
Article:
Miltner, K.M. orcid.org/0000-0001-6964-1023 (2019) Girls who coded: gender in twentieth century U.K. and U.S. computing. Science, Technology, & Human Values, 44 (1). pp. 161-176. ISSN 1552-8251

https://doi.org/10.1177/0162243918770287

Girls Who Coded: Gender in 20th Century U.K and U.S. Computing (Review Essay)

Hicks, M. (2017). *Programmed inequality: How Britain discarded women technologists and lost its edge in computing*. MIT Press.

Abbate, J. (2012). *Recoding gender: women's changing participation in computing*. MIT Press.

Ensmenger, N. L. (2012). *The computer boys take over: Computers, programmers, and the politics of technical expertise*. MIT Press.

In May 2017, the *Wall Street Journal* caused a commotion when it reported that women software engineers at Facebook were 35% more likely to have their code rejected by the company's internal peer review system (Seetharaman 2017). The results of the original study, conducted by a former Facebook employee during her tenure at the company, suggested that the rejection rate implied more intense scrutiny when it came to the code of women engineers. However, a follow-up study carried out by Facebook's head of infrastructure concluded that while women were receiving higher rates of code rejection and notes, the issue was connected to rank, not gender. While Facebook acknowledged that "the current representation of senior female engineers both at Facebook and across the industry is nowhere near where it needs to be" (Statt 2017), it denied that gender was a causal factor in the uneven rates of code rejection. Instead, Facebook chastised its employees for leaking the information, arguing that such stories damage Facebook's "recruiting brand" and makes it harder to hire women (Wong 2017).

The issue of gender discrimination in the technology industry is a hot topic these days, with journalists, policymakers, tech executives, scholars, and others decrying the exclusion of women and girls from an industry that has been positioned as "a bright spot in an otherwise dull economy" (Wagstaff 2012) and "a ticket to economic salvation for the masses" (Farag 2016). The assumption underlying most attempts to fix tech's "women problem" (Bloom 2017) is that there is a 'pipeline' issue. This assumes—for a variety of reasons ranging from discouragement to unappealing stereotypes—that girls and women are uninterested in tech careers or lack the skills to get in the door. Most of the solutions to address this purported problem focus on teaching girls and women to code, with the expectation that these programming skills will be the key to self-determination in the contemporary digital economy.

However, as Marie Hicks (2017) argues in *Programmed Inequality: How Britain Discarded Its Women Technologists and Lost Its Edge In Computing*, "initiatives to get girls, women, and people of color to train for STEM jobs cannot undo the underlying structures of power that have been designed into technological systems over the course of decades" (p. 325). The close relationship between masculinity and computing has been explored in a variety of contexts, (e.g., Kendall 2000, Eglash 2002), but in drawing attention to the gendered power dynamics associated with technological practice and systems, Hicks gestures toward a literature that examines the imbrication of gender and technology more broadly. As Lerman, Oldenziel, and Mohun point out, "gender analysis illuminates our understandings of technology, and attention to technology illuminates our understandings of gender" (2003, p. 5), and the co-constitutive relationship between gender and technology can be seen on multiple levels ranging from individual identity to

institutional and structural relations. Unfortunately, as the work of Rossiter, (1982), Cockburn (1985), Oldenziel (1999), and others have illustrated, definitions of technological skill and expertise have historically been constructed in such a way that privileges the masculine, rendering the feminine as "incompatible with technological pursuits" (Wajcman 2006).

Computing is a discipline that is very clearly framed as the "natural" domain of men and the masculine; in late 2017, an article in progressive news outlet The Guardian queried whether or not there was "an underlying biological explanation" as to why Google's engineering workforce is only 20% women, given its "progressive ideals and family-friendly ethos" (Devlin and Hern 2017). Indeed, the employment statistics of Silicon Valley corporations are often used as evidence that men are somehow more suited to computing work than women. However, several recent monographs on the history of computing contradict this ahistorical conceptualization. Building upon the work of Sadie Plant (1997) and Jennifer Light (1999), Janet Abbate's (2012) *Recoding Gender*, Nathan Ensmenger's (2012) *The Computer Boys Take Over,* and Hicks's (2017) *Programmed Inequality* show how computer programming was originally the purview of women. For the first two decades of the modern computing era, programming was originally associated with (and predominantly done by) women; however, as programming shifted from a low-status, feminized task to work that was seen as central to control of corporate and state resources, women were edged out.

