TAYLORED FLEXIBILITY:

AGILE, CONTROL, AND THE SOFTWARE LABOR PROCESS

by

LARISSA PETRUCCI

A DISSERTATION

Presented to the Department of Sociology
and the Division of Graduate Studies of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

September 2021

DISSERTATION APPROVAL PAGE

Student: Larissa Petrucci

Title: Taylored Flexibility: Agile, Control, and the Software Labor Process

This dissertation has been accepted and approved in partial fulfillment of the requirements for the Doctor of Philosophy degree in the Department of Sociology by:

Jill Harrison	Chairperson
Eileen Otis	Core Member
Ellen Scott	Core Member
Anita Chari	Institutional Representative

and

Andrew Karduna	Interim Vice Provost for Graduate Studies

Original approval signatures are on file with the University of Oregon Division of Graduate Studies.

Degree awarded September 2021

DISSERTATION ABSTRACT

Larissa Petrucci

Doctor of Philosophy

Department of Sociology

September 2021

Title: Taylored Flexibility: Agile, Control, and the Software Labor Process

This dissertation research examines the work arrangements of software workers in high-technology industries in order to raise questions, dispel myths, and develop a labor process theory of knowledge-based work in the 21st century. Software work is largely regarded as a "sunrise" occupation: full of opportunities for interesting work in a flexible environment. Moreover, software production, like other forms of knowledge-based work, is presumed to pose challenges to managerial control methodologies, because of employers' increasing dependence on software workers' skills and creativity and the difficulty of subjecting complex, immaterial, and cognitive work like software production to traditional methods of control. As a result, knowledge-based work appears to require new forms of control, distinct from those used in manufacturing settings.

This research, however, reveals continuities between managerial methodologies used in manufacturing-based settings and those used to organize software work through an analysis of Agile, a popular project management methodology. Agile's roots are in Toyota's lean production processes, though Agile also draws upon tenets of Taylor's scientific management as well as High-Commitment Management schemes. Drawing upon 45 interviews with workers and managers who use Agile, as well as content analysis of Agile training videos, I show how Agile aims to achieve what I call Taylored

Flexibility: an attempt to maintain flexibility to respond to the complex and turbulent nature of knowledge-based work alongside strategies to render invisible and immaterial work like software production more calculable and predictable. This dissertation also explores collective organizing strategies of software workers, emphasizing how struggles over control may not take the traditional form of conflict over pay, benefits, or the conditions of work, but of the outcomes of labor. Through this research I show that managerial strategies used to achieve Taylored Flexibility complicate common understandings about control over knowledge-based work in the new economy, showing how hybrid control regimes can operate as powerful mechanisms to render knowledge-based work more productive.

CURRICULUM VITAE

NAME OF AUTHOR: Larissa Petrucci

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene
San Francisco State University, San Francisco

DEGREES AWARDED:

Doctor of Philosophy, Sociology, 2021, University of Oregon.
Master of Science, Sociology, 2016, University of Oregon
Bachelor of Science, Sociology, 2014, San Francisco State University

AREAS OF SPECIAL INTEREST:

Work Economy and Organizations
Labor and Labor Movements
Race, Class and Gender
Public Policy

PROFESSIONAL EXPERIENCE:

Research Assistant, Labor Education & Research Center, University of Oregon, 2019-2021

Graduate Employee, Department of Sociology, University of Oregon, 2014-2019

GRANTS, AWARDS, AND HONORS:

Wasby-Johnson Dissertation Award, Department of Sociology, University of Oregon (UO), 2020

General University Scholarship, UO, 2020

Strategic Training and Action Research Award, Labor Education & Research Center, UO, 2020

Charles W. Hunt Excellence in Undergraduate Teaching, UO, 2019

Feminist Scholarship, Gender, Work and Organization, Sydney Conference, 2018

Research Award for Data Collection and Presentation, UO, 2017

Joan Acker Graduate Fellowship, UO, 2016-2017

PUBLICATIONS:

Loustaunau, Lola, Lina Stepick, Ellen Scott, Larissa Petrucci, and Miriam Henifin. Forthcoming "No Choice but to Be Essential: Expanding Notions of Precarity During COVID-19." *Sociological Perspectives*

Shtob, Daniel A. and Larissa Petrucci. 2020 "Gendered Care Work and Environmental Injustice: A Feminist Analysis of Educators' Emotional Labor in Disaster Recovery." *Environmental Justice* Ahead of Print https://doi.org/10.1089/env.2020.0038

