Article

Work and Occupations
2017, Vol. 44(4) 376–423

DOI: 10.1177/0730888417726119
journals.sagepub.com/home/wox

Working Algorithms: Software Automation and the Future of Work

Benjamin Shestakofsky[1]

Abstract

While some argue that the rise of software automation threatens workers with obsolescence, others assert that new complementarities between humans and software systems are likely to emerge. This study draws on 19 months of participant-observation research at a software firm to investigate how relations between workers and technology evolved over three phases of the company's development. The author finds two forms of human–software complementarity: computational labor that supports or stands in for software algorithms and emotional labor aimed at helping users adapt to software systems. Instead of perfecting software algorithms that would progressively push people out of the production process, managers continually reconfigured assemblages of software and human helpers, developing new forms of organization with a dynamic relation to technology. The findings suggest how the dynamism of the organizations in which software algorithms are produced and implemented will contribute to labor's enduring relevance in the digital age.

Keywords

technology, algorithms, automation, computational labor, emotional labor, organizational change

[1]University of California, Berkeley, CA, USA

Corresponding Author:
Benjamin Shestakofsky, University of California, 410 Barrows Hall, Berkeley, CA 94720, USA.
Email: bshestakofsky@berkeley.edu

How is software automation transforming work and employment? In light of rapid advancements in artificial intelligence (AI), big data, and machine learning, many social scientists and information technologists predict that in diverse sectors of the economy, "smart" machines will increasingly replace the human element of production, leaving mass joblessness in their wake (Benzell, Kotlikoff, LaGarda, & Sachs, 2015; Brynjolfsson & McAfee, 2014; Elliott, 2014; Ford, 2015; Frey & Osborne, 2017; Kaplan, 2015; Sachs & Kotlikoff, 2012; Srnicek & Williams, 2015). Others dispute the technological unemployment thesis, arguing that although software technologies will supplant some workers, they will also give rise to new and previously unforeseen complementarities between human labor and digital systems (Arntz, Gregory, & Zierahn, 2016; Autor, 2015a, 2015b; Bessen, 2015a, 2015b; Davenport & Kirby, 2016; Ekbia & Nardi, 2017; Howcroft & Taylor, 2014; Mindell, 2015; Mokyr, Vickers, & Ziebarth, 2015). Existing studies attempt to project employment trends by marshaling statistical models (Arntz et al., 2016; Autor, Levy, & Murnane, 2003; Benzell et al., 2015; Frey & Osborne, 2017; Sachs & Kotlikoff, 2012), historical comparison and deductive logic (Autor, 2015a; Bessen, 2015b; Mokyr et al., 2015), and evidence aggregated from prior research, news reports, conversations with scientists, and in-person demonstrations of cutting-edge technologies (Brynjolfsson & McAfee, 2014; Davenport & Kirby, 2016; Ekbia & Nardi, 2017; Ford, 2015; Kaplan, 2015). Despite the recent influx of interest, however, we still know surprisingly little about the conditions under which software systems function autonomously—and when they rely on the assistance of complementary human workers—in real-world settings.

This study contributes to contemporary debates around the future of work by examining the processes through which assemblages of software and workers coevolve within a particular organizational setting. I draw on 19 months of participant-observation research at a San Francisco-based tech start-up called AllDone, which aimed to transform local service markets by using technology to more efficiently connect buyers and sellers of services ranging from house cleaning to wedding photography to tutoring and beyond.[1] I demonstrate the uneven development of relations between software and the people who create and use it. Each phase of AllDone's development revealed mismatches between humans and machines generated by the firm's shifting strategic imperatives. To address these lags, AllDone relied on two forms of complementary labor performed by a distributed, online workforce located across the Philippines and the Las Vegas area. The results suggest how the dynamism of the organizations in which software algorithms are produced and implemented will contribute to labor's enduring relevance in the digital age.

In the following section, I evaluate the contemporary debate surrounding software automation and the end of work in light of prior research on the relationship between work and technology. I argue that existing studies of software automation neglect to examine organizational dynamics that are likely to contribute to the emergence, reproduction, and transformation of human–software configurations. After introducing the research site, I present evidence from three phases of AllDone's development to demonstrate how a high-tech firm's assemblages of workers and machines unfolded over time as managers confronted problems within a shifting organizational context. I conclude by discussing how the internal dynamism of firms can drive ongoing innovation in human–machine assemblages rather than progressive automation.

Work and Technology in Historical Perspective

The history of industrial capitalism has been punctuated by waves of anxiety surrounding technological change, with scholars and critics repeatedly asking whether the evolution of production technologies could progressively reduce aggregate demand for labor (Brick, 2000; Keynes, 1963; Simon, 1965; Winner, 1977). In spite of monumental scientific achievements, such predictions have not yet come to pass. While in some cases new technologies have *substituted* for workers, they have also consistently created new *complementarities* between people and machines (Autor, 2015a).

Economists and economic historians commonly cite the historical recurrence of three processes through which technological innovation can produce jobs. First, when technology replaces some human activities, it may also increase demand for other types of work. For example, during the industrial revolution, mechanics tended to new machines, while supervisors, accountants, and administrators emerged to manage large-scale enterprises. Substitution in one industry may also be counterbalanced by new complementarities in others: In the early 20th century, automation caused massive job loss in the U.S. agricultural sector that would eventually be offset by gains in manufacturing and services (Autor, 2015a). Second, product innovation can create new sources of consumer demand, giving rise to new occupations and even sectors of the economy (Mokyr et al., 2015). For instance, people are employed to build and maintain hardware, software, and telecommunications networks using Internet and mobile technologies that did not begin to achieve widespread distribution until the late 20th century. Third, automation that reduces the cost of labor can result in cheaper goods, which can increase consumer demand and in turn boost demand for labor. When power looms automated an estimated 98% of the labor required to weave a yard of cloth in the 19th century, consumer demand increased so sharply that the number of weaving jobs actually rose, and workers' remaining tasks became increasingly valuable (Bessen, 2015b).

Close examinations of production technologies in a variety of industries have supported theories of complementarity by demonstrating that, rather than fully replacing human workers, automation may instead create new and sometimes unanticipated interactions between humans and mechanized systems. The high cost, limited capacities, fragility, or rigidity of automated equipment often leads employers to search for human helpers. During the industrial revolution, the introduction of innovative technologies to a variety of manufacturing processes intensified the physical labor required for production in and around new, fast-paced machines (Samuel, 1977). In chemical plants, the "scientific work" performed in a modern control room characteristic of continuous production methods can go hand in hand with the "donkey work" of shoveling fertilizer, lifting and carrying massive bags, driving trucks, and maintaining machinery amid heat, thick dust, and noise (Nichols & Beynon, 1977). Plant owners who invested in automated machine tools found that "[i]n reality, [numerical control] machines do not run themselves... the new equipment, like the old, requires a spectrum of manual intervention and careful attention to detail" (Noble, 1999, p. 172). Automation and computerization can also spur the development of new skills by transforming the nature of work tasks (Walker, 1958; Zuboff, 1988).

Technological innovations can replace work, transform it, or create new jobs; technology can also lead to de-skilling or up-skilling and at times will have no discernable effect on work. What, then, determines how any given technological advancement will affect work and employment? Rejecting a technological determinism according to which the outcomes of innovation are predetermined by scientific progress or rational considerations of efficiency, science and technology scholars have demonstrated that the relationship between work and technology is the product of a constellation of social relations (MacKenzie & Wajcman, 1999). The effects of technology on work are thus inseparable from the social settings—more specifically, the organizational contexts—in which they interact (Barley, 1986; Barley & Kunda, 2001;

Liker, Haddad, & Karlin, 1999; Wajcman, 2006). The existence of a technological object alone does not dictate if or how an organization will use it; power relations between workers and managers, managerial philosophy, and the processes through which new technologies are developed and implemented are just some of the factors that can produce particular, contingent sociotechnical systems (Liker et al., 1999). Barley's (1986) study of the introduction of computed tomography scanners in two hospital radiology departments is exemplary in this regard, demonstrating that the same technological innovation affected work roles and practices differently depending upon the distribution of technical expertise within each department. The effects of technology on work are nonlinear and complex; understanding the conditions that give rise to variable configurations of workers and software requires examination of the organizations in which they are embedded.

Is This Time Different?

At the end of the 20th century, some social scientists and critics were again cautioning that the historical balance between substitution and complementarity would soon break down. This time, it was advancements in information and communications technologies that heralded a coming crisis of widespread unemployment (Aronowitz & DiFazio, 1994; Noble, 1995; Rifkin, 1995). In recent years, such warnings have taken on renewed urgency amid the emergence of ''big data'' and ''machine learning'' techniques in software development that may help programmers overcome longstanding barriers to automating tasks.

Until recently, computers were largely limited to performing ''codified, repetitive information-processing tasks'' (Autor, 2015b, p. 247). Programmers developed software capable of simulating human cognition by instructing it to ''follow precise, well-understood procedures''—for example, ''the mathematical calculations involved in simple bookkeeping [or] the retrieving, sorting, and storing of structured information typical of clerical work'' (Autor, 2015a, p. 11). However, there remain many tasks that cannot be simulated in this manner because humans cannot articulate the exact ''rules'' necessary for accomplishing them. Following philosopher Michael Polanyi's (1966) statement that ''[w]e know more than we can tell,'' Autor (2014) observes that ''the tasks that have proved most vexing to automate are those demanding flexibility, judgment and common sense—skills that we understand only tacitly'' (p. 136). For instance, most untrained human children could visually identify whether or not an object is a chair with a high degree of accuracy by reasoning about what the object is "for." It is far more difficult, however, to program a computer to consistently succeed at the same task because it is difficult to define a specific set of attributes that a chair will possess: Some have four legs, while some have none; some have a back, arms, or wheels, while others do not (Autor, 2015a).

Rather than telling software precisely how to perform certain operations, machine learning allows computers to infer patterns approximating tacit rules from large sets of "training" data (Alpaydin, 2014; Pratt, 2015). Imagine again the task of visually identifying a chair. Software engineers could "feed" their program a vast number of human-completed operations (i.e., thousands of images marked "chair" or "not chair"), and the software would use statistical modeling akin to inductive logic, rather than deductive reasoning, to "learn" how to perform the operation (Autor, 2015a).

Contemporary *discontinuity theorists* argue that breakthroughs in machine learning and AI signal an inflection point in the historical link between technological innovation and job creation (Brynjolfsson & McAfee, 2014; Elliott, 2014; Frey & Osborne, 2017; Kaplan, 2015; Sachs & Kotlikoff, 2012; Srnicek & Williams, 2015). Machine learning can teach computers to perform tasks across a variety of occupational domains that previously required tacit knowledge and complex human cognition. In some settings, computers can make more accurate and less biased judgments than well-trained humans, process language, and even detect human emotions and facial expressions (Tufecki, 2015). Discontinuity theorists cite the rapid development of self-driving cars; speech recognition software capable of understanding complex communication (like the technology behind the iPhone's Siri); increasingly accurate algorithmic language translation services; the Watson computer that in 2011 beat *Jeopardy!* champions at their own game; and continued advances in factory robotics (Brynjolfsson & McAfee, 2014; Ford, 2015). Indeed, AI is already substituting for the work of journalists (Smith, 2015), human resources managers (Miller, 2015), medical lab technicians (Tufecki, 2015), and equities traders (Popper, 2016). In light of these trends, some social scientists and information technologists warn that societies must prepare for a future in which employment opportunities will continue to diminish as "smart" machines encroach on an increasing percentage of job tasks. According to one widely cited estimate, 47% of U.S. employment is at risk of computerization within the next 10 to 20 years (Frey & Osborne, 2017).

Continuity theorists, on the other hand, predict that the complex historical dynamics of human–machine interaction will continue to hold (Arntz et al., 2016; Autor, 2015a, 2015b; Bessen, 2015a, 2015b; Davenport & Kirby, 2016; Ekbia & Nardi, 2017; Mindell, 2015; Mokyr et al., 2015). Economist David Autor (2015a, 2015b) has produced the most thorough and convincing case for continuity, emphasizing that those interested in the future of work must look beyond substitution to identify the limitations of computerization and the complex interactions between technology and employment.