Each of these books focus on the latter half of the 20th century, starting with Allied computing efforts in WWII and progressing through the 'computer revolution' of the 1950s, 60s and 70s. While other histories of computing have tended to focus on specific individuals (e.g., Bernhardt 2016), technologies, (e.g., Gazzard 2016) and institutions (e.g., Pugh 1995), Hicks, Abbate, and Ensmenger center their analyses on computing labor, recognizing that labor is "the necessary connective between the political, economic, and technical elements of computing history" (Hicks 2017, p. 5). They also understand, in a traditionally Science and Technology Studies (STS) fashion, that histories of technology are inextricably histories of social relations, and that per Star (1991), things always could have been otherwise. As Ensmenger (2012) argues, "there is never a single, ideal type toward which any given technology will inevitably evolve" and that technological development is shaped by those with the power to set technological and economic priorities (p. 26). In a similar vein, Hicks argues that not only do the "less tangible components of computing systems" shape priorities and possibilities at a particular moment, but that those decisions continue to reverberate for decades afterwards in a social form of path dependence (2017, p.5).

Ensmenger posits that "nowhere are the social dimensions of technological development more apparent than in the history of computing", given that "specific technologies are developed to solve specific problems, for specific users, in specific times and places" (2010, p. 26). This argument is one that Hicks and Abbate also make, but their analyses hinge on the assertion that gender and sexuality specifically play a crucially formative role in the history of computing. Echoing Cockburn's argument that "people are gendered by the jobs they do and in turn jobs are actively gendered by virtue of who does them" (2009, p. 271), Abbate illustrates how "masculinity and femininity were part of the cultural vocabulary that was used to define what a computer was and who was best qualified to use one" (2012, p. 4). All three of these histories rest on the assumption that labor—and gendered labor, specifically—made computing what it is today, far

more so than the hardware or exceptional individuals that are often the focus of computing histories.

Hicks in particular avoids narratives centered on individual women, and instead looks at women in the British computing industry as a class of workers. They[1] do so for a variety of reasons; first, to rethink computing narratives that position individuality and innovation as central; secondly, to avoid the endorsement of an "ahistorical fiction of technological meritocracy" that ignores the ways that the gender, class, nationality, sexuality, and race of technological workers influence how their work is perceived and valued; and third, to avoid technological boosterism and the lionization of computing skill that is often attached to histories that aim to "unearth women's contributions to computing" (p. 17). While these decisions undoubtedly strengthen Hicks's excellent work, the most essential contribution of this class-based analysis is that it allows Hicks to offer an incisive structural critique of the ultimately self-defeating policies put in place by the British technocrats of the mid-to-late 20th century.

The core argument of *Programmed Inequality* is that computerization is "an explicitly hegemonic project" that often helps certain groups of people consolidate power at the expense of others, and that in the British context, gender and class discrimination were at the heart of this process. Unlike the United States, where computerization was shaped and driven by a variety of institutional actors, computerization in Britain was largely controlled by the state. Throughout the book, Hicks deftly shows how British computing was dependent on the creation of a gendered "technological underclass" whose membership was determined by the British societal norms of the mid-twentieth century. Despite the fact that the sexist labor patterns instituted by the government often had negative economic consequences and ran counter to the state's overall desire to modernize, the government remained insistent upon maintaining gendered labor segregation. While a less insightful scholar might assume that this structural discrimination was an unfortunate side effect of the regressive gender politics of the mid-twentieth century, Hicks makes a different and more powerful argument: that the deskilling and feminization of women's labor was at the core of computerization in Britain, both enabling the uptake of computers in the post-war period and resulting in the eventual failure of the British computing industry in the 1980s.