Petrucci, Larissa. 2020. "Theorizing Postfeminist Communities: How Gender-inclusive Meetups Address Gender Inequity in High-tech." *Gender, Work, and Organization* 27(4): 545-564

Musselman, A. Malori, Andrea P. Herrera, Diego Contreras Medrano, Dan Michael Fielding, Nicole A. Francisco, and Larissa Petrucci. 2020. "Dissonant Discourses in Institutional Communications on Sexual Violence." *Women, Politics and Policy* 41(2): 144-168

Alvarez, Camila, Lola Loustaunau, Larissa Petrucci, and Ellen Scott (equal co-authors). 2019. "Impossible Choices: How Workers Manage Unpredictable Scheduling Practices." *Labor Studies Journal* 45(2) 186 - 213

UNDER REVIEW:

Petrucci, Larissa, Lola Loustaunau, Ellen Scott, and Lina Stepick. "Can We Legislate to Increase Schedule Predictability? Analyzing Oregon's Fair Work Legislation." *Industrial and Labor Relations Review* (2nd Review)

Otis, Eileen and Larissa Petrucci. "Outsourcing Feminism: Hegemonic Femininity and the Symbolic Politics of Supply Chains." *Gender & Society* (1st Review)

ACKNOWLEDGMENTS

This dissertation is the result of much support, love, mentorship, and camaraderie. To my committee, Jill Harrison, Eileen Otis, Ellen Scott, and Anita Chari, thank you for all you have done to encourage and facilitate my growth as a sociologist and scholar. Jill, thank you for the feedback on so many iterations of this project, for the letters of recommendation, for your excitement about my work, and for your guidance over the last seven years. Eileen, thank you for the opportunity to collaborate, for honest and generous discussions, and for making me feel like I have valuable ideas and theoretical contributions. Ellen, thank you for your mentorship, for showing me how to translate my academic interests into policy work, for bringing me on to one of the most amazing and fulfilling projects I got to conduct during graduate school, for financially supporting me, for treating me as a colleague, and for showing me how to make space for ourselves as humans in the research process. I must also thank the Labor Education & Research Center at UO, especially Bob Bussel and Gordon Lafer, for the opportunity to work as a Research Assistant, a job which undoubtedly paved the way for my future work. Lina Stepick, I still can’t believe I got the opportunity to work with and know you. Thank you for helping me develop professionally, for giving me the confidence to see myself as a labor researcher, and for being a constant source of support.

I want to thank my family, Tom, Shelly, Luke, and Janaye, for listening to me talk for the better part of a decade about classes, committees, union actions, the peer-reviewed publication process, conferences, and so many other aspects of my life in grad school. Dad, thank you for encouraging me to finish my PhD, even if it meant more years of hearing “wait, are you still in school?” I’m finally done! Mom, thank you for too many

phone calls to count, for getting dessert delivered to my apartment after a tough day of teaching, for all of the cards, for keeping track of my friends, teachers and supervisors' names, for sending me care packages every time I moved, and for supporting me in everything I've tried to accomplish. And to my more recent family – thank you to Derek, Lizanne, and the boys, for introducing me to Tennent's, for taking a genuine interest in my life, and for welcoming me into your family as one of your own. I love you all!

Thank you to my colleagues and friends. To the Feminist Agenda, for the writing retreats, the vegan brunches, and for showing me the many ways we can love our friends. To Aunt Daddy, thank you for bringing fun into my life, and reminding me there's more to grad school than grad school! To ACL, GTFF, and all of my comrades, for helping me grow more than any formal education ever could, for building collective struggle, and for helping me remember what matters. To Oscar and Savannah, for bringing beauty and joy and love into my life. To Lola. It's difficult to imagine what this dissertation would have looked like without you, and impossible to picture what my life would be like if you weren't in it. Thank you for collaborating with me, for trusting my ideas, for helping me build better ones, for sharing in my happiness, for crying with me, for helping me make decisions, for dinner at your house (special thank you to Juan Sebastián), for reading and talking about my work, for talking about it more, for sharing yours with me, and for letting me be a part of your life. To Cameron. Thank you for making the last few years in Eugene more special than I thought Eugene could be! For proofreading the emails, applications, and manuscripts, for never acting bothered by the many virtual meetings, for witnessing the highs and lows, and for cheering me on. Thank you for making me laugh, for making me bread, and for making me feel loved.