Although in the popular imagination software algorithms often take on mythic qualities of near-omnipotence (Ziewitz, 2016), in reality, developers continue to grapple with their limitations. Continuity theorists remain skeptical that computers can provide an adequate substitute for tacit knowledge in many settings. Although on average machine learning may be capable of providing accurate output in a variety of applications, machine-learning algorithms often fail in ways that are difficult to predict. For example, whereas discontinuity theorists cite the fact that a computer defeated human *Jeopardy!* champions as evidence of computing's swift progress, Autor emphasizes how the computer got one question spectacularly wrong.[2] Although machine learning can offer "*good and useful approximation*[*s*]" of mental processes (Alpaydin, 2014, p. 2), results often remain inconsistent and incomplete. Furthermore, it can be difficult, if not impossible, to reverse engineer machine-learning algorithms to explain the reasoning underlying their decisions (Burrell, 2016).

Scholars have also called into question the purported autonomy of AI by emphasizing the essential role of human activity in the implementation of AI systems (Ekbia & Nardi, 2017; Goldberg, 2015; Mindell, 2015). (See Appendix A for further details on debates surrounding the nature of AI.) For example, many observers seem unaware that machine-learning algorithms are often powered by human labor—not only of the software engineers who design them but also of workers who create vast amounts of training data to teach them (Irani, 2015c), which in some cases must be frequently updated to adapt to changing environments. In addition to the aforementioned limitations of machine learning, some cite the high cost of building and maintaining digital machinery and the difficulty of programming it to adjust to new and unpredictable conditions as reasons why many employers will continue to rely on humans to perform nonroutine work (Autor, 2015a, 2015b; Ekbia & Nardi, 2017). Many tasks requiring flexibility, situational adaptability, creativity, judgment, intuition, spoken language, interpersonal interaction, and persuasion are unlikely to be fully automated anytime soon.

Continuity theorists' argument that the rise of "smart" machines does not signal the end of work holds not only that many work tasks will not be automated but also that, as in the past, technological change will increase demand for labor that complements new machinery. Yet, it remains difficult to predict the shape of new human–machine configurations: While one can regularly find news stories about technologies that threaten to substitute for human labor, "[t]he offsetting effects of complementarities and rising demand in other areas are, however, far harder to identify as they occur" (Autor, 2015a, p. 26). Economists have sought to deduce future employment trends from theory and historical comparison; however, their approaches are pitched at a level of abstraction that precludes systematic investigation of continuity's form and dynamics within concrete social settings. With their methodological pluralism, sociologists and organization scholars are uniquely positioned to offer ethnographic insights capable of advancing debates around the future of work. Organizational ethnographers influenced by the perspectives of social studies of science and technology have been "especially attuned to the contradictions, contingencies, and nuances of technological development" (Wajcman, 2006, p. 782). Only by examining organizations in which software algorithms are being developed and implemented can researchers uncover the nature of human–software complementarities that are arising, the contextual factors that influence their form, and how these configurations change over time.

This study examines how the relationship between work and technology evolved during three phases of a start-up company's development. During the first phase of my research, AllDone's executives hoped to take advantage of the company's first round of venture capital funding by increasing demand for the product. Software engineers prioritized building and expanding systems that drew new users to the website and facilitated their activity. Developers confronted *machine lag*, or gulfs between their imagination and the realities of technology's limitations and the firm's scarce resources. They addressed machine lag by deploying *computational labor* in the form of Philippine workers who performed repetitive information-processing tasks to complement software infrastructure. During the second period of research, managers shifted their focus toward generating a sustainable revenue stream that would ensure the company's longevity and attract a second round of funding. *Human lag* arose when users were reluctant to accept the performance of AllDone's software systems. Human lag was addressed by the *emotional labor* of phone support workers distributed across the Las Vegas area who managed relationships with users and helped them adjust to AllDone's changing product and policies. During the final phase of research, AllDone secured a second round of VC funding and began to combine its previous goals of expansion and revenue generation. The dual goals of growing the customer base while simultaneously squeezing more money out of their activities created tensions that systematically reproduced machine lag and human lag. In response to this *permanent lag*, AllDone continued to expand the ranks of complementary workers whose efforts supported technological systems.

Research Setting

This study draws on data gathered during 19 months of participant-observation research conducted at AllDone between February 2012 and August 2013. (See Appendix B for details on data-gathering and analytic procedures.) The organization comprised three work teams. For the majority of my tenure with the company, the San Francisco office (known as AllDone San Francisco, or ADSF) was home to about 20 full-time employees in engineering, design, marketing, business, and operations divisions. A distributed, work-from-home team of 200 people across the Philippines (AllDone Philippines, or ADP) typically handled routinized data-processing tasks. And a distributed, work-from-home team of 10 in the Las Vegas area (AllDone Las Vegas, or ADLV) interfaced with AllDone users via telephone.[3] Members of the two remote teams generally held full-time, open-ended independent contractor positions. ADP contractors were recruited, hired, and supervised via oDesk, a platform designed to facilitate online freelancing projects. ADLV workers were recruited using local job postings on Craigslist and then supervised via oDesk.

AllDone had launched its nationwide, online marketplace in early 2010. The company was one of many aiming to build "Amazon for local services," a website that would eventually make it as easy to find and hire providers of local services online as it is to buy products. More than 600 service categories were represented, ranging from home improvement (e.g., plumbers and electricians), to event services (e.g., DJs and caterers), to guitar teachers, locksmiths, and many others. Buyers who visited AllDone from a computer or mobile device were presented with a text box in which they could enter the type of service they were looking for. Buyers would then fill out a short form, answering three or more questions about the details of the job. Completed buyer requests were distributed via e-mail or text message to sellers (service providers) in the buyer's area who might be capable of performing the job. Those who were available and interested could pay AllDone a fee to send a quote to the buyer including a price estimate, information about what was included in their price, and a pitch describing why they were qualified for the job. Buyers and sellers managed any subsequent communication, provision of service, and payment; AllDone assumed no formal responsibility for the outcome of market activities.

High-tech start-ups often look to venture capital (VC) firms to supply needed funds. VCs receive an equity stake in emerging enterprises in exchange for capital that entrepreneurs can use to fund operations and growth, along with the VC's counsel, credibility, and connections. Venture-backed start-ups are funded in stages or rounds, with early investors typically receiving the most favorable terms, having invested when the company's prospects were more difficult to evaluate. VCs derive returns from the appreciation of the value of the firms in which they invest. Rather than funding companies that aim to steadily build reliably profitable businesses, VCs develop portfolios of high-risk, high-reward firms, seeking exponential returns via the sale of their investments in corporate acquisitions or initial public stock offerings (Gompers & Lerner, 2004; Kenney & Florida, 2000; Zider, 1998).

Like most start-up companies, AllDone developed amid conditions of uncertainty and fluidity. Nascent firms in unsettled markets typically struggle to secure scarce resources including customers, capital, and employees. Unable to make substantial investments in research or long-term strategic planning, entrepreneurial firms are frequently presented with unanticipated challenges and opportunities. Entrepreneurs thus pursue a strategy of "opportunistic adaptation," developing ad hoc responses to unexpected problems as they arise and experimenting with a multitude of new initiatives whose outcomes are uncertain (Bhidé, 2000). AllDone's executives frequently shifted the company's strategic direction as they sought to convince successive rounds of VCs to provide the resources the company needed to supercharge its growth.

As a successful high-tech start-up that would eventually be valued at more than $1 billion, AllDone provided an ideal setting in which to examine the relationship between work and technology in the digital age. The company was part of a new wave of firms using software to "disrupt" traditional local service markets. Like Uber, Airbnb, and a host of other digital platform providers, AllDone aggregated a vast array of local markets into one online clearinghouse. Such online marketplaces are designed to make it easier for buyers and sellers to locate and transact with one another while also providing an opportunity for platform providers to extract fees from users and profit from their brokerage positions.

AllDone experienced rapid growth during the course of my fieldwork. By June 2013, more than 250,000 sellers had signed up with

AllDone. During that month alone, 25,000 submitted at least one quote to a consumer. AllDone was acquiring 4,000 new sellers per month, and sending sellers more than 100,000 consumer requests, representing a 200% increase in request volume in just 5 months, and a 400% increase over the previous year.

After leaving the field and reviewing my fieldnotes, I identified three analytic phases, each corresponding to a roughly 6-month period of my research and the company's development. The construction of each analytic phase was based on my observations of major "breakpoints," when exogenous events and shifts in organizational strategy offered new "occasions for structuring" the relationship between work and technology (Barley, 1986). In the empirical sections that follow, I trace three distinct configurations of software and labor at AllDone. The first phase was a period of expansion in which the firm grew its team in the Philippines to perform computational labor that complemented software systems designed to attract new users to the website and facilitate market activity. In the second period, when revenue generation became of paramount importance, the company relied on emotional labor provided by phone support agents in Las Vegas to build, repair, and preserve users' trust in AllDone. With its new business model in place, AllDone then raised a second round of VC funding. In this third phase of development, the company combined its prior goals of expansion and revenue generation. As the ranks of software engineers swelled, their increased pace of innovation necessitated the continual intervention of complementary workers to bring technology up to speed with developers' vision and to help users adapt to changing systems.

Phase I: From Machine Lag to Computational Labor

> We know what the future of local services is [...]. [...] But we're not the only people that know this is the future. And, more importantly, there's lots of people—smart, scrappy, and well-funded people—building our vision. [...] Someone is going to do it. And it looks like it's going to happen soon. [...] We just have to finish building faster than anyone else and we will win. We have to. (Carter, AllDone's President, in an e-mail to San Francisco staffers)

Executives planned to take advantage of AllDone's first round of VC funding by prioritizing the expansion of the company's staff and user base. Only by growing quickly could AllDone hope to secure the advantages that accrue to "first movers" in an industry that continued to welcome an ever-expanding array of competitors. Enlarging the four-person engineering staff would significantly increase AllDone's pace of innovation, enhancing the company's ability to devise plans, run experiments, assess results, and implement changes to the product.

AllDone's software engineers were heavily involved in selecting their new colleagues, sacrificing their productivity to build the team. Reflecting on the first quarter of 2012, Josh, ADSF's product manager, reported that AllDone's engineers had "accomplished very little" in terms of their production goals because they had been "very, very focused on recruiting" activities that reportedly consumed at least half of their work time.

The push for rapid growth created *machine lag*, as software developers' needs and imagination frequently outstripped the capabilities of technology and available engineering resources. To address machine lag, AllDone expanded its digital assembly line in the Philippines, where workers performed *computational labor* to complement software systems. I use the term *computation* to refer to "the transformation of information by means of an algorithm or program" that "defines rules by which information will change" (Gershenson, 2013, p. 62; cf. Ekbia & Nardi, 2017; Irani, 2015a; Michelucci, 2013). Humans perform computational labor when they implement information-processing algorithms. Computational labor allowed AllDone to accomplish its goal of spurring substantial user growth in spite of the impediments to software production imposed by recruiting activities. During the first quarter of 2012, AllDone received almost 50% more consumer requests than it had during the last 3 months of 2011; during the second quarter, that mark increased again by 75%.

Computational labor complemented software systems in three ways. In some instances, workers' tacit skills gave them an advantage over computer code in performing *nonroutine tasks*. In other cases, the company relied upon *reverse substitution*, using workers to imitate software algorithms that otherwise would have been too costly or time-consuming to produce. In addition, some ADP staffers provided *workarounds*, performing routine tasks to subvert systems designed to detect software automation.

Nonroutine Tasks

In some instances, AllDone complemented software systems with human workers because people possessed competencies grounded in tacit knowledge that could not easily be programmed. For example, ADSF turned to Filipino workers to undertake a marketing project that software alone was ill-suited to handle. AllDone focused its buyer acquisition efforts on ‘‘search engine optimization’’ (SEO). SEO describes a set of techniques designed to bump a website’s pages to the top of search engine results. One way that websites can enhance their standing in search engine results is by obtaining incoming links from other websites—particularly from websites that search engine companies believe are widely respected by Web users. In early 2012, ADSF conducted a survey of AllDone sellers about the local business environments in which they operated. The company then packaged the survey results on its website and tried to get other websites to link to those new pages. AllDone paid a public relations firm $30,000 to assist with outreach to news organizations.

Simultaneously, two ADSF managers orchestrated a ‘‘data mining’’ experiment to find out if a combination of technology and workers could outperform the professional PR firm. Managers in San Francisco asked Christine, a team leader in the Philippines, to recruit two dozen workers to join a temporary survey team. The ADSF managers created detailed documents instructing members of this new team on how to systematically scour the Web for the first and last name, e-mail address, Twitter handle, and organizational affiliation of people and platforms in the United States that might publish a story about local business issues. It would have been exceedingly difficult to teach software to accurately gather unstructured data from such a vast array of sources, each of which was formatted in a unique manner. AllDone’s human workers, however, possessed a tacit understanding of how to identify the desired information and could perform these operations with relative ease.