The story of British computing starts in WWII with the codebreaking and cryptanalysis apparatus at Bletchley Park. Kept secret for decades after the war, the work conducted at Bletchley is now understood to have been crucial to the Allies' victory, shortening the war by at least two years and saving hundreds of thousands of lives (Hinsley 2001). The success of D-Day invasions at Normandy, for example, was largely made possible by the intelligence decoded by the Colossus computers—and, as Hicks points out—the women who were responsible for operating them. Popular renditions of British codebreaking efforts, such as Oscar-winner *The Imitation Game*, have placed individual codebreakers such as Alan Turing at the heart of their narratives. Through meticulous documentation, Hicks sets the historical record straight by showing how the success of the British wartime intelligence operation depended on the efforts of thousands of workers, a large majority of whom were women. Described by Churchill as "the geese who laid the golden eggs but never cackled", most of the workers at Bletchley were "Wrens", members of the Women's Royal Naval Service (WRNS).

[1] They is Hicks's preferred pronoun.

Despite the fact that the WRNS were central to the success of Britain's codebreaking efforts, the levels of skill and training that were required for their work have often been downplayed or obscured, even in the context of narratives that position them as war heroes. When the truth about the Colossus computers began to emerge in the 1970s, the work of the WRNS was assumed to be low-level support work instead of integral to the codebreaking itself. Hicks argues that the intense secrecy that surrounded wartime codebreaking efforts contributed to this devaluation, alongside the perception that because highly skilled work had been done by women, it was "somehow implicitly lower in skill and importance" (2017, p. 52). Hicks also points a finger at narratives that privilege technological artifacts over the workers who ensured their successful operation, including a recent exhibit at the London Museum of Science which described the Wrens' work as passively "tending" the machines. In an even more egregious example of erasure, Hicks cites the failure of the Bletchley Park Historical Site to identify and credit Dorothy Du Boisson and Elsie Booker in the display of the only surviving picture of a Colossus being operated.

Hicks contends that the "double-edged sword of feminization" was both a help and a hindrance to early British computing since it "defined understandings of how to structure and deploy large-scale computational projects", particularly in the British Civil Service (2017, p. 52). Such understandings rested on labor divisions that were aimed at keeping government costs low and rooted in gendered expectations of women's role in the workplace. In the immediate postwar period, the Treasury reorganized the Civil Service to create a new class of low-wage machine workers who were invariably women. When equal pay legislation was passed in the late 1950s, the machine operator classes were excluded, since there was no comparable class of jobs occupied by men. This created a formal division between the managerial classes and the workers responsible for the calculation and tabulation work that the expanding welfare state was increasingly reliant upon.

While purportedly sensible from a short term, cost-savings standpoint, Hicks reveals how the structure that this gendered labor division created ended up hampering the government's shift towards modernization and automation, since it "presaged a postindustrial order in which gender and automation were interdependent categories" (2017, p. 121). As computer jobs became key symbols of industrial and social modernity in the 1960s, a "prestige gap" developed between the underclass of information workers who performed computing work and the value that was accorded to that work. Despite the fact that there was a perceived shortage of skilled technical labor to run the government's increasingly numerous computing projects, the large pool of technically skilled women in the Civil Service were ignored in favor of the largely male Executive Classes who had no technical training at all. Hicks adroitly demonstrates the disastrous consequences this had not only for the British government, but for the nation as a whole.

While Prime Minister Harold Wilson's famed "White Heat" speech promised a social revolution predicated on a technical one, Hicks convincingly demonstrates that the Labour party's ambitious vision failed to come to fruition because it relied upon the continued functioning of "ingrained, antimeritocratic hierarchies" based on employment practices that foreclosed social mobility for women in particular. These hierarchies reverberated through a variety of industries in post-War Britain, but had particularly acute consequences for the British computing industry, which collapsed in the 1980s. Hicks locates the failure of ICL, Britain's state-run computing

company, in the government's decades-long policy decisions aimed at centralizing technological control. Instead of engineering a labor force that would have enabled ICL to offer a broader range of product options (among other things), the British government's narrow focus on their technocratic goals hampered the economic growth they hoped would come with modernization.

