R., Barley, S., & Sheppard, S. (2018). Bridging the gaps between engineering education and practice. In R. Freeman & H. Salzman (Eds.), Engineering in a Global Economy (pp. 129–165). Chicago: Chicago University Press.

Clancey, W. J. (1997). The conceptual nature of knowledge, situations, and activity. In Feltovich, P. J., Ford, K. M. & Hoffman, R. R. (Eds). Expertise in Context: Human and Machine. MIT/AAAI Press. pp. 249-291.

Cole, M. (1996). Cultural psychology: A once and future discipline. Cambridge, MA: Harvard University Press.

Collins, A., Brown, J., & Newman, S. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. Knowing, learning, and instruction: Essays in honor of Robert Glaser, p. 32-42.

Cuban, L. (2001). Oversold and underused: Reforming schools through technology, 1980-2000. Harvard University, Cambridge.

Dahlgren, M. A., Hult, H., Dahlgren, L. O., Segerstad, H., & Johansson, K. (2006). From senior student to novice worker: Learning trajectories in political science, psychology and mechanical engineering. Studies in Higher Education, 31(5), 569-586.

Elder, G. (1994). Time, human agency, and social change: Perspectives on the life course. Social Psychology Quarterly, 57, 4–15.

Eraut, M. (2004). Transfer of knowledge between education and workplace settings. In H. Rainbird, A. Fuller, & A. Munro (Eds.), Workplace Learning in Context (pp. 210-221). London: Routledge.

Gee, J. P. (2000). Identity as an analytic lens for research in education. Review of Research in Education, 25, 99–125.

Geiger, R. S., & Ribes, D. (2011). Trace ethnography: Following coordination through documentary practices. In Proceedings of the 44th Hawaii International Conference on System Sciences, January 4-7.

Greeno, J. (2006). Learning in Activity. In Sawyer, K. (Ed). Cambridge Handbook of Learning Sciences (pp. 79-96). Cambridge University Press, New York, NY.

Hasu, M. (2005). In search of sensitive ethnography of change: Tracing the invisible handoffs from technology developers to users. Mind, Culture, and Activity, 12(2): 90-112.

Huff, J, Smith, J. A., Jesiek, B. K., Zoltowski, C. B., and Oakes, W. C. (2018). Identity in Engineering Adulthood: An Interpretative Phenomenological Analysis of Early-Career Engineers in the United States as They Transition to the Workplace. Journal of Emerging Adulthood.

Jesiek, B., Johri, A., Brozina, C. & Korte, R. (2020). Work-in-Progress: Novel Ethnographic Approaches for Investigating Engineering Practice. Proceedings of the American Society for Engineering Education Virtual Conference.

Jesiek, B. K., Trellinger, N. M., and Nittala, S. (2017). Closing the practice gap: Studying boundary spanning in engineering practice to inform educational practice. In Proceedings of the 2017 IEEE Frontiers in Education Conference, Indianapolis, IN, October 18-21.

Johri, A. (2018). Lifelong Learning Ecology of FLOSS: Participatory and Personalized Learning Over Time and Space. Proceedings of ACM OpenSym Conference on Open Collaboration 2018.

Johri, A. (2015a). Impressions in Action: The Socially Situated Construction of Expertise Impressions in the Workplace. Journal of Organizational Ethnography, 4(1): 44-63.

Johri, A. (2015b). Supporting Global Virtual Work through Blogs and Micro-Blogging. Proceedings of HICSS 2015, pg. 422-431.

Johri, A. (2014). Engineering Knowing in the Digital Workplace: Aligning Sociality and Materiality in Practice. In T. Fenwick & M Nerland (eds), Reconceptualising Professional Learning: Sociomaterial Knowledges, Practices, and Responsibilities. London: Routledge.

Johri, A. (2012a). From a Distance: Impression Formation and Impression Accuracy among Geographically Distributed Coworkers. Computers in Human Behavior, Vol. 28(6):1997-2006.

Johri, A. (2012b). Learning to Demo: The Sociomateriality of Newcomer Participation in Engineering Research Practices. Engineering Studies, Vol.4, Issue 3, pp. 249-269.

Johri, A. (2011). Look Ma, No Email! Blogs and IRC as Primary and Preferred Communication Tools in a Distributed Firm. Proceedings of the ACM Conference on Computer Supported Cooperative Work, pg. 305-308.

Johri, A. (2008). Boundary Spanning Knowledge Broker: An Emerging Role in Global Engineering Firms. In Proceedings of 38th ASEE/IEEE Frontiers in Education Conference, Saratoga Springs, NY. pp. S2E-7-S2E-12.

Johri, A., Olds, B. M. & O'Connor, K. (2014). Situative Frameworks for Engineering Learning Research, In Johri, A. & Olds, B. M. (Eds), Cambridge Handbook of Engineering Education Research, Cambridge University Press, New York, NY, pg. 47-66.