Team members spent weeks accumulating and classifying data about 50,000 journalists, bloggers, nonprofits, politicians, and think tanks, recording entries into a complex series of spreadsheets. I then wrote detailed instructions for ADP survey team members to ‘‘clean’’ and standardize each entry to ensure that an ADSF manager could use an automated system to send each target a ‘‘personalized’’ e-mail pitch. ADP’s survey team then recorded recipients’ responses to ADSF’s e-mails in the spreadsheets. Finally, I wrote instructions specifying under what conditions survey team members should follow-up via e-mail or Twitter with contacts who had not yet responded. The experimental survey team was wildly successful, yielding hundreds of incoming links—50 times as many stories as had the PR firm that AllDone had retained for the same purpose, at one third of the cost.[4]

The survey project introduced me to what members of ADSF called "establishing process," or routinizing knowledge work to be offshored to a mass of flexible workers distributed across the Philippines. If ADSF's software engineers wrote code in specialized programming languages to guide the central processing units and servers powering the company's technological infrastructure, I had become a process engineer writing code in plain English to guide the people who constituted what managers often called AllDone's "human machine." Like the software engineers, I too was crafting algorithms, "sequence[s] of instructions that should be carried out to transform [any given] input to output" (Alpaydin, 2014, p. 2). And like computer code, the instructions I wrote were designed to leave nothing to the imagination, sometimes including graphical decision trees to help workers understand how to handle all possible contingencies (e.g., Figure 1). With every operation explicitly detailed, workers became nearly as interchangeable as central processing units in a network, with each person developing approximately the same interpretation of each task. My algorithms were "debugged" by managers in the Philippines who reviewed the tasks I assigned and asked me questions to help me clarify the instructions before they were distributed to team members for human computation.

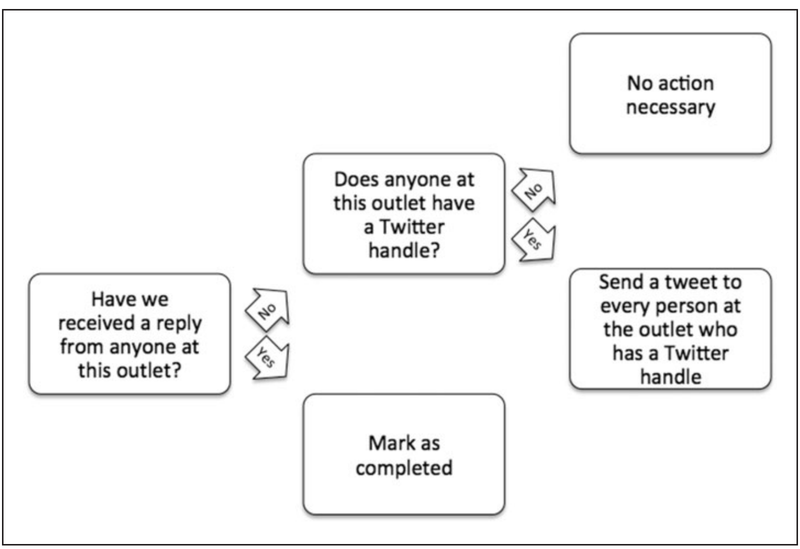

Figure 1. Example of algorithmic decision tree.

ADP managers would also field questions from team members as they arose, reducing the number of queries that made it to San Francisco.

Recognizing the value of ADP's functional and numerical flexibility, members of ADSF began to imagine new tasks requiring tacit knowledge that could be completed by human workers. When the survey project was completed, the team that had powered it was disbanded as easily (from the perspective of ADSF managers) as it had been summoned. However, the experiment proved so powerful that other members of ADSF were soon asking me to devise algorithmic instructions for their own "special projects" to take advantage of ADP's human computation. For example, I established a process through which members of ADP would promote job openings at AllDone by gathering contact information for and reaching out to college computer science programs. I initially offered such projects on an ad hoc basis to ADP team members who wanted more work, but later Christine was asked to reconstitute her group as a permanent special projects team. ADSF staffers could call upon this group's efforts whenever they wanted to quickly and cheaply execute a "random" data-gathering or -processing task without using up (or waiting for) scarce technological resources. Just as software engineers often rely on "software-as-a-service"—integrating other companies' ready-made software products into their own code—all of ADSF could access ADP's "humans-as-a-service" (Irani, 2015a), using a flexible, on-demand workforce to pick up or drop projects at a moment's notice. Managers encouraged all ADSF employees—from software engineers to the office manager—to "outsource" as many tasks as possible. As my supervisor wrote in my initial performance review:

> [I]f you spend all your time on grinder projects that take many hours, we lose you for other projects that could be equally valuable for you to work on. You should always think about how you can set up a process, delegate to someone else (an outside contractor or someone on ADP/ADLV), and move on.

Offloading "grinder projects" to ADP had two benefits for ADSF. First, "outsourcing" freed more costly workers from routine tasks, allowing them to spend more time on more "valuable" work. Second, ADP altered ADSF employees' vision of technology's horizons, expanding the company's scope of action by unlocking innovative schemes (like the survey project) that could be completed quickly, cheaply, and effectively only by an agglomeration of human workers.[5]

Reverse Substitution

Conceptualizing, developing, testing, and refining complex software algorithms is a costly and time-consuming process. Even when software could technically be crafted to handle a task, AllDone's small team of software engineers often believed that the resources required to code and implement it would be prohibitive or could be better allocated elsewhere. In these instances, developers relied upon *reverse substitution*, stripping routine computation work from computers and displacing it onto workers. These workers stood in for software algorithms not because full automation was impossible but because developers believed that achieving it would be inefficient. Although machine learning uses computational "brute force" to simulate human cognition, reverse substitution replaces software algorithms with the "brute force" of large-scale, routinized labor processes.

Because AllDone's SEO strategy was yielding an ever-increasing volume of buyer requests, the company had to connect far more buyers with sellers. Indeed, AllDone's core function as a broker was to connect potential buyers with sellers of local services. However, AllDone's developers chose not to develop software algorithms to perfect this process. Instead of devoting scarce engineering resources to matching, AllDone maintained a workforce in the Philippines to manually construct every introduction.

Members of ADP's matching team used a Web portal that displayed the details of each buyer request. They began by vetting requests and deleting those that appeared to be fraudulent (e.g., a request placed by "Mickey Mouse"). Team members were then provided with a list of local AllDone sellers who might be eligible to receive the request because they worked in related service categories. Workers would select all of the sellers who they judged to be an appropriate match, and the sellers would then be automatically notified via e-mail or text message of the incoming request. AllDone's users would never know that human workers, rather than a computer algorithm, had handcrafted each introduction. To keep up with the rapid rise in request volume during the first half of 2012, the matching team's managers hired 38 new members, more than doubling the team's size from 30 to 68.

If in some instances software engineers used reverse substitution to manage long-term organizational processes, in others reverse substitution was deployed to reduce the engineering burden that came with developing new and experimental product features. In one such case, members of ADP were used to test whether it would be worth an engineer's time to "code up" a new product feature. ADSF's product team wanted to determine whether providing a link to a seller's Web page on Yelp (a consumer review website) would increase a buyer's likelihood of contacting the seller. Yelp offers developers tools that allow them to embed Yelp users' business information directly into their own websites. However, Bill, the engineer in charge of the project, preferred not to spend his time learning how to use Yelp's tools without first knowing whether or not the new feature was likely to succeed. He devised a test whereby members of ADP substituted for software algorithms by manually searching for 9,000 sellers on Yelp and gathering information from their user profiles. Bill experimented with putting some of this information on relevant AllDone pages, and, upon finding that that it did not have a statistically significant effect on buyer behavior, abandoned the test. By using ADP workers to stand in for software infrastructure, Bill was able to save valuable engineering time that otherwise would have been wasted learning how to use development tools to conduct a test that was destined to fail. ADP workers enacting reverse substitution often supported what software engineers referred to as "quick and dirty" tests. By manually executing algorithmic tasks, they provided proof of concept before developers invested time and resources in devising elegant software solutions.[6]

Computers are generally expected to replace humans in "codified, repetitive information-processing tasks" (Autor, 2015b, p. 248). At AllDone, however, resource constraints or the experimental nature of new product features could lead to the reverse outcome, with humans performing programmable digital processes by hand. The existence of software solutions did not necessarily result in their deployment at AllDone; instead, the direction of substitution depended upon the organizational context in which it was implemented.

Workarounds

Users do not always passively accept technological innovations; instead, they often develop interpretations of and adaptive responses to technology—also known as "workarounds"—that can undermine designers' intentions (Lee, Kusbit, Metsky, & Dabbish, 2015; Lim, Dey, & Avrahami, 2009; Yang & Newman, 2013; Zuboff, 1988).[7] At AllDone, workarounds combined elements of reverse substitution and nonroutine tasks. Like reverse substitution, the efforts of employees providing workarounds stood in for software algorithms that technically could have been created to perform a task. In these cases, routinized human labor possessed a comparative advantage over computer code because workers' performance of tasks using tacit knowledge allowed them to subvert a system designed to detect and prevent software automation that followed precise rules.

ADP's writing team helped AllDone work around the systems that search engine companies implement to prevent Web developers from "gaming" their search algorithms. In addition to accruing incoming links from other websites, another SEO technique is to create Web pages rich in the "keywords" that potential users are likely to search for (e.g., "best locksmith," "affordable tutor"). Ideally, AllDone's software engineers would develop software algorithms to add vast amounts of keyword-rich text to their Web pages. However, this strategy was deemed too risky because search engine companies deploy their own algorithms to detect autogenerated SEO text and penalize websites that attempt to "game" their systems by posting artificial content.

Given the risks of automating SEO content generation, AllDone instead built a writing team in the Philippines to help the firm attract buyers. A software engineer set up an administrative portal that would show writers descriptions of AllDone sellers and the most popular keywords for the services they offered. Every month, ADP team members wrote approximately 50,000 keyword-rich descriptions of the services that sellers offered, which ADSF engineers then added to thousands of automatically generated pages.[8] During the second quarter of 2012, the writing team almost doubled its output as its managers in the Philippines opened 16 new positions (growing from 37 to 53) and implemented a change in the team's pay scale (from an hourly wage to a piece-rate) to boost productivity. Direct observation of human–software complementarities at AllDone reveals an advantage of tacit knowledge not previously predicted by continuity theorists. When software systems are designed to defend against algorithmic efforts to "game" them, human variability and imperfection may be able to circumvent software's strictures.

At the same time that ADP's writing task functioned to draw buyers to the website, it operated in tandem with machine-learning algorithms designed to recruit sellers. Vince, an ADSF software engineer, created a "Web crawler" program that would scour the Internet for information associated with prospective sellers. The machine-learning algorithm was tuned by "big data" gathered from ADP workers who viewed the algorithm's assessments of more than 10,000 Web pages. Workers recorded whether the program had accurately determined whether the Web page belonged to a prospective seller and whether it had accurately determined each seller's name, e-mail address, services offered, and location.

The workers' corrections were then fed back into the program to help to refine the algorithm.

When a buyer placed a request on AllDone, her request would be sent to relevant sellers who had already established profiles on AllDone and also to all of the e-mail addresses that the crawler had identified as belonging to potential sellers of the service in that area who might be interested in joining AllDone and getting in touch with the buyer. By early 2013, the crawler had visited 12 billion Web pages and found 7 million potential sellers; of those, 400,000 had signed up (though about one third would later deactivate their profiles), at a cost of 40 cents per acquisition.[9] AllDone thus used a combination of human labor and machine learning to build a nationwide user network at minimal cost.

In sum, during the first phase of my fieldwork, AllDone confronted machine lag as software engineers' vision outstripped the limitations of technology and the organization's resource deficits. The company deployed functionally and numerically flexible computational labor to support growth and innovation, complementing software algorithms by enacting nonroutine tasks, reverse substitution, and workarounds.

Phase II: From Human Lag to Emotional Labor

> With Series A [funding rounds], you're selling a dream [to investors]. With Series B, you're selling a spreadsheet. (Peter, AllDone's CEO)

During the second phase of my research, AllDone's executive team redirected its focus from generating more user activity to generating more revenue. They began to present data in weekly staff meetings tracking the firm's progress toward becoming a profitable, self-sustaining company—and, by implication, a worthy investment for later stage venture capitalists.