Robustly researched and written in a clear and compelling voice, *Programmed Inequality* offers an original and convincing analysis of the role gender discrimination played in the British computer revolution. Hicks offers a trenchant critique of how technological progress often goes hand-in-hand with oppressive outcomes that run counter to economic justice and social progress. Drawing parallels between the industrial and technological revolutions, this book provides another example of how the deskilling, rationalization, and feminization of labor contributes to modernization processes that promote social and economic stratification. It also provides a cautionary tale for those who are invested in "a fiction of progress through technology alone" (p. 286), illustrating the social and economic effects that particular fiction can introduce.

Recoding Gender also tells a story of women in computing from World War II until the end of the 20th century. It covers some of the same ground as *Programmed Inequality* in that it recounts the experiences of British women during that period, but it also includes the experience of American women. Like Hicks, Abbate comes to the conclusion that gender played a heretofore unacknowledged and central role in the history of computing; she also asserts that women's contributions to computing were shaped by gendered assumptions about technical skill, cultural expectations regarding paid employment, and the shifting importance of computing in Anglo-American society. However, this is largely where the similarities between the two books end. Where *Programmed Inequality* offers a scathing critique of the structural forces that deprived British women of meaningful technological careers while simultaneously hamstringing the British state and computing industry, Abbate insists that *Recoding Gender* is "not a story of oppression and failure" (p. 2). Instead, Abbate highlights "the bold and creative strategies of women who loved computing work, excelled at it, and forged successful careers" (p. 2). This decision is rooted in a well-intentioned desire to acknowledge the agency of her research subjects and provide fodder for contemporary policy interventions. However, with her silver-lining tendency to highlight the successes of individual women, Abbate's analysis ends up falling short of the structural critique that makes *Programmed Inequality* so powerful, and occasionally ends up reproducing some of the contemporary discourses that dominate current discussions of women in technology.

Like *Programmed Inequality, Recoding Gender* starts in WWII, telling the story of Bletchley Park and the Wrens that operated the Colossus computers. Like Hicks, Abbate underscores the high level of skill that the Wrens needed to do their work, including mechanical aptitude, memorization, mental math skills, and physical endurance. Also included in this wartime history is the story of ENIAC, a computer built at the University of Pennsylvania to help with calculations related to ballistic missile trajectories. ENIAC wasn't completed until after the war was over, so its impact was relegated to the postwar computing industry and the Cold War; one of the first major ENIAC projects was generating modeling for thermonuclear bombs. While John Mauchly and John Presper Eckert are given most of the credit for ENIAC since they were responsible for its hardware, Abbate points out that the ENIAC would have been useless without its six-woman programming team: Jean Jennings, Betty Snyder, Frances Bilas, Kay McNulty, Marlyn Wescoff, and Ruth Lichterman. Abbate shows how women working on the Colossus and

ENIAC projects had similar experiences in the sense that they were underestimated by their project leaders and relegated to positions that were thought to be rote and unskilled. However, there were several major differences between the American and British projects that influenced the postwar prospects of the women who worked on them, including divisions of labor, training models, and levels of secrecy. The Colossus computers were literally dismantled after the war, and the women who worked on them sworn to a lifetime of secrecy thanks to the provisions of the Official Secrets Act. On the other hand, the ENIAC women had the option to continue working, and many of them did, with some going on to distinguished careers in computing.