To Grandpa Joe

TABLE OF CONTENTS

Chapter Page

Chapter Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION: MAKING SOFTWARE WORK PRODUCTIVE

> To make knowledge work productive will be the great management task of this century, just as to make manual work productive was the great management task of the last century.
>
> Peter Drucker (1992: 290)

> Capital therefore has an immanent drive, and a constant tendency, towards increasing the productivity of labour, in order to cheapen commodities and, by cheapening commodities, to cheapen the worker.
>
> Karl Marx (1867: 436-437)

In February 2020 Vox published an article that asserted "it's a good time to be a software engineer." This claim stems, in part, from the perception of software work as a "sunrise" occupation: full of opportunities for interesting work and intellectual autonomy in a laid-back and flexible environment (Bergvall-Karenborn and Howcroft 2013). Software engineering and development occupations remain highly sought after, with the labor market still years away from saturation (BLS 2020). In fact, software jobs are expected to grow four times faster than any other occupation in the next decade. Software jobs are also known for their generous pay, with the average software developer making an income of $110,140 per year in 2020 (BLS 2020). The growth and importance of software occupations in the 21st century economy is reflective of economic shifts that began in the 1970s in the United States, namely the transition from an emphasis on manufacturing-based production to knowledge-based production. The rise of a more mobile form of capital, finance capital, and technological innovations that allowed employers to run their companies from a distance, jump started the project of deindustrialization, offshoring, and outsourcing. As part of this process, the United States economy increasingly replaced manufacturing work with service-based industries, information technology, and knowledge production, signaling what scholars began to identify as a "new" economy driven by creative, expert knowledge workers (Bell 1973; Caruso 2016; Florida 2002; Schiller 1973; Toffler 1970).[1]

This "new" economy goes by many names: the information society, the network society, the post-industrial society, the post-Fordist economy, the digital economy, and the knowledge-based economy, to name a few.[2] The workers who fuel this new economy also have a number of aliases: the creative class, the new class, digital workers, and, perhaps most popular, knowledge workers (Bently 1990; Drucker 1969; Florida 2004; Nomikos 1989; Wuthnow and Shrum 1983). At the heart of these monikers, I argue, lies an attempt to make sense of the potentially transformative role of knowledge and

[1] Though knowledge-based production has become increasingly central to the U.S. economy, the manufacturing work of the "old" economy has not disappeared. The claim the contemporary economy is increasingly "weightless," more dependent on intangible services or immaterial commodities than physical ones (Quah 1997), can be understood as a Eurocentric one, ignoring the ways that manual labor has not disappeared, it is increasingly outsourced to the Global South. As Huws (1999: 33) argues, "the decline in manufacturing employment is usually demonstrated within a particular national context, or that of a group of nations (for instance the OECD nations, NAFTA or the EU). This fails to take account of the manufacturing employment which has simply been relocated to another part of the globe." Huws also argues that classification systems of what "counts" as material labor may obscure transformations in material occupations. She writes, "the 'decline' of agricultural employment, which is visible in terms of the numbers of people actually working the land, can only be demonstrated by leaving the mechanisation of farming and the commodification of food production out of the picture. If you were to include, for example, all the people employed in making tractors, fertilisers and pesticides, and all the people engaged in packing and preparing food, and those involved in its distribution to supermarkets as part of the agricultural workforce, the graph would slope less steeply."

[2] Daniel Bell (1973) and Herbert Schiller (1973) spearheaded early discussions of the post-industrial knowledge *society*, exploring the impacts of the growing prominence of industries and consumption based on information and the use of digital technologies. Scholars agree that it can be difficult to empirically identify and measure the characteristics of a knowledge-based economy and to track its growth. Powell and Snellman (2004) use the term knowledge economy to explain a rise in "the production of goods and services based on knowledge-intensive activities that contribute to an accelerated pace of technical and scientific advance" (201).

information technology, and the workers who produce it, both in the structure of a global capitalist political economy and in the organization of firms who must develop managerial strategies to harness the skills of knowledge workers. In other words, though the logic of capital remains the acquisition and expansion of surplus value (profit), scholars have begun to raise questions about how digital commodities, knowledge-based labor, and information technology may transform the relation between workers and employers, posing challenges to managerial control and requiring new strategies to make knowledge workers more productive. These concerns have raised a number of questions: has capital become dependent on the technical labor of knowledge workers like software engineers, shifting the balance of power in favor of labor? What kinds of demands can businesses make over the labor of professional knowledge workers? What organizational forms are needed to support the accumulation of value from the knowledge-based worker? How can capital best extract labor power from knowledge workers, or how can organizations make workers *do* knowledge work? How do workers respond to these managerial control regimes and organizational structures? And, depending on answers to the questions, is it really a good time to be a software engineer?