Efforts to convert user growth into revenue growth drew top management's attention to—and at times exacerbated—instances of *human lag*, when users were reluctant to accept or resisted AllDone's software systems. Users who were confused by or dissatisfied with AllDone's service were less likely to become or remain paying customers; worse, some threatened to damage AllDone's reputation. AllDone's phone support team in the Las Vegas area complemented technological systems by performing *emotional labor* to help users adapt to the software. These frontline workers regulated their emotional display in an effort to instill particular emotional states in the customers with whom they interacted (Hochschild, 1983; Wharton, 2009). ADLV supported revenue growth by managing relationships with sellers, helping to keep them engaged and satisfied with the product.

ADLV's emotional labor was aimed at building, repairing, and preserving users' trust in AllDone's service. Trust is an essential element of market brokerage. For brokerage relationships to endure in competitive markets, buyers and sellers must believe that the broker is capable of meeting their needs and that the broker will not exploit its advantageous position to unduly gain at their expense (Stovel & Shaw, 2012). Trust-takers engage in emotion management to produce trustworthiness, ''perform[ing] acts of self-presentation in order to win the trust-giver's trust'' (Bandelj, 2009, p. 356). Trust-takers attempt to generate positive emotional responses in trust-givers by signaling their commitment, congruence of expectation, competence, and integrity (Beckert, 2006). ADLV's emotional labor mobilized human intuition, creativity, problem-solving skills, and powers of persuasion in interpersonal interactions to solve problems of trust—functions that were nearly impossible to automate, yet integral to the operation of AllDone's software and business.

Building Trust

Sellers often struggled to understand how to take advantage of AllDone's marketplace. Because AllDone had prioritized building a nationwide network of users as quickly as possible, its Web pages were designed to funnel sellers directly into the signup process, minimizing the presentation of complex information about how the service worked. Once they had joined AllDone and begun to use the marketplace, some participants struggled to understand core features of the product, including the quoting system, the payment structure, and the user interface. Misunderstandings and mistrust could be exacerbated by ADSF's frequent experimentation with the user interface and market rules. Initial tests of new features might be underdeveloped, and engineers often chose not to provide affected users with explanations of experimental changes. Rather than investing limited engineering resources in perfecting experimental product features that might eventually be abandoned, it was far easier for AllDone to, as one manager put it, ''throw bodies at the problem'' to help keep users engaged.

When users required extensive assistance, members of ADLV undertook interactive emotional labor to help confused buyers and sellers understand market rules and norms. In these cases, members of

AllDone Las Vegas taught sellers about the AllDone process over the phone, as in a 15-minute exchange that I observed in team leader Carol's home office.

> As we're sipping tea at her desk, the phone rings. Carol puts on a smile as she answers on speakerphone. The man on the other end speaks haltingly, asking for Theresa. "We don't have a Theresa," Carol responds, "but can I help you with anything?" After a little back-and-forth, Carol deduces that this seller, Ted, received a request from a buyer named Theresa. But Ted doesn't seem to understand what that means, or what he can do with the request. In fact, I'm beginning to get the sense that Ted doesn't even understand what AllDone is.
>
> After a few failed attempts at gathering Ted's account information, Carol is finally able to locate him in the system. Now that Ted has learned what a request is, Carol wants him to take a look at his queue. "Are you close to a computer?" she asks. "Uhhhhhh" is Ted's hesitant reply. "Or I could go over them [with you over the phone] if you'd like!" Carol offers without missing a beat.
>
> It turns out that Theresa's request won't work for Ted: he says he refinishes furniture, but doesn't upholster it. Carol launches into an explanation of how AllDone works. "AllDone, as you're unaware of what we do, we're a marketplace for local service providers." She outlines how requests are gathered from buyers and distributed to sellers, and informs him that it's "your responsibility to check your dashboard to see the leads you've been sent."
>
> "In other words, it costs me a dollar ninety-nine?"
>
> "Yes, if you'd chosen to accept Theresa's lead."
>
> Finally, Ted says that he wants to talk to his business partner about how they'll use AllDone. Carol says that Ted can call her back after he's done so, when she can "walk you through the whole process and get you up and running and get you lots of business, I hope!"

Carol's conversation with Ted highlights the challenges inherent in AllDone's efforts to rapidly draw a large number of new users into the marketplace. Carol was available to patiently explain to Ted how he could use AllDone to meet potential buyers. ADLV team members were equipped to explain both formal rules of exchange (e.g., how to reach out to potential clients; AllDone's fees) and informal norms (e.g., how to build a positive online reputation). Team members often went to great lengths to satisfy users—for example, they might offer to upload photos to a seller's profile if the seller was having trouble learning how to do so herself. ADLV employees thus endeavored to alleviate sellers' confusion and leave them confident that AllDone could support them in growing their businesses.

Repairing Trust

When sellers were dissatisfied with the results of the introductions AllDone provided, their trust in the company could be shaken. AllDone used ADLV to rebuild sellers' trust when market outcomes did not align with their expectations. Support agents strove to turn detractors into advocates by listening, counseling, reassuring, and caring for individual sellers.

Sellers who invested time in creating profiles and then paid to submit quotes to buyers tended to expect that their efforts would yield new clients. Some sellers' expectations were violated when they did not get hired for jobs or did not even receive written replies from buyers. Some sellers accused AllDone of brokering in bad faith: of connecting them with people who were just "price shopping" and not serious about hiring someone, or even of fabricating fraudulent consumer requests to increase revenue. After receiving no contact from buyers on two quotes, one seller complained via e-mail: "Like so many of these online systems yours is inherently corrupt. And I think you must know this. Neither of these folks are serious. I've wasted a bag of groceries for my kids on false leads." More than one seller would note that if five wedding photographers quoted on a bad request, AllDone would have made $75, with the sellers having no chance to realize any returns.

When users' experiences fell short of their expectations—when they complained of feeling dissatisfied, taken advantage of, misled, disappointed, or disillusioned—threats formed that could, on the aggregate, jeopardize AllDone's position as a broker. A recorded phone call from one seller began:

> Let me tell you: I canceled my subscription to AllDone. I hope you're recording this—you guys are the biggest piece of shit I have seen [. . .]. It is absolutely, utterly useless. On the one hand, I feel you guys have stolen

> my money. On the other hand, I never got any responses. I make it a point to badmouth AllDone to other photographers.

The seller's anger is rooted in a betrayal of trust (''I feel like you guys have stolen my money''). These perceptions could be damaging to a broker whose revenue stream relies upon a reputation for making high-quality connections.

Members of ADLV endeavored to repair relationships with sellers who believed that the company was taking advantage of them, applying interactive emotional labor aimed at ''turning them around'' or reversing sellers' sense of betrayal and convincing them that it was in their interest to continue to use AllDone's services. In addition to offering material concessions such as refunds or credit, ADLV team members most frequently used a combination of three tactics to accomplish this: ''tough love,'' education and counseling, and personalized attention.

Phone agents dispensed ''tough love'' to help sellers develop more realistic expectations about market outcomes. During a videoconference between ADLV team member Nancy, Carol, and I, Nancy recounted an exchange with a contractor who was upset that the in-home estimates he had given to AllDone buyers had not resulted in jobs. ''You have to give them a wake-up call,'' she said, summarizing the conversation. '''I understand [your frustration]—but how's your business outside of AllDone? Do you usually get every job you go on?' 'No, I don't.' 'Well, this is the same thing.' [...] They need to hear this.'' In offering ''tough love,'' team members attempted to persuade sellers to accept blame for their poor market outcomes.

In the process of adjusting sellers' expectations, team members would also try to teach sellers how they could improve their performance in the future by demonstrating their competence and motivation to potential customers (Cook, Cheshire, Gerbasi, & Aven, 2009). When sellers called because they were upset that buyers were not responding to their quotes, Nancy would often investigate their records and see that the sellers had been ''really sloppy'' with their responses to buyers, sending quotes containing poor spelling or grammar that buyers might find unprofessional. Nancy reported frequently asking sellers to imagine that they were the buyer reading the seller's message.

> ''Would *you* respond if you received that quote? You have to work on this. The leads will not work themselves.'' [...] You have to show them how to stand out from the rest, how to answer in a nice way so people want to respond.

Carol concurred: ‘‘Often they put a generic thing in there. Nancy and I tell them you’ve got to stand out.’’ At an ADLV team party, Casey, another phone agent, remarked, ‘‘I need a button to press to be like, ‘you need to work on your profile’’’ because she frequently found herself repeating this instruction to sellers.

Sellers at risk of leaving AllDone were often simply reassured by the personalized attention that they received from support agents, sometimes within the context of ongoing relationships. As Carol explained to me during my first visit to her home office,

> We have special-needs [sellers]. You hear all about their personal life: ‘‘my boyfriend,’’ etcetera. We do therapy, as well, at ADLV! But we give them love. We handle them all the same way, we give them love, and whatever they need.

I later observed what team members often dubbed ‘‘AllDone love’’ in action during a 45-minute call with a seller:

> The phone rings again. This woman is having trouble logging in to her AllDone account for her drapery business. Carol says she is happy to help, and adds some pleasant chit-chat as she looks up the account. ‘‘How’s your day going so far, Sue? Looking forward to the holiday weekend?’’
>
> Sue says that she hasn’t been receiving requests. ‘‘Do you have time?’’ Carol asks. ‘‘We can go into your user page and take a look, see what’s going on. Because we generally find the more information you give us, the better we can match you with the requests that come in.’’
>
> Carol’s phone chirps to signal that calls are coming in on the other line and going straight to voicemail while she coaches Sue. ‘‘Yay!’’ she cries out at one point, clapping her hands three times. ‘‘Good job, honey, you’re good at this!’’ Sue tells Carol about her new iPad and her issues with identity theft. ‘‘We’ll have to see if we can get you some more business, Sue,’’ Carol says, redirecting the conversation. She adjusts Sue’s service categories, and her travel preferences, as Sue describes the parts of Wisconsin where she is most likely to find a market for her drapery business. ‘‘My God, can you imagine doing drapery for a mansion that size?’’ Carol marvels politely. ‘‘Goodness me. Well, drapery is a skill, not everybody can do it. What a great talent that is, Sue.’’

> Carol goes on to explain, "We're not like some companies where you can pay to get on top of the list" of buyer searches. Now Sue is having trouble logging into her account from her iPad. Carol finally wraps it up: "You can call this number any time and you can call us whenever you need us. We have a plan, Sue! We have a plan," she giggles. "Great chatting with you, Sue! I'll be talking to you later on or tomorrow."

Sue had called because she was having trouble accessing her account, but she then revealed that the service was not working as she'd expected it to: She had not been receiving requests from potential buyers. Carol patiently walked Sue through her account settings to improve the introductions she'd receive. In addition to providing the seller with accurate and useful information, Carol flattered her and engaged her in conversation unrelated to her drapery business. At the end of a long call, Carol promised to follow-up with Sue to make sure that everything had worked out.[10] Carol's interventions were aimed at making Sue feel like the company cared about her and wanted to help her business succeed.

When market outcomes violated sellers' expectations, AllDone deployed emotional labor to help users adjust to its software systems and to secure sellers' ongoing participation in the market. Each aspect of the task was sufficiently nuanced to require human intervention. Workers assessed market participants' emotional states, displayed empathy, understood and addressed their needs, and quickly developed relationships that in many cases restored users' confidence and trust in the company. Team members used their interpersonal skills to persuade many sellers to continue their relationship with AllDone by convincing them that, although the system hadn't met their expectations in particular instances, in general AllDone itself was trustworthy and fair.

Preserving Trust

AllDone's pursuit of revenue could also exacerbate human lag by leading to changes in the service that users perceived as unfair. A year earlier, in a bid to incentivize sellers to submit more quotes to buyer—and to convince potential investors that AllDone could connects a high volume of market participants—the company had created a "subscription" payment option that allowed sellers to send unlimited quotes to buyers for a flat monthly fee. Quote volume increased dramatically, helping AllDone secure its first major round of funding. Soon, however, management began to worry that subscriptions were limiting AllDone's revenue potential: Analyses showed that even as AllDone's investments in SEO were bringing more and more buyers to the website to place requests, revenue growth was not keeping pace with user growth. In effect, sellers' responses to the avalanche of new requests had already been paid for. For example, a seller paying a flat fee to respond to 10 requests per month in 2011 paid the same price to respond to 20 requests per month in 2012.