Despite these success stories, American women as a whole went on to face a highly discriminatory workplace in the postwar era, albeit not in the rigidly institutionalized manner found in the British context. Instead of relegating women to specific, low-wage job classes in government like the British did, Abbate shows how employers in the United States relied on socially constructed conceptualizations of (and proxies for) skill to keep women and other marginalized groups out of higher-wage and management-level positions in computing firms. Citing Cockburn (1985), she lays out how social judgments determine what capacities are seen as necessary for any particular job:

> Skill is a social construct: neither the skills required to do a job nor the skill possessed by an individual can be defined in purely objective terms. Although it is possible to identify physical, intellectual, or social abilities that are relevant to particular jobs, there is never only one way to define these requirements. Other abilities can be substituted, work processes can be reorganized, different tools can be used, and varying criteria can be applied to job performance. (p. 40)

Abbate then goes on to show how American corporations used mathematical training, a college education, and aptitude training as proxies for technical skill, all of which had different outcomes for American women looking to enter the computer industry. She also argues how efforts to improve software production in the 1960s—known as the "software crisis"—were social contestations that reflected ideas about labor and gender. These debates over programming methods and competing visions for the future of the field were fundamentally struggles over the professional identity of the programmer, struggles that also undergirded the movement to redefine programming as "software engineering" in the late 1960s and early 1970s.

The first three chapters of *Recoding Gender* have a fair amount in common with *Programmed Inequality* in the sense that they make similar, structurally-based arguments about the forces that shaped the computing industry and hampered its women workers. However, where Hicks actively avoids the uncritical reification of technological skill (and the tech boosterism that accompanies it), Abbate often plays into those tropes, particularly in discussions of Grace Murray Hopper and her trailblazing accomplishments. Overall, the last two chapters of the book highlight the accomplishments of individual women in the business and academic worlds in a manner that diminishes the structural challenges faced by women in computing, and at times seems to applaud a bootstraps approach that takes on a vaguely neoliberal flavor.

Chapter 4 tells the stories of Dame Stephanie (Steve) Shirley and Elsie Shutt, two computing entrepreneurs in Britain (Shirley) and the United States (Shutt). While they were not

the only woman entrepreneurs in the postwar era (Dina Vaughan is another entrepreneur who was briefly mentioned), Shirley and Shutt made their mark—and their money— by deploying a business model that leveraged the underutilized talents of women programmers who were excluded from the job market after they had children. By letting these women work from home on a part-time basis, Shirley and Shutt built successful businesses by tapping a skilled labor pool that was being ignored for purely sexist reasons. Shirley and Shutt's stories are undoubtedly inspiring and deserving of recognition, but in celebrating their successes, Abbate plays up their exceptionalism and downplays the structural disadvantages that women in computing faced on the whole. When discussing Shutt and Shirley's business strategies, Abbate states that they offered "high quality work for low prices" without any critical discussion of the fact that they had to actively devalue their work to get a foot in the door. Also lacking is a satisfying analysis of the affective labor that Shutt and Shirley had to regularly perform, downplaying their femininity in order to appear "professional" and minimizing any masculinity to avoid appearing "unnatural" (pp. 132-33). Furthermore, this chapter is missing any real examination of race or class privilege; Abbate treats the material and emotional support that Shutt and Shirley received from their husbands as an afterthought. While celebrating alternative business models that take mothers seriously is certainly a worthwhile endeavor, stating that Shutt and Shirley "chose to innovate their way around discrimination" (p. 140) frames the transcendence of major structural barriers as an issue of individual responsibility.

Chapter 5 shifts to a discussion of academic computer science and runs into some of the same issues present in Chapter 4. While this chapter goes into a fair amount of detail regarding the obstacles that female scholars have faced in this arena, Abbate instead chooses to focus on "their resourceful strategies for gaining credentials, finding alternative ladders to visibility and career advancement and creating their own professional forums and events" (p. 9). Abbate does this in a well-intentioned attempt to offer a starting point for changing a gendered computing culture, but she does so in a way that seems to overenthusiastically—and even naively— position the individual achievements of exceptional women as a possible way forward for women in the field more generally. A section on "Alternative Paths to Success in Academia" relates a series of outlier stories—most notably women without PhDs getting tenure-track jobs— and then offers them up as novel routes to success without appropriately acknowledging the unique and privileged circumstances that facilitated these outcomes. Similarly, Abbate spends an entire chapter section lauding professional societies as venues for advancement, but then finishes up by acknowledging that their overall effect on the gender imbalance in computing was limited. She closes the chapter by exclaiming that "a wide range of women" were able to make computer science their "intellectual and professional home" thanks to their "determination and ingenuity" (p. 175). This very well may be the case, and I have no wish to diminish the accomplishments of women who succeeded against considerable odds. Nonetheless, to paraphrase Audre Lorde, Abbate's approach involves dismantling the master's house with the master's tools, and positioning these exceptional narratives as potential starting points for transformation forecloses a more critical analysis that could help identify more structurally-oriented avenues for change.