Johri, A., & Olds, B. (2011). Situated engineering learning: Bridging engineering education research. Journal of Engineering Education, 100(1), 151–185.

Korte, R. (2018). Learning to practice engineering in business: The experiences of newly hired engineers beginning new jobs. In S. Christensen et al. (Eds.), The Engineering-Business Nexus: Symbiosis, Tension and Co-evolution (pp. 341-361). Springer.

Korte, R. (2009). How newcomers learn the social norms of an organization: A case study of the socialization of newly hired engineers. Human Resource Development Quarterly, 20(3): 285-306.

Kozinets, R. (2006). Netnography. Handbook of qualitative research methods in marketing, pg. 129-142, Edward Elgar Cheltenham, England.

Kozinets, R. (2002). The field behind the screen: Using netnography for marketing research in online communities. Journal of marketing research, 39(1):61-72.

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

Lee, D. M. S. (1986). Intellectual, motivational, and interpersonal qualities and NOT academic performance predicted performance on first job. IEEE Transactions on Engineering Management, EM33(3), 127-133.

Lemke, J. L. (2001) The Long and the Short of It: Comments on Multiple Timescale Studies of Human Activity, Journal of the Learning Sciences, 10:1-2, 17-26,

Lemke, J. L. (2000). Across the scales of time: Artifacts, activities, and meanings in ecosocial systems. Mind, culture, and activity, 7(4), 273-290.

Marcus, G. E. (1995). Ethnography in/of the world system: The emergence of multi-sited ethnography. Annual review of anthropology, 24(1): 95-117.

Nasir, N. S. (2002). Identity, goals, and learning: Mathematics in cultural practice. Mathematical Thinking and Learning, 4, 213–248.

Nicolini, D., Gherardi, S. and Yanow, D. (2003). Knowing in organizations: A practice-based approach. New York: M. E. Sharpe.

Roth, W. M. (2014) Reading Activity, Consciousness, Personality Dialectically: Cultural-Historical Activity Theory and the Centrality of Society. Mind, Culture, and Activity, 21:1, 4-20.

Roth, W. M. (2001) Situating Cognition, The Journal of the Learning Sciences, 10:1-2, 27-61.

Saxe, G. B., & Esmonde, I. (2005). Studying cognition in flux: A historical treatment of "Fu" in the shifting structure of Oksapmin mathematics. Mind, Culture, and Activity, 12, 171–225.

Schatzki, T. (2001). Introduction: Practice theory. In T. R. Schatzki, K. K. Cetina, & E. V. Savigny (Eds.), The Practice Turn in Contemporary Theory (pp. 42-55). New York: Routledge.

Serpell, R., & Hatano, G. (1997). Education, literacy and schooling in cross-cultural perspective. In J. W. Berry, P. R. Dasen, & T. M. Saraswathi (Eds.), Handbook of cross-cultural psychology (Vol. 2, 2nd ed., pp. 345–382). Boston: Allyn and Bacon.

Siemens, G. (2005). Connectivism: A learning theory for the digital age. International journal of instructional technology and distance learning, 2(1), 3-10.

Stevens, R., Johri, A. & O'Connor, K. (2014). Professional Engineering Work. Johri, A. & Olds, B. (Eds). The Cambridge Handbook of Engineering Education Research, Cambridge University Press, New York, NY.

Stevens, R., O'Connor, K., Garrison, L., Jocuns, A. & Amos, D. (2008). Becoming an engineer: Toward a three dimensional view of engineering learning. Journal of Engineering Education, 97(3): 355-368.

Trevelyan, J. (2019) Transitioning to engineering practice. European Journal of Engineering Education, 44:6, 821-837.

Trevelyan, J. (2010). Reconstructing engineering from practice. Engineering Studies, 2(3), 175-195.

Trevelyan, J. (2007). Technical coordination in engineering practice. Journal of Engineering Education, 96(3), 191–204.

Von Hippel, Eric (2001). Learning from open-source software.MIT Sloan management review, 42(4):82-86.

Weller, M. (2007). The distance from isolation: Why communities are the logical conclusion in e-learning. Computers & Education, 49(2), 148-159.

Wenger, E. (1998). Communities of Practice: Learning, Meaning, and Identity. Cambridge: Cambridge University Press.

Yoo, Y., Lyytinen, K. J., Boland, R. J., & Berente, N. (2010). The next wave of digital innovation: Opportunities and challenges: A Report on the Research Workshop 'Digital Challenges in Innovation Research' (June 8, 2010). Available at http://ssrn.com/abstract=1622170

Zussman, R. (1985). Mechanics of the middle class: Work and politics among American engineers. Berkeley, CA: University of California Press.