AllDone's senior managers developed a new payment model through which revenue would increase linearly with request volume. Sellers would pay to purchase AllDone's internal currency of "coins" and would relinquish coins for each quote they wished to submit to buyers. The more quotes a seller submitted, the more she would have to pay AllDone. The stakes were high in the company's transition to the coin system. AllDone's most valuable relationships were generally with sellers who submitted a large volume of quotes in competitive service categories and locations. Some would see the price they paid to contact potential clients increase by as much as 10 times. The transition would succeed only if AllDone could convince a substantial fraction of its high-value subscribers to remain active on the market while paying higher fees. If the change alienated too many existing sellers, revenue would plummet, leaving the company's future in doubt.

ADSF first experimented with explaining the change to a small group of sellers via automated e-mail messages and received overwhelmingly negative responses. Adam, the director of engineering, soon sent an e-mail to me and Josh, the product manager, with an idea for subsequent tests: "What if we did the coin transition for the high value subscribers via individual phone call?" Josh liked the idea—not only could this method help AllDone preserve relationships with its most important clients, but one-on-one phone calls would also allow the firm to better understand users' questions and concerns, as the phone team could record sellers' reactions and relay them to ADSF to help inform the company's messaging around the transition to the coin system.

The following week, Carol and Nancy placed nearly 100 experimental transition calls. After they had finished the calls, we held a videoconference to discuss the results. While both Carol and Nancy remained professional, they were clearly exasperated: As expected, most of the affected sellers were livid. "We're beaten up this week," Carol told me. Nancy likened the calls to a "frontline battle"; Carol called it "guerilla warfare" conducted "in the trenches." "They're insane," Nancy said of the sellers she called. "People don't like change." Nancy recalled telling some sellers about the transition: "So many people were like, 'What are you talking about?'" Later, the sellers would read the e-mail AllDone's engineers had sent them detailing the termination of subscriptions "and call up screaming."

I was worried about the emotional toll that these calls would take on the team. It made me very uncomfortable to hear about the verbal thrashing that Carol and Nancy were absorbing, given that if their tests were successful, the entire team would be enlisted to complete this task with thousands of additional sellers. I sent an e-mail to Adam and Josh detailing the results of ADLV's initial tests, emphasizing how "brutal" the calls had been, and forwarding a message from Carol detailing their difficulties. What stood out to Adam, however, was not the abuse that sellers were heaping upon Carol and Nancy, but rather Carol's description of what the team had been able to accomplish under pressure: "We've managed to get most of them to come around and agree to try the coins, but some of these calls are taking a lot of time and a lot of tap dancing, *with jazz hands, of course!*" Carol and Nancy had patiently listened to sellers' concerns and worked to persuade angry sellers that AllDone would continue to provide a strong return on their investment. In most cases, a difficult 15- or 20-minute conversation would help to assuage sellers' fears. Adam's reply underscored the critical function of ADLV's emotional labor:

> My first thought is this is a *really* good use of the phone team if they are able to consistently turn people's anger into trying the new program. At the end of the day we have to convert people to paying 5-10x as much as they were previously for the same thing, so there is no tap dancing around that issue or tricks we can pull (other than messaging it the best way we can).

The company's head software engineer was now convinced that there was no technological solution to the problem. AllDone's software systems alone could not manage sellers' feelings of betrayal amid a radical change in the payment structure. It was precisely the fact that ADLV's one-on-one conversations with sellers were so difficult that revealed just how crucial they would be to a successful transition.

After 4 months of testing and refining the program, the product team was confident that the company could safely undertake a site-wide transition to the coin system because ADLV had demonstrated that they could convince many subscribers to pay more to continue to use AllDone. Carol and I began to train every member of ADLV to make transition calls, and coins were introduced to all sellers over a period of 5 weeks. Every week, ADLV team members called hundreds of AllDone's most active subscribers, announced the change, and tried to persuade sellers to try the new program. In total, they reached out to nearly 5,000 high-value subscribers and received calls from many more.

Like customer support representatives in offline settings, ADLV's female, frontline workers paid an emotional toll to uphold exchange relationships and the accumulation of profit (Brotheridge & Grandey, 2002; Hochschild, 1983; Weeks, 2007). Team members were battered with insults and verbal abuse for 8 hours a day throughout the transition period, and none escaped without being brought to tears. But the rollout proved successful: Revenue climbed as a significant portion of former subscribers continued to use AllDone in spite of the higher fees. Years later, executives continued to view the termination of subscriptions as the most pivotal moment in AllDone's history, unlocking the company's revenue growth for years to come. As Carter later reflected in an e-mail, "We absolutely could not have made this critical transition without Vegas." Technology alone was incapable of solving the problems that arose when AllDone sought to profit from using software to administer a nationwide marketplace. Instead, the company used emotional labor to help bring users' expectations in line with its software systems.

Phase III: Permanent Lag

> When I interview engineering candidates, one of the most common questions they ask me is what's changed the most at AllDone since I started. I tell them that at first we had no time to work on the stuff we wanted to work on. Now we have enough people to do what we want. (Brett, AllDone software engineer)

After AllDone completed its transition to a new business model, its executives began to meet with VC firms and soon accepted more than $12 million from one of the industry's leading VCs. With considerable financial resources at their disposal, AllDone's leaders sought to combine the company's two prior strategies. As in Phase I, AllDone would again seek to grow the organization, the user base, and the product; this time, however, expansion would occur within the revenue generation framework that was introduced in Phase II, which would help the firm remain an attractive target for later stage VC investments. The dual goals of expansion and revenue generation existed in tension with one another: On one hand, AllDone sought to draw more users into the website and convert them into active, paying customers; on the other hand, the company's attempts to monetize their activities could spur user dissatisfaction and exit. These tensions contributed to the reproduction of machine lag and human lag, as developers' ambitions continued to race ahead of technology, and users continued to chafe at AllDone's software systems. The expansion and institutionalization of AllDone's complementary work teams to address these problems demonstrated management's recognition that machine lag and human lag were not temporary features of software development to be shed by a more mature firm, but rather that they represented a *permanent lag* inherent to the dynamics of the enterprise.

Much of AllDone's newfound funding was funneled into the engineering team, greatly expanding the company's capacity to develop software. Members of the organization began to confront the possibility that these developments could sever the previously symbiotic relationship between technological systems and human workers. ADSF's engineers were making significant progress toward automating the labor-intensive processes of vetting buyer requests and manually matching buyers and sellers, tasks that now occupied nearly half of ADP's 200 workers. Automation was accomplished through the implementation of machine-learning algorithms that were "trained" by the vast dataset of workers' past operations. The vetting algorithm used buyer request text and metadata to determine which were most likely to be rejected by human screeners, and the matching algorithm learned which types of requests, in which locations, were most likely to be matched with particular types of sellers. The software algorithms took over an increasing percentage of each task as developers continued to tune them. ADSF engineers had formulated plans to automate other ADP functions as well (e.g., running background checks on sellers and sending follow-up e-mails to apologize to buyers who did not receive quotes), or to create systems that would offload ADP's labor onto users (e.g., the process of categorizing sellers in the AllDone database). Carter asked Ross, the leader of ADP's matching team, to create detailed plans to prepare for the termination of a significant portion of his team. Hoping to cushion the blow of the anticipated layoffs, the cofounders reached out to executives at other start-ups in search of a company that might "adopt" the team for its own purposes.

Although much of ADP's reverse substitution was itself reversed by automation, the anticipated cuts to ADP's workforce never materialized. In fact, as the ranks of ADSF software engineers increased from 8 to 50 over the subsequent 2 years, ADP swelled from 200 to 800 workers. Meanwhile, recognizing the importance of emotional labor to the company's continued success, executives moved the company's phone support functions to an office in the Salt Lake City area (known as AllDone Salt Lake, or ADSL), where ADLV's team of 10 was replaced by 200 employees. Even as AllDone raised three additional rounds of VC funding totaling more than $250 million—and achieved a valuation of more than $1 billion—technological change increased demand for existing functions that could not be automated while creating new work in and around changing software systems. Below, I detail two instances in which AllDone used human workers to compensate for the systematic reproduction of machine lag and human lag.

Machine Lag and Computational Labor

Although software engineers' increased rate of innovation led to the automation of some computational labor, it also created new demand for computational labor designed to balance the interests of sellers with those of the firm. One example emerged when ADSF's product team attempted to build new software systems to mitigate the tension between user growth and revenue generation. Although the transition to the coin system had enabled revenue growth, half of the sellers who had submitted five or more quotes per month before the transition were no longer submitting any quotes afterward, indicating widespread dissatisfaction. Previously, a seller with a subscription could, without incurring any additional costs, send a quote to a buyer whose request was vague simply to ask for more information. With the introduction of coins, sellers were submitting consumer requests to far greater scrutiny because they now had to risk their money on a job that might be revealed to be a poor fit once they paid to submit a quote and then received additional information from the consumer. Even sellers who had not previously been using the subscription model found themselves subject to a more restrictive refund policy.

In response to these concerns, AllDone's product team sought to devise a process that would allow sellers to solicit more information from buyers while simultaneously ensuring that they continued to pay to pitch their services. ADSF engineers tested a "Q&A" feature that would allow a select group of sellers to ask buyers for more information about their requests before submitting a quote. However, some sellers discovered that they could use the Q&A feature to subvert AllDone's payment model: Rather than simply posing a query, sellers might include their contact information or a price estimate as if they were submitting a paid quote; others asked buyers to reply with their own contact information. To combat sellers' abuse of the system, ADSF's product team publicized guidelines specifying which kinds of questions were permitted and forbidden, but the issues continued. When engineers built software algorithms to filter phone numbers and e-mail addresses out of Q&A messages, sellers developed "workarounds" like spelling out their phone numbers ("seven seven zero ... ") and e-mail or Web addresses ("Jim [at] AceConstruction [dot] com"). When some sellers abused the system, others who had abided by the rules would denounce the violators in rancorous messages to the buyer and to AllDone.

In light of sellers' problematic responses to the Q&A feature, the product team called upon complementary workers to prevent sellers from circumventing AllDone's payment model. One developer created a Web portal through which members of ADP could screen each message before it was distributed. Workers studied the program's rules and vetted each message accordingly, editing or deleting those that violated AllDone's guidelines. They also edited questions for grammar and spelling errors to improve buyers' impressions of AllDone sellers. Users whose messages were edited or deleted received an automated e-mail informing them of the nature of their transgression. When the Q&A program was proven to have no significant deleterious effect on quote volume, it was instituted site-wide. Whenever a market participant submitted a question or response, day or night, workers in the Philippines would be on call to vet it immediately and distribute only approved messages to buyers and sellers. Workers' tacit knowledge and common sense allowed them to determine which sellers were trying to cheat the system, helping AllDone balance its goals of expansion and revenue generation.

Human Lag and Emotional Labor

As the engineering corps grew, ADSF's pace of experimentation and production increased. When developers worked to optimize AllDone's product, they introduced the possibility that users would struggle to adapt to novel and sometimes opaque systems. Emotional labor remained crucial to sustaining sellers' trust amid continual innovation and periodic price increases.

The activities of the engineering subunit in charge of AllDone's matching algorithms provide one such example. Previously, buyer requests had been distributed to all of the eligible sellers in any given area, with AllDone accepting quotes from the first five sellers to respond. But as AllDone grew, some cities became saturated with hundreds of active sellers, many of whom were frustrated that they could not respond quickly enough to be one of the first five sellers to submit their quotes to buyers. ADSF's software developers began to use machine learning to optimize request distribution, gathering information about sellers' past behavior to determine who was most likely to submit a quote on each incoming buyer request. The software then "reasoned" from this information to generate decisions about which sellers should receive new requests. These new matching algorithms were designed to send requests to only as many people in a given locality as the models predicted would be needed to deliver at least three quotes to each buyer within 24 hours.

As ADSF tinkered with the matching algorithm, some of the sellers who had previously been frustrated that they received too many requests that they were unable to quote on were now upset that they were no longer receiving enough requests at all. When sellers noticed that something had changed, they could reach out to phone support agents who worked to reassure them that AllDone was refining its systems to create the best possible experience for its users. The growth of AllDone's phone support team reflected the significant role that workers continued to play in educating users and keeping them engaged amid ADSF's frequent experimentation with market systems.