Nathan Ensmenger's *The Computer Boys Take Over* is not exclusively a history of women in computing, but it is a story about gender; the titular reference to the "computer boys" necessarily means that Ensmenger deals with who those computer boys were, and the process by which they replaced computer girls as the ideal computer worker. In doing so, he often offers the structural

critique missing from *Recoding Gender*. The core conceit of this book is that the rise to dominance of computer technology is about software; Ensmenger's argument is that any discussion of the "computer revolution" is a discussion of software, because it is software that transformed "the ways in which [people] work, live, consume, recreate, and engage in social and personal relationships" (p. 6). He goes on to argue:

> Software is where the technology of computing meets social relationships, organizational politics, and personal agendas. All technologies are to a certain extent social constructions, but in the case of software, the social dimensions of technology are particularly apparent. (p. 8)

As such, Ensmenger asserts, disputes about computerization and software were really social negotiations about organizational power, authority, and professional identity. Like Hicks, Ensmenger's history traces the transformation of the computer from scientific instrument to tool for control—in this particular narrative, corporate control. As part of this transformation, he also follows the evolution of the imagined computer user, revealing a highly contested social process that ends up being rife with gendered dynamics.

Ensmenger covers a lot of the same historical territory as Hicks and Abbate, starting with ENIAC in the late 1940s and focusing on the major developments in computing over the following three decades. The first half of the book is largely concerned with chronicling the debates over what constituted "programming" and consequently, who fit the mold of the ideal programmer. Like Hicks and Abbate, Ensmenger traces the progression of computer programming from feminized, low-status clerical (and subclerical) work into a highly-valued, highly-paid, largely autonomous occupation, and in doing so documents the various negotiations and "crises" that shaped the programming profession into the masculinized domain it became (and remains). However, whereas Hicks argues that the gendered dynamics of British computing involved active, institutionalized discrimination, Ensmenger argues that the American bias in favor of male programmers was "a combination of laziness, ambiguity, and traditional male privilege" (p. 79) whose impact on women was more of a side effect than a goal. Whether a feminist historian would have come to a similar conclusion is another question, particularly since Ensmenger spends a fair amount of time contending with the belief that good programmers were "born, not made" (pp. 19, 54, 68, 81). As Abbate argues, determinations of inherent worth are necessarily social constructions, and Ensmenger alludes to the many gendered, raced, and classed assumptions incorporated in that assertion.

The second half of the book focuses on the various processes of legitimation that were initiated in the 1950s and 1960s as programmers of various stripes attempted to lay claim to the profession. Ensmenger expertly traverses the various social, disciplinary, and intellectual battles that took place as this process unfolded; the chapters on the formation of computer science as an academic discipline and the establishment of the term "software engineering" were particularly rich. Emerging from these case studies is a compelling argument about the ways in which professionalization operates as a process of distinction that not only establishes standards of quality, but excludes "undesirables and competitors" (p. 165). In an anecdote reminiscent of Tressie McMillan Cottom's (2017) brilliant work on for-profit universities, Ensmenger relates how qualifications from predatory for-profit vocational schools were used to automatically eliminate

candidates—who, of course, were mostly from marginalized groups— from employment pools. Ending the book with one of its most valuable contributions, Ensmenger deploys the STS concept of boundary work to explain how programmers utilized the "internal inconsistencies" of the field as ideological tools to accomplish their professional agenda *du jour*. Specifically, Ensmenger's programmers deployed art and science as "rhetorical resources to be used in pursuit of professional development and institution-building strategies" (p. 230); this meant that they would variably paint programming as an artistic, artisanal, scientific, or engineering pursuit depending on the aim that they were looking to achieve. Naturally, these rhetorical strategies had distinct consequences for the gendered nature of the discipline that are still visible today.