A Labor Process Theory of Software Work

To answer these questions about the organization and management of knowledge-based work, this dissertation takes software workers as the site of investigation, as they are considered archetypal knowledge workers driving the growth of the new knowledge-based economy (Reich 1991; Marks and Scholarios 2007; Scarbrough 1999).[3] Despite

[3] "Programming" is one of the most commonly used terms to denote all the activities present in the software production process. It is also denoted by other terms like data processing, software engineering (Brooks 1975), information systems work (Orlikowsky 1988) and information-

their prominent role in the contemporary economy, the work that goes into the production of software remains somewhat of a mystery, with limited research taking on a labor process theory framework of software production. Labor process theory is a tradition that was ignited by Marxist political economist Harry Braverman in the 1970s to explore the way that work is organized under capitalism, particularly focusing on the strategies by which managers compel workers to exert the labor power, or to do their work. In this dissertation, I argue that exploring and analyzing software production through a labor process theory framework can help to illuminate how knowledge-based production is performed in the new digital economy, and the strategies by which employers make knowledge-work *more* productive.

Specifically, this research analyzes the impacts of Agile, a lean project management methodology, on the software labor process. Through a labor process theory of Agile, I show that the strategies used to manage the labor of software workers are not all that different from strategies used in automobile manufacturing. I argue that returning our attention to traditional labor process theory and management approaches in industrial manufacturing are necessary to understand the nature of software production today, and knowledge-based work more broadly. Additionally, I draw upon theories of more contemporary managerial strategies and organizational structures like team-based work, post-bureaucratic organizations, and project management to argue that software employers use a hybrid managerial approach to achieve a flexible, predictable, and

technology work (Friedman 1990). In this study, the term "software work" is used to capture all of the activities in the software development life cycle (Ilavarasan 2007).

disciplinary software labor process.[4] I call this hybrid strategy Taylored Flexibility: an attempt to maintain flexibility to respond to the complex and turbulent nature of knowledge-based work alongside strategies to render invisible and immaterial work like software production more calculable and predictable, with the goal of improving productivity. In the sections that follow, I set the stage for a labor process theory of software production by defining knowledge work, explaining the labor process theory tradition and framework, explicating the evolution of managerial control strategies from the 20th to 21st century, exploring what contemporary research suggests control over software work looks like, and describing the major management methodology being used to organize software production today: Agile.

Software Workers as Knowledge Workers

The term "knowledge work" was first coined by Peter Drucker (1969) to describe occupational and organizational transformations that were trending away from manual labor and towards more conceptual and intellectually based labor. Knowledge work remains a highly contested term and concept (Kelloway and Barling 2000), with competing definitions of what knowledge work is and who knowledge workers are (McKercher and Mosco 2008). Knowledge work is most often used to refer to the production of original information or knowledge, the process of adding value to existing knowledge or information, packaging knowledge for employees or consumers, or applying knowledge to a production process (Davenport et al. 1996; Florida 2002). Some scholars prefer an expansive category of knowledge work to include all workers involved

[4] I use the term disciplinary not necessarily in the punitive sense, but to refer to the process by which the worker is habituated to the capitalist mode of production. I follow Braverman's assertion that "the necessity for adjusting the worker to work in its capitalist form...becomes a permanent feature of capitalist society" (1974: 96).

in the knowledge production process such as package delivery drivers or workers who assemble computers (Pellow and Park 2002; Head 2003). Feminist labor scholars point out that insofar as knowledge work relies on human capital, which is gendered and racialized, debates about what "counts" as knowledge work is filtered through assumptions about gender and race (Nishikawa 2011; Walby 2011).

Kelloway and Barling (2000) helpfully synthesize various literatures on what counts as knowledge work and offer three main thematic definitions: knowledge work as a profession (Bently 1990; Choi and Varney 1995; Dove 1998; Janz et al. 1997; Nomikos 1989), knowledge work as an individual characteristic (Ahmad 1981; Brophy 1987; Tampoe 1993), and knowledge work as an individual activity (Conn 1984; Helton 1988; Fox 1990). The authors caution against defining knowledge work through what they see as the fictitious stable category of knowledge worker, suggesting that knowledge work should be understood "not as an occupation, but as a dimension of work" (Kelloway and Barling 2009: 291). In this sense, any worker can engage in knowledge production, and whether or not they do so is primarily a result of their access to training, their motivation for producing knowledge, and opportunities to perform knowledge work. In this study (and others), software workers are considered knowledge workers whose primary dimensions of work *are* completed through cognitive processes. The job of the software worker includes translating problems or goals into working computer programs, which is achieved by drawing upon software programming languages to construct the "series of programme instructions needed to direct the computer to do a particular job" (Ilavarasan 2007: 384). Put simply, software workers are tasked with using their technical and cognitive skills to solve software-based problems, leaving little doubt that they are, regardless of what definition you use, knowledge workers.