AllDone's developers did not anticipate that the company would ever settle on a perfected, static set of rules and systems. According to Brett, ADSF's lead on the matching algorithms, "there are no easy answers" because software algorithms cannot determine "one clear right way" to balance the shifting interests of buyers, sellers, and the company. Frequent changes to the product put pressure on AllDone's phone support agents to assuage users' concerns and to transmit user feedback to ADSF so that it could be incorporated into further changes. For another project, Brett's team used machine-learning algorithms to formulate a dynamic pricing model. The model was trained on data from past requests to predict how many sellers would be likely to submit a quote for each new request. If the model predicted that a request would be desirable, it would charge sellers more to send a quote; if it predicted low demand, prices would decrease. Even before releasing the feature, Brett was anticipating the response: "I know [sellers] are gonna be upset" when the program is implemented and some discovered that they would have to pay more to submit quotes, he explained. Brett confirmed that ADSL would again play a crucial role in intervening to sustain sellers' trust in AllDone as the company tried to squeeze more revenue out of users.

ADSF's increased pace of production and ongoing experimentation with market rules and systems increased demand for computational labor to help software systems meet developers' needs and emotional labor aimed at bringing users up to speed with changes. Machine lag and human lag were never permanently "solved," as the company continually recalibrated its systems to balance its own interests with those of its users. The expansion of the complementary labor that addressed these issues revealed that the uneven development of relations between humans and machines was not simply an artifact of an immature firm but instead represented a permanent lag inherent in the dynamics of the enterprise.

Discussion

While some argue that the rise of software automation threatens workers with obsolescence, others assert that new complementarities between humans and software systems are likely to emerge. This study of a software firm operating on the frontiers of the digital economy offers a longitudinal account of the processes through which computer code and human labor coevolved within a particular organizational context. During Phase I, when AllDone prioritized expansion with little regard for profitability, the pace of product innovation often outstripped the capabilities of technology and the company's resources. The firm relied on computational labor to grow its user base: Workers were called upon because their tacit knowledge gave them a comparative advantage over software in performing nonroutine work; they executed routine algorithms by hand in a process I call reverse substitution; and they enacted workarounds designed to "game" software systems. In Phase II, the strategic pivot to revenue generation made it increasingly important for the company to help users adjust to its software systems. AllDone used emotional labor to build users' trust in its service, to reactively repair damaged ties with users, and to proactively preserve users' trust in the face of changes to market rules. During Phase III, machine lag and human lag were systematically reproduced by AllDone's simultaneous prioritization of expansion and revenue generation. This permanent lag increased demand for complementary labor, and the intertwining of technology and work remained a durable aspect of organizational development amid accelerating automation.

By closely examining an organization where software algorithms were produced and implemented, this study makes both empirical and theoretical contributions to research on how software automation is transforming work and employment. Continuity theories reject technological determinism and instead anticipate that digital technologies will spur new human–machine complementarities. Empirically, I have detailed two ways in which human workers can complement software systems: by providing computational labor that supports or stands in for software algorithms and emotional labor aimed at helping users adapt to software systems. As an increasing proportion of economic exchange becomes technologically mediated, demand for workers who supplement software infrastructure is likely to increase. Human brokers, support agents, consultants, and salespeople whose skills in advising and persuading clients are difficult to automate may play an important role in smoothing the misunderstandings and frustrations that can arise when humans interact with software systems (cf. Bessen, 2015a).

Theoretically, I have identified one mechanism that generates the systematic reproduction of the machine lag and human lag that give rise to shifting configurations of workers and machines: the internal dynamics of organizations. Organizations are generally dynamic insofar as they are constantly competing for scarce resources—including customers and capital—and even because of the ingenuity of their customers, who may devise creative ways to avoid paying fees. The organizational contexts in which sociotechnical systems are embedded matter for how human–machine configurations emerge, endure, and change. The specific articulations of AllDone's work teams changed as the company adapted its goals to meet venture capitalists' expectations. At each stage of AllDone's development, software alone was unable to resolve all of the problems that the firm faced as it grew; indeed, the application of technology itself often caused new problems that created new demand for human workers capable of performing complementary tasks. Instead of perfecting software algorithms that would progressively push people out of the production process, managers continually reconfigured assemblages of software and human helpers, developing new forms of organization with a dynamic relation to technology.

Conclusion

Formulated as a refutation of discontinuity, continuity theory fails to grasp these organizational dynamics of continuity: the *discontinuity in continuity*. Contemporary inquiries into the relationship between software automation and work have largely investigated macrolevel phenomena such as labor markets, job categories, and general work tasks. This study reveals processes through which, at the level of the firm, complementary labor is continually incorporated into and displaced from software systems. At AllDone, full automation of every task was impractical given the instability of the firm's environment and product. Software engineers' innovations often required the assistance of complementary workers. When particular complementary tasks were automated, outdated labor processes were typically replaced by new functions that supported new innovations. As continuity theorists predict, human labor remained integral to the operation as software automation increased at AllDone; however, the texture of continuity was itself dynamic and discontinuous, the result of repeated transformations in human–machine configurations. The very fact of continual change in markets and firms in an age of so-called "disruption" is what makes the continual application and reconfiguration of complementary labor vital to organizational success, whether workers are helping a firm experiment with new processes to grow faster, managing users' emotional responses to rapid change, or doing what software alone cannot accomplish. One of the sources of labor's enduring relevance in the digital age may thus be the dynamism of firms.

In linking the phases of capital accumulation to organizational form within a new venture, this study also suggests that the rate of organizational change will be an important determinant of the outcomes of human–software configurations. This study was based on observations of variation within one organization over time. Researchers may discover additional human–machine complementarities and develop further insights into their dynamics by drawing comparisons between organizations. The use of online, distributed workers to complement computer code extends far beyond software start-ups: Large and well-established organizations in both the high-tech field (Google, LinkedIn, Netflix, AOL) and in more "traditional" sectors (Unilever, Walmart, Coca-Cola, Proctor & Gamble, U.S. Army) frequently call upon them as well (Bergvall-Kåreborn & Howcroft, 2014; Kingsley, Gray, & Suri, 2015). Nascent firms relying on "opportunistic adaptation" (Bhidé, 2000) to secure scarce resources in unsettled markets may be more likely to experiment with using low-cost and abundant workers to complement software infrastructure than established firms operating in more stable markets, which are likely to experience a lower rate of strategic "breakpoints" (Barley, 1986). In larger firms, human–software configurations are likely to change more slowly than at start-ups like AllDone. This stability could correspond to a higher rate of automation and substitution. In short, the outcomes of technological innovation are unlikely to be monolithic but will instead be dependent upon the context in which software development is applied and the rate at which that context changes.

Sociologists and organization scholars have devoted surprisingly little attention to the ways in which software automation is transforming work and employment. This is particularly unfortunate because their diverse methodological approaches make them uniquely suited to contribute to the debate by examining the processes through which work and technology are coevolving within dynamic organizational contexts. Instead of accepting discontinuity theorists' vision of a linear march toward autonomous machines that replace humans, researchers must continue to view technological innovations with the following questions in mind: "Where are the people? Which people are they? What are they doing? When are they doing it?" (Mindell, 2015, p. 13). When scholars examining the future of work and employment focus their inquiries only on how technology substitutes for workers, they are likely to miss ever-more intimate interconnections between people and technology. The shape of this interplay—and our ability to render it visible—may be among the defining issues for scholars of work and employment in the 21st century.

Appendix A: Defining AI

In the most general sense, computer scientists agree that AI aims "to simulate [the] intelligence and rationality of humans" (Zackova, 2015, p. 32) or to create "systems that can reasonably be called intelligent" (Russell & Norvig, 2010, p. 34). Yet, scientists and practitioners have long debated the boundaries of "intelligence," with their positions shifting over time as expectations for AI have ebbed and flowed along with the field's successes and failures (Ekbia, 2008; Wise, 1998). For example, the first computer programs capable of simulating human chess players were widely hailed as "intelligent"; today, the architecture behind early programs would be viewed as nothing more than rudimentary decision trees. An AI program can manipulate some of the structures of language to hold a coherent conversation with a human but cannot "understand" the logic of language as humans do nor can it comprehend why different forms of expression are appropriate in different contexts (Dreyfus, 1992; Winograd & Flores, 1987). A machine-learning algorithm might predict outcomes with a high degree of accuracy, but its opacity can make it difficult to reverse engineer the algorithm to interpret the reasoning underlying its decisions (Alpaydin, 2014; Burrell, 2016). There remains widespread disagreement surrounding the extent to which such software should be considered "intelligent."

Practitioners approach devising AI systems from many perspectives, resulting in a great deal of definitional ambiguity (Ekbia, 2008). Some focus on processes through which computers can "think" and reason. Others emphasize different sorts of "behavior," or the output of software systems. Both camps are themselves split between those who strive to develop AI that replicates human thought or action and those who aim to produce programs that measure up to a nonhuman, "rational" ideal (Russell & Norvig, 2010).

In practice, most real-world applications of AI aim to construct "rational agents" programmed to act "so as to achieve the best outcome or, when there is uncertainty, the best expected outcome" (Russell & Norvig, 2010, p. 4). Although in AI's earliest incarnations computer scientists envisioned smart machines that could act upon general-purpose laws across domains, more recent advances are typically task-specific: One set of algorithms might be programmed to play chess; another to estimate bus arrival times; another to diagnose diseases; and yet another to write poetry (Zackova, 2015). Eschewing the Herculean task of modeling the workings of the human brain, modern AI systems based on "neural networks" or "machine learning" instead use large data sets and statistical logic to infer and respond to patterns (Autor, 2015a; Dreyfus, 1992; Pratt, 2015).

AI has already become part of our everyday lives: "AI technologies underlie many Internet tools, such as search engines, recommender systems, and Web site aggregators" (Russell & Norvig, 2010, pp. 26–27). AllDone is no different in this respect. Rather than attempting to build an integrated, "generalized" AI, AllDone's software engineers developed discrete machine-learning algorithms to perform particular tasks.

Questioning the Human–Machine Boundary in AI

Popular understandings of AI frequently draw too sharp a distinction between human and machine. Researchers from a variety of disciplinary backgrounds have demonstrated that, in practice, the boundaries between workers and technologies are more fluid than we are often led to believe. Such scholars call into question the autonomy—indeed, the very artificiality—of AI.

AI has long been associated with notions of autonomy (Ekbia, 2015; Ekbia & Nardi, 2017), raising expectations of machines acting free from direct human involvement. However, just as computer scientists have debated what constitutes "intelligence," so too have they called into question the notion of "artificiality." Since the early days of computing, theorists and practitioners have pointed to the role of human interaction in supporting the functioning of "smart" machines (Mindell, 2002;

Suchman, 2007). After finding that AI systems frequently failed in complex, unpredictable, and dynamic real-world environments (Dreyfus, 1992), many of the early pioneers of AI eventually came to disavow the notion that "intelligent" computer systems should be designed to function without human intervention. Instead, leading figures in the AI community began to advocate that programmers focus on creating useful assemblages of software and humans that would simultaneously take advantage of the computational powers of machines and the multifarious (and sometimes mysterious) capacities of human cognition (Winograd & Flores, 1987). This would require developers to reframe how they thought of AI: Instead of conceiving of computers as replacing human brains, programmers began to consider how AI works "in the context of human practice," as part of the broader social systems into which they are inserted and which they enable (Winograd & Flores, 1987, p. 4).

Scholars across the natural sciences, social sciences, and humanities are becoming increasingly attentive to how human social practices interface with and are often constitutive of "intelligent" software systems. In the words of a Microsoft researcher, "[i]nformation systems are always swarming with people; we just can't always see them" (Gillespie, 2016, p. 26). Conceiving of AI as fully "artificial" reinforces a vision of autonomy that does not always reflect reality (Romportl, 2015). According to engineer and roboticist Ken Goldberg, today's AI practitioners often focus on "designing interfaces that keep humans in the loop." (2015, p. 321). Rather than building toward an AI "singularity," Goldberg argues, engineers are instead embracing "multiplicity," a term that describes "an emerging category of systems in which diverse groups of humans work together with diverse groups of machines to solve difficult problems." (p. 321). Accounts of the power of AI to eliminate the human element of production, Goldberg (2017, p. 7) posits, often miss "the essential role that humans play" in smart software systems (p. 7). In his examinations of robotic vehicles, historian and engineer David Mindell, too, challenges the "myth of full autonomy" by pointing to the ways in which "[a]utomation changes the type of human involvement required and transforms but does not eliminate it" (Mindell, 2015, p. 10). In addition, scholars of human-computer interaction (Lee et al., 2015) and computer-supported cooperative work (Raval & Dourish, 2016) continue to emphasize the often-overlooked activities of human workers in AI systems.