The history lessons offered by these three books are almost painfully relevant for scholars of (and practitioners in) contemporary technological fields; as Hicks points out, "histories of hidden or devalued computing labor connect powerfully with current trends in information technology and prompt questions about the categories of privilege that silently structure our computing systems today" (2017, p. 313). One clear takeaway from all of these works is how skill—and technical skill in particular—is deployed as a proxy to keep certain groups in positions of power. As Abbate argues,

> When it comes to individual workers, assessments of skill are colored by assumptions about the capacities, interests, and appropriate spheres of action of different social groups. Since technical skill conveys power—including prestige, access to well-paid employment, and the opportunity to shape the tools used by a whole society—the dominant groups in society tend to assert their "natural" superiority in these fields. (p. 40).

Edward David of the illustrious Bell Labs once made the assertion that "with the right people, all problems vanish" (Ensmenger 2012, p. 148); while David was writing in the 1950s, this discourse is still alive and well in places like Silicon Valley, who use rigid perceptions of "the right people" to exclude marginalized groups from certain job categories. In 2015, Engineering Manager Leslie Miley left Twitter because his efforts to diversify the company's engineering staff met with resistance from other members of the company's leadership. In a Medium post explaining his departure, Miley explained how arbitrary indicators of "quality" were used to avoid hiring outside of established patterns:

> There were also the Hiring Committee meetings that became contentious when I advocated for diverse candidates. Candidates who were dinged for not being fast enough to solve problems, not having internships at "strong" companies and who took too long to finish their degree. Only after hours of lobbying would they be hired. Needless to say, the majority of them performed well. Personally, a particularly low moment was having my question about what specific steps Twitter engineering was taking to increase diversity answered by the Sr. VP of Eng at the quarterly Engineering Leadership meeting. When he responded with "diversity is important, but we can't lower the bar," I then realized I was the only African-American in Eng leadership. (Miley 2015)

Returning to the Facebook case study at the start of this review, these books cast Facebook's insistence on rank as an explanation for the women engineers' code rejections in quite a different light. They also encourage critical evaluation of Learn to Code movements that target

women and people of color; as Hicks insightfully argues, "history shows that initiatives like these mean little if the participants being targeted for empowerment in a field are still disempowered in the wider world" (2017, p. 312). And indeed, the arguments put forth by these works raise the question of whether or not empowerment is the ultimate goal; recent work from Virginia Eubanks (2012), Sarah T. Roberts (2016), and Brooke Erin Duffy (2017) illustrate how invisible, poorly remunerated, and feminized labor is the bedrock upon which the technology industry rests. And, as Hicks in particular shows, there are clear consequences that come with discriminatory structures—and not just for those who are discriminated against. The technology giants of the present would do well to avoid the mistakes of the past in their focus on "the limited and myopic goals of small but powerful segments of society" (Hicks 2017, p. 17)—the likelihood of that, however, is another issue entirely.

Acknowledgments: The author would like to thank Mike Ananny, Christina Dunbar-Hester, and the two anonymous reviewers for their insightful comments on this piece.

Author Bio: Kate M. Miltner is a PhD Candidate at the USC Annenberg School for Communication and Journalism and a Joint Fellow at the UC Berkeley Center for Technology, Society and Policy and the UC Berkeley Center for Long-Term Cybersecurity.

References

Bernhardt, C. (2016). *Turing's Vision: The Birth of Computer Science*. MIT Press.

Cockburn, C. (1985). *Machinery of dominance: Women, men, and technical know-how*. Northeastern University Press.

Cockburn, C. (2009). On The Machinery of Dominance: Women, Men, and Technical Know-How. *WSQ: Women's Studies Quarterly* 37(1), 269-273.