The work of informatics scholars Hamid Ekbia and Bonnie Nardi is particularly relevant to understanding AllDone as a case of AI in action. Ekbia and Nardi (2017) contend that whereas observers tend to focus on how AI catalyzes automation that reduces demand for human labor, AI can also spur *heteromation*, a term that denotes how humans ‘‘are drawn back into the computational fold in new ways,’’ often operating ‘‘on the margins of machines and computerized organizations.’’ Researchers and journalists have documented many pertinent examples of software systems that are widely recognized as AI, yet whose functioning is nonetheless enabled by hidden human labor. Indeed, much of the recent impact of AI on our everyday lives can be attributed to the fusion of specialized software systems with human cognition:

– Google’s search algorithms sort through billions of Web pages to return relevant results to users. Google relies not only on ‘‘smart’’ software that learns people’s preferences by tracking clicks and gathering user feedback but also on input from paid human workers who rate the relevance of search results according to Google’s written guidelines. Workers’ ratings are then fed back into the search algorithms to improve their performance (Irani, 2015c).
– Or consider the driving directions generated by Google Maps to help users reach their destination as quickly as possible. Google Maps’ software algorithms draw on current and historical user location data to predict traffic conditions. But they also rely on the activities of hundreds of human operators whose numbers are commensurate with the ‘‘big data’’ that they handle. Operators use street-level photographs to manually code features like speed limits, one-way streets, or prohibited left turns into accurate digital representations of the physical world (Madrigal, 2012).
– Similarly, the AI technology behind self-driving cars also requires human workers to painstakingly label thousands of images to train the software that ‘‘sees’’ roads, pedestrians, street signs, and other objects (Both, 2014).
– Netflix combines human and machine intelligence to provide users with recommendations in ‘‘personalized genres’’ (e.g., ‘‘British set in Europe Sci-Fi & Fantasy from the 1960s’’). Netflix offered a $1 million prize to the engineering team that could best improve the accuracy of its recommendation algorithms, which make suggestions based on users’ stated preferences, viewing history, and similar viewers’ taste profiles. However, the company ultimately found the new models unsatisfactory and instead chose to hire workers to view and classify all of its content. Workers receive a 36-page training document to assist them in labeling each movie or TV show according to dozens of attributes. These workers’ efforts help Netflix categorize

content into nearly 77,000 microgenres that can be microtargeted to individual viewers (Madrigal, 2014).
- Facebook's News Feed uses machine-learning algorithms to determine which posts to present to each user and which materials to hide. However, software alone cannot keep up with the deluge of violent, disturbing, and inappropriate content uploaded by users every day. Like other social media companies, Facebook employs content moderators who follow complex guidelines to screen out material that threatens to make Facebook an unsafe—and unprofitable—environment (Hopkins, 2017; Roberts, 2016). In response to violent incidents streamed live on Facebook, the company recently announced plans to hire 3,000 additional content moderators (Goel, 2017).

AllDone's founders may have initially imagined that they would be building a fully autonomous software system. However, like many other software developers, they soon learned that they could develop software most effectively by combining machines with human workers to execute particular functions. As in the above examples, I conceive of AI not as a finished product emptied of human labor nor do I view AllDone's product as a unitary "AI interface." Instead, AllDone's product is constituted by an assemblage of evolving sociotechnical systems in which humans are at times incorporated into and at other times displaced from computational infrastructure. Aware of AI's limitations, developers integrate human workers into AI systems to exploit the comparative advantages of each. At AllDone, human workers did not simply "fill the gaps" when technology fell short. The coconstitution of software algorithms and human labor was instead integral to the design and implementation of myriad sociotechnical systems. This is, I believe, one of the important contributions of the article, which demonstrates that only by observing software systems within their social contexts can we understand the complex dynamics of human–machine interaction.

Appendix B: Methodology

I entered the field intending to investigate the relationship between organizational culture and employee commitment in San Francisco's youthful high-tech industry. A friend helped me gain access to AllDone by introducing me to Martin—a former high school classmate and one of AllDone's cofounders—via e-mail. Martin agreed to meet with me and following our conversation offered me an unpaid internship. I would come to the office 1 day per week and assist Martin with marketing projects in exchange for research access. Within a month, Martin proposed that I take a part-time, paid position. As I began an assignment involving members of AllDone's remote team in the Philippines, the focus of my research extended to include the human–machine systems that I quickly discovered were integral to AllDone's operations. A few months later, senior executives and I agreed that I would take on a full-time role for 1 year while continuing my research activities. I became AllDone's director of customer support and operations manager, reporting to Carter, AllDone's president. In this role, I interacted frequently with leaders and team members across the organization. I participated in three weekly meetings with ADSF principals and held multiple weekly calls and videoconferences with ADP and ADLV leaders; I also traveled to the Philippines three times and to Las Vegas on nine occasions to meet with team leaders and employees. In addition, I reviewed thousands of documents and e-mails during and after my tenure with the company.

Engaging in systematic and sustained participant-observation research within an organization operating on the frontiers of the digital economy allowed me to observe the complementarities that emerged between workers and innovative software systems. I employed a longitudinal research design (Fine, Morrill, & Surianarain, 2009) to observe how the relationship between work and technology unfolded over time within a dynamic organizational context. Whereas most studies of work and technological change examine how the introduction of new machinery affects the organization and execution of work, I investigate how configurations of software and workers were transformed as an organization adapted to shifting pressures from investors, competitors, and users.

I recorded extensive jottings (Emerson, Fretz, & Shaw, 2011) while I was in the field, usually on a work computer as events were occurring, sometimes on a mobile phone, and occasionally in a small notebook. On my subway ride home each night, I began to turn jottings into full fieldnotes; I then reread the fieldnotes and wrote analytic commentary on the day's events. This process typically took 1 to 2 hours or longer. Reviewing and analyzing each day's fieldnotes helped me identify emergent patterns in the data, link data to concepts and themes, integrate insights, and formulate questions to investigate in subsequent fieldwork. Systematically engaging in such mental activities allows researchers who are personally involved in their field sites to sustain a "professional distance" essential to generating insights from data (Anteby, 2013). After leaving the field as a full-time employee, I continued to gather data by conducting informal interviews with informants across the organization and examining public sources.

Acknowledgments

The author gratefully acknowledges Michael Burawoy, Stephen Barley, Herbert Docena, Martin Eiermann, Fidan Elcioglu, Neil Fligstein, Marion Fourcade, Andrew Jaeger, Shreeharsh Kelkar, Zachary Levenson, Katherine Maich, Christopher Muller, Josh Seim, Shelly Steward, Jonah Stuart Brundage, and Kim Voss for comments on previous versions of this article. The article also benefitted from the feedback of participants in the UC Berkeley Center for Culture, Organization, and Politics workshop and the 2015–2016 Berkeley Connect workshop, and from comments provided by two anonymous reviewers.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This project was supported in part by a grant from the Horowitz Foundation for Social Policy.

Notes

1. The company's name and the names of individuals have been changed.
2. When provided the clue, "its largest airport was named for a World War II hero; its second largest, for a World War II battle" under the category "U.S. Cities," Watson's answer was Toronto, a Canadian metropolis.
3. Each of AllDone's three work sites was demographically distinct. All ADSF staffers were educated at selective or elite colleges, almost all were in their 20s, and all but two were male. Most members of ADP were college educated and between 20 and 40 years old, and more than two thirds were women. ADLV's staff was almost exclusively female; most were middle-aged and did not have college degrees.
4. AllDone's distributed workforce was used to perform a variety of other nonroutine tasks to complement software. These included reviewing seller profiles and removing those that violated AllDone's community guidelines, as well as proofreading and editing the text that sellers wrote on their profile pages to ensure they met a minimum editorial standard of professionalism.
5. AllDone's software engineers not only experimented with code but also experimented with labor to increase the pace of production. The nature of software work is altered when engineers have access to computational labor: Developers who outsource the most tedious tasks are able to innovate more quickly, making their work more "creative," both symbolically and, arguably, in practice (Irani, 2015b).
6. Other instances of reverse substitution included the following: screening out sellers who appeared on the Department of Justice's national sex offender

registry; adding badges to profiles that passed a series of verifications; checking sellers' professional license numbers against relevant state databases; running voluntary criminal background checks; and sending personalized e-mails apologizing to buyers whose requests received zero quotes from sellers.

7. Some Uber drivers, for example, become frustrated by the lack of transparency into the algorithms that assign passengers to drivers. In response, drivers attempt to decipher and then "game" the system by developing alternative behavioral strategies that subvert the patterns imposed by Uber's software (Lee et al., 2015).
8. For example, the following description of a seller's services contains two phrases (shown here in italics) that buyers commonly include in Web searches: "Meghan Traynor is a passionate *professional wedding photographer* who uses an artistic documentary style in *taking wedding photos*. She also does lifestyle portraits touched with her signature editing effects."
9. Each new seller signup helped to generate new consumer requests by expanding the content on AllDone's Web pages. ADP writers could pen keyword-rich descriptions of each new seller, which when posted to the website would help to increase search traffic to AllDone's pages, thereby producing more consumer requests. More requests meant more e-mails to send to prospective sellers, which would lead to still more signups in a virtuous cycle.
10. Some ADLV team members were so devoted to clients that they would transgress company directives. Employees frequently provided sellers with their personal phone extensions even though managers advised them not to so that sellers with whom agents had established relationships could reach them directly the next time a problem arose.

References

Alpaydin, E. (2014). *Introduction to machine learning* (3rd ed.). Cambridge, MA: MIT Press.

Anteby, M. (2013). Relaxing the taboo on telling our own stories: Upholding professional distance and personal involvement. *Organization Science*, *24*(4), 1277–1290.

Arntz, M., Gregory, T., & Zierahn, U. (2016). The risk of automation for jobs in OECD countries: A comparative analysis (OECD Social, Employment and Migration Working Papers No. 189). Paris, France: OECD Publishing.

Aronowitz, S., & DiFazio, W. (1994). *The jobless future: Sci-tech and the dogma of work*. Minneapolis, MN: University of Minnesota Press.

Autor, D. (2014). Polanyi's Paradox and the shape of employment growth (National Bureau of Economic Research Working Paper No. 20485). Cambridge, MA: National Bureau of Economic Research.

Autor, D. H. (2015a). Why are there still so many jobs? The history and future of workplace automation. *Journal of Economic Perspectives*, *29*(3), 3–30.

Autor, D. H. (2015b). Paradox of abundance: Automation anxiety returns. In S. Rangan (Ed.), *Performance and progress: Essays on capitalism, business, and society* (pp. 237–260). New York, NY: Oxford University Press.

Autor, D. H., Levy, F., & Murnane, R. J. (2003). The skill content of recent technological change: An empirical exploration. *Quarterly Journal of Economics*, *118*(4), 1279–1333.

Bandelj, N. (2009). Emotions in economic action and interaction. *Theory and Society*, *38*, 347–366.

Barley, S. R. (1986). Technology as an occasion for structuring: Evidence from observations of CT scanners and the social order of radiology departments. *Administrative Science Quarterly*, *31*(1), 78–108.

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

Beckert, J. (2006). Trust and markets. In R. Bachmann & A. Zaheer (Eds.), *Handbook of trust research* (pp. 318–331). Northampton, MA: Edward Elgar Publishing.

Benzell, S. G., Kotlikoff, L. J., LaGarda, G., & Sachs, J. D. (2015). Robots are us: Some economics of human replacement (National Bureau of Economic Research Working Paper No. 20941). Cambridge, MA: National Bureau of Economic Research.

Bergvall-Kåreborn, B., & Howcroft, D. (2014). Amazon Mechanical Turk and the commodification of labour. *New Technology, Work and Employment*, *29*(3), 213–223.

Bessen, J. (2015a). Toil and technology. *Finance & Development*, *52*(1), 16–19.

Bessen, J. (2015b). *Learning by doing: The real connection between innovation, wages, and wealth*. New Haven, CT: Yale University Press.

Bhidé, A. V. (2000). *The origin and evolution of new businesses*. New York, NY: Oxford University Press.

Both, G. (2014, April 3). What drives research in self-driving cars? Part 2: Surprisingly not machine learning. *CASTAC*. Retrieved from http://blog.castac.org/2014/04/03/what-drives-research-in-self-driving-cars-part-2-surprisingly-not-machine-learning/

Brick, H. (2000). *Age of contradiction: American thought and culture in the 1960s*. Ithaca, NY: Cornell University Press.

Brotheridge, C., & Grandey, A. (2002). Emotional labor and burnout: Comparing two perspectives of "people work.". *Journal of Vocational Behavior*, *60*, 17–39.

Brynjolfsson, E., & McAfee, A. (2014). *The second machine age: Work, progress, and prosperity in a time of brilliant technologies*. New York, NY: W.W. Norton.

Burrell, J. (2016). How the machine "thinks": Understanding opacity in machine learning algorithms. *Big Data & Society*, *3*(1), 1–12.

Cook, K., Cheshire, C., Gerbasi, A., & Aven, B. (2009). Assessing trustworthiness in providers. In K. Cook, C. Snijders, V. Buskens & C. Cheshire (Eds.), *eTrust: Forming relationships in the online world* (pp. 189–214). New York, NY: Russell Sage.

Davenport, T. H., & Kirby, J. (2016). *Only humans need apply: Winners and losers in the age of smart machines*. New York, NY: Harper Business.

Dreyfus, H. (1992). *What computers still can't do: A critique of artificial reason*. Cambridge, MA: MIT Press.

Ekbia, H. R. (2008). *Artificial dreams: The quest for non-biological intelligence*. New York, NY: Cambridge University Press.

Ekbia, H. R. (2015). Heteronomous humans and autonomous agents: Toward artificial relational intelligence. In J. Romportl, E. Zackova & J. Kelemen (Eds.), *Beyond artificial intelligence: The disappearing human-machine divide* (pp. 63–78). New York, NY: Springer.

Ekbia, H. R., & Nardi, B. A. (2017). *Heteromation and other stories of computing and capitalism*. Cambridge, MA: MIT Press.

Elliott, S. W. (2014). Anticipating a luddite revival. *Issues in Science and Technology*, *30*(3), 27–36.

Emerson, R. M., Fretz, R. I., & Shaw, L. L. (2011). *Writing ethnographic fieldnotes* (2nd ed.). Chicago, IL: University of Chicago Press.

Fine, G. A., Morrill, C., & Surianarain, S. (2009). Ethnography in organizational settings. In D. A. Buchanan & A. Bryman (Eds.), *The SAGE handbook of organizational research methods* (pp. 602–619). Thousand Oaks, CA: Sage.

Ford, M. (2015). *Rise of the robots: Technology and the threat of a jobless future*. New York, NY: Basic Books.

Frey, C. B., & Osborne, M. A. (2017). The future of employment: How susceptible are jobs to computerisation? *Technological Forecasting and Social Change*, *114*, 254–280.

Gershenson, C. (2013). Information and computation. In P. Michelucci (Ed.), *Handbook of human computation* (pp. 61–70). New York, NY: Springer.

Gillespie, T. (2016). Algorithm. In B. Peters (Ed.), *Digital keywords: A vocabulary of information society and culture* (pp. 18–30). Princeton, NJ: Princeton University Press.

Goel, V. (2017, May 3). Facebook scrambles to police content amid rapid growth. *The New York Times*. Retrieved from https://www.nytimes.com/2017/05/03/technology/facebook-moderators-q1-earnings.html

Goldberg, K. (2015). Countering singularity sensationalism. *Nature*, *526*, 320–321.

Goldberg, K. (2017, January 1). Re: A.I. *New York Times Magazine*, 7.

Gompers, P., & Lerner, J. (2004). *The venture capital cycle* (2nd ed.). Cambridge, MA: MIT Press.

Hochschild, A. (1983). *The managed heart: Commercialization of human feeling*. Berkeley, CA: University of California Press.

Hopkins, N. (2017, April 21). Revealed: Facebook's internal rulebook on sex, terrorism, and violence. *The Guardian*. Retrieved from https://www.theguardian.com/news/2017/may/21/revealed-facebook-internal-rulebook-sex-terrorism-violence

Howcroft, D., & Taylor, P. (2014). "Plus ca change, plus la meme chose?"—Researching and theorising the "new" new technologies. *New Technology, Work and Employment*, *29*(1), 1–8.

Irani, L. (2015a). The cultural work of microwork. *New Media & Society*, *17*(5), 720–739.
Irani, L. (2015b). Difference and dependence among digital workers: The case of Amazon Mechanical Turk. *South Atlantic Quarterly*, *114*(1), 225–234.
Irani, L. (2015c, January 15). ‘‘Justice for ‘data janitors’.’’ *Public Books*. Retrieved from http://www.publicbooks.org/nonfiction/justice-for-data-janitors/
Kaplan, J. (2015). *Humans need not apply: A guide to wealth and work in the age of artificial intelligence*. New Haven, CT: Yale University Press.
Kenney, M., & Florida, R. (2000). Venture capital in Silicon Valley: Fueling new firm formation. In M. Kenney (Ed.), *Understanding Silicon Valley: The anatomy of an entrepreneurial region* (pp. 98–123). Stanford, CA: Stanford University Press.
Keynes, J. M. (1963). Economic possibilities for our grandchildren. In *Essays in persuasion* (pp. 358–373). New York, NY: W.W. Norton.
Kingsley, S. C., Gray, M. L., & Suri, S. (2015). Accounting for market frictions and power asymmetries in online labor markets. *Policy & Internet*, *7*(4), 383–400.
Lee, M. K., Kusbit, D., Metsky, E., & Dabbish, L. (2015). Working with machines: The impact of algorithmic and data-driven management on human workers. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 1603–1612.
Liker, J. K., Haddad, C. J., & Karlin, J. (1999). Perspectives on technology and work organization. *Annual Review of Sociology*, *25*, 575–596.
Lim, B. Y., Dey, A. K., & Avrahami, D. (2009). Why and why not explanations improve the intelligibility of context-aware intelligent systems. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2119–2128.
MacKenzie, D., & Wajcman, J. (1999). The technology of production: Introduction. In D. MacKenzie & J. Wajcman (Eds.), *The social shaping of technology* (pp. 141–151). Philadelphia, PA: Open University Press.
Madrigal, A. C. (2012, September 6). How Google builds its maps—And what it means for the future of everything. *The Atlantic*. Retrieved from https://www.theatlantic.com/technology/archive/2012/09/how-google-builds-its-maps-and-what-it-means-for-the-future-of-everything/261913/
Madrigal, A. C. (2014, January 2). How Netflix reverse engineered Hollywood. *The Atlantic*. Retrieved from https://www.theatlantic.com/technology/archive/2014/01/how-netflix-reverse-engineered-hollywood/282679/
Michelucci, P. (Ed.). (2013). *Handbook of human computation*. New York, NY: Springer.
Miller, C. C. (2015, June 25). Can an algorithm hire better than a human? *The New York Times*. Retrieved from http://www.nytimes.com/2015/06/26/upshot/can-an-algorithm-hire-better-than-a-human.html
Mindell, D. (2002). *Between human and machine: Feedback, control, and computing before cybernetics*. Baltimore, MD: Johns Hopkins University Press.
Mindell, D. A. (2015). *Our robots, ourselves: Robotics and the myths of autonomy*. New York, NY: Viking.

Mokyr, J., Vickers, C., & Ziebarth, N. L. (2015). The history of technological anxiety and the future of economic growth: Is this time different? *Journal of Economic Perspectives*, *29*(3), 31–50.

Nichols, T., & Beynon, H. (1977). *Living with capitalism: Class relations and the modern factory*. Boston, MA: Routledge.

Noble, D. F. (1995). *Progress without people: New technology, unemployment, and the message of resistance*. Toronto, Canada: Between the Lines.

Noble, D. F. (1999). Social choice in machine design: The case of automatically controlled machine tools. In D. MacKenzie & J. Wajcman (Eds.), *The social shaping of technology* (pp. 161–176). Philadelphia, PA: Open University Press.

Polanyi, M. (1966). *The tacit dimension*. New York, NY: Doubleday.

Popper, N. (2016, February 25). The robots are coming for Wall Street. *New York Times Magazine*. Retrieved from http://www.nytimes.com/2016/02/28/magazine/the-robots-are-coming-for-wall-street.html

Pratt, G. A. (2015). Is a Cambrian explosion coming for robotics?. *Journal of Economic Perspectives*, *29*(3), 51–60.

Raval, N., & Dourish, P. (2016). Standing out from the crowd: Emotional labor, body labor, and temporal labor in ridesharing. *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, 97–107.

Rifkin, J. (1995). *The end of work: The decline of the global labor force and the dawn of the post-market era*. New York, NY: Putnam.

Roberts, S. T. (2016). Commercial content moderation: Digital laborers' dirty work. In S. U. Noble & B. M. Tynes (Eds.), *The intersectional internet: Race, sex, class and culture online* (pp. 147–160). New York, NY: Peter Lang.

Romportl, J. (2015). Naturalness of artificial intelligence. In J. Romportl, E. Zackova & J. Kelemen (Eds.), *Beyond artificial intelligence: The disappearing human-machine divide* (pp. 211–216). New York, NY: Springer.

Russell, S., & Norvig, P. (2010). *Artificial intelligence: A modern approach* (3rd ed.). San Francisco, CA: Prentice Hall.

Sachs, J. D., & Kotlikoff, L. J. (2012). Smart machines and long-term misery (National Bureau of Economic Research Working Paper No. 18629). Cambridge, MA: National Bureau of Economic Research.

Samuel, R. (1977). Workshop of the world: Steam power and hand technology in mid-Victorian Britain. *History Workshop*, *3*, 6–72.

Simon, H. A. (1965). *The shape of automation for men and management*. New York, NY: Harper & Row.

Smith, S. V. (2015, May 20). An NPR reporter raced a machine to write a news story. Who won? *Planet Money*. Retrieved from http://www.npr.org/sections/money/2015/05/20/406484294/an-npr-reporter-raced-a-machine-to-write-a-news-story-who-won

Srnicek, N., & Williams, A. (2015). *Inventing the future: Postcapitalism and a world without work*. New York, NY: Verso.

Stovel, K., & Shaw, L. (2012). Brokerage. *Annual Review of Sociology*, *38*, 139–158.

Suchman, L. (2007). *Human-machine reconfigurations: Plans and situated actions* (2nd ed.). New York, NY: Cambridge University Press.

Tufecki, Z. (2015, April 18). The machines are coming. *New York Times*. Retrieved from http://www.nytimes.com/2015/04/19/opinion/sunday/the-machines-are-coming.html

Wajcman, J. (2006). New connections: Social studies of science and technology and studies of work. *Work, Employment & Society*, *20*(4), 773–786.

Walker, C. R. (1958). Life in the automatic factory. *Harvard Business Review*, *36*(1), 111–119.

Weeks, K. (2007). Life within and against work: Affective labor, feminist critique, and post-Fordist politics. *Ephemera: Theory and Politics in Organization*, *7*(1), 233–249.

Wharton, A. S. (2009). The sociology of emotional labor. *Annual Review of Sociology*, *35*, 147–165.

Winner, L. (1977). *Autonomous technology: Technics-out-of-control as a theme in political thought*. Cambridge, MA: MIT Press.

Winograd, T., & Flores, F. (1987). *Understanding computers and cognition: A new foundation for design*. Boston, MA: Addison-Wesley.

Wise, J. M. (1998). Intelligent agency. *Cultural Studies*, *12*(3), 410–428.

Yang, R., & Newman, M. W. (2013). Learning from a learning thermostat: Lessons for intelligent systems for the home. *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 93–102.

Zackova, E. (2015). Intelligence explosion quest for humankind. In J. Romportl, E. Zackova & J. Kelemen (Eds.), *Beyond artificial intelligence: The disappearing human-machine divide* (pp. 31–44). New York, NY: Springer.

Zider, B. (1998). How venture capital works. *Harvard Business Review*, *76*(6), 131–139.

Ziewitz, M. (2016). Governing algorithms: Myth, mess, and methods. *Science, Technology, & Human Values*, *41*(1), 3–16.

Zuboff, S. (1988). *In the age of the smart machine: The future of work and power*. New York, NY: Basic Books.

Author Biography

Benjamin Shestakofsky is a PhD candidate in the Department of Sociology at the University of California, Berkeley. His research centers on how digital technologies are affecting work and employment, organizations, and economic exchange.