Cottom, T. M. (2017). *Lower Ed: The Troubling Rise of For-Profit Colleges in the New Economy*. The New Press.

Devlin, H. and Hern, A. (2017, August 8). Why are there so few women in tech? The truth behind the Google memo. *The Guardian*. https://www.theguardian.com/lifeandstyle/2017/aug/08/why-are-there-so-few-women-in-tech-the-truth-behind-the-google-memo

Douglas, S. J. (1989). *Inventing American Broadcasting 1899-1922*. Johns Hopkins University Press.

Duffy, B. E. (2017*). (Not) Getting Paid to Do What You Love: Gender, Social Media, and Aspirational Work*. Yale University Press.

Eglash, R. (2002). Race, sex, and nerds: From black geeks to Asian American hipsters. *Social Text* *20*(2), 49-64.

Eubanks, V. (2012). *Digital dead end: Fighting for social justice in the information age*. MIT Press.

Farag, B. (2016, February 24). Please don't learn to code unless… *LinkedIn Pulse*. Retrieved from https://www.linkedin.com/pulse/please-dont-learn-code-unless-basel-farag?trk=hp-feed-article-title-share

Gazzard, A. (2016). *Now the chips are down: The BBC micro*. MIT Press.

Hinsley, F. H. (2001). *Codebreakers: the inside story of Bletchley Park*. Oxford University Press, USA.

Kendall, L. (2000). "Oh no! I'm a nerd!" Hegemonic masculinity on an online forum. *Gender & Society* *14*(2) 256-274.

Lerman, N., Oldenziel, R., & Mohun, A. P. (2003). Introduction: Interrogating Boundaries. In: Lerman, N., Oldenziel, R., & Mohun, A. P. (Eds.). *Gender and technology: A reader*. JHU Press.

Light, J. S. (1999). When computers were women. *Technology and culture*, *40*(3), pp. 455-483.

Miley, L. (2015, November 3). Thoughts on Diversity Part 2. Why Diversity is Difficult. *Medium*. Retrieved from https://medium.com/tech-diversity-files/thought-on-diversity-part-2-why-diversity-is-difficult-3dfd552fa1f7

Oldenziel, R. (1999). *Making Technology Masculine: Men, Women and Modern Machines in America.* Amsterdam University Press.

Plant, S. (1997). *Zeroes + Ones: Digital women + the new technoculture*. Doubleday.

Pugh, E. W. (1995). *Building IBM: shaping an industry and its technology*. MIT Press.

Roberts, S. T. (2016). Commercial Content Moderation: Digital Laborers' Dirty Work. In Noble, S.U., and Tynes, B. M. (eds). *The Intersectional Internet: Race, Sex, Class and Culture Online*. Peter Lang.

Rossiter, M. (1982). *Women Scientists in America.* Johns Hopkins Press.

Seetharaman, D. (2017, May 2). Facebook's Female Engineers Claim Gender Bias. *The Wall Street Journal.* Retrieved from https://www.wsj.com/articles/facebooks-female-engineers-claim-gender-bias-1493737116

Star, S.L. (1991). Power, technology, and the phenomenology of conventions: On being allergic to onions. In Law, J. (ed). *A Sociology of Monsters: Essays on power, technology and domination.* New York: Routledge

Statt, N. (2017, May 2). Facebook rejects female engineers' code more often, analysis finds. *The Verge.* Retrieved from: https://www.theverge.com/2017/5/2/15517302/facebook-female-engineers-gender-bias-studies-report

Wagstaff, K. (2012, July 16). Can we fix computer science education in America? *Time*. Retrieved from http://techland.time.com/2012/07/16/can-we-fix-computer-science-education-in-america/

Wong, J. C. (2017, May 2). Facebook: leaking info about gender bias damages our 'recruiting brand'. *The Guardian*. Retrieved from: https://www.theguardian.com/technology/2017/may/02/facebook-gender-bias-female-engineers-code