While the range of definitions of knowledge work have certainly sparked debate about the nature of knowledge-based production, the question of how to drive knowledge-worker *productivity* has been particularly central in management and organizational behavior literature and is the core question driving this dissertation research. This is because as knowledge production becomes increasingly fundamental to growth of the 21st century economy, it is necessary for firms to determine how to best improve innovation, efficiency, productivity, and adaptability to maintain a competitive edge over other knowledge-intensive firms (Drucker 1979; 1999; Whitely 2006; Atkinson & Court 1998). Productivity of knowledge work is of key concern to knowledge intensive firms, because if "companies can enhance knowledge-worker productivity in this century anywhere near as much as they did with manual labor over the course of the last one (an increase of roughly 50 times) the payoffs will be astronomical" (Davenport, Thomas and Cantrell 2002: 34). Thus, as knowledge work continues to drive capital accumulation, the question of how to manage knowledge workers in the most profitable manner will remain a priority. In this chapter, I outline the managerial strategies and organizational structures that have been particularly popular for making knowledge-work productive, but first, we take a detour into labor process theory in order to establish a framework for analyzing managerial control in the workplace, and the relation between capital and labor.

Conceptualizing Control from a Labor Process Theory Perspective

Determining the best strategies to compel workers to get work done efficiently and productively has been the goal of management, and the purpose of its emergence. In the organizational behavior and management literatures, managerial control may be understood as a "question of how to influence employees' behaviour towards organizational goals" (Lövstål and Jontoft 2017: 42). From a Marxist or labor process theory perspective, the necessity of managerial control over employee behavior is the result of a capitalist labor process, in which workers sell their labor power as a commodity to the capitalist in order to produce commodities for the market. In other words, in order to produce goods, the capitalist hires workers who are willing to sell their labor power, or "the aggregate of those mental and physical capabilities existing in a human being, which he exercises whenever he produces a use value of any description" (Marx 1990: 119) and puts them to work to produce the commodity. Though workers and employers may *appear* to enter the employment arrangement as equals, as the worker is no longer legally forced to work as was the case in a feudal production system, the employment relation between the worker and the capitalist is necessarily coercive, because the primary reason the worker enters into the employment agreement at all is because "social conditions [of capitalism] leave him or her no other way to gain a livelihood" (Braverman 1974: 53).[5]

The worker enters into the employment relation in a position of subjugation to capital, selling their labor power as a commodity to the employer in order to obtain a wage such that they can reproduce their own life. However, when the worker sells their labor power, they are selling a "peculiar" commodity, a commodity that is embodied in a human with free will that has "infinite adaptability...so that its surplus product may be

[5] Put differently, P.K. Edwards (1990: 128) states that "in capitalism, workers lack the means of production and have to sell their labour power in order to live. This is the mode of exploitation."

continuously enlarged" (Braverman, 1974, p. 38).[6] In other words, while the machine has limits on how much profit it can help to produce for the capitalist (at least until a newer machine is built), the profit that the capitalist can produce from the worker is the result of how much work the worker is willing to exert themself. In this sense, while the employer is able to purchase the worker's labor as a commodity, they are not purchasing a known quantity of labor, but the *capacity* for workers to labor:

> What he [the capitalist] buys is infinite in potential, but in its realization it is limited by the subjective state of the workers, by their previous history, by the general social conditions under which they work as well as the particular conditions of the enterprise, and by the technical setting of their labor. The work actually performed will be affected by these and many other factors, including the organization of the process and the forms of supervision over it…(Braverman 1974: 57).

To explain further, when the employer purchases a person's capacity to labor, or their labor power, they are purchasing "on the one hand a potentially malleable commodity" which functions as the essential resource for expanding profits, but on the other hand, "a commodity ultimately controlled by an independent and event hostile will" (i.e., the worker) (Friedman 1977: 78). Thus, in order to realize and expand profits, the employer must engage in a process in which they can ensure the worker produces commodities in the most efficient and productive way. The process through which workers exercise their labor power, and by which managers organize production in order to create profit, is the capitalist labor process. As Barrett (2001:19) explains,

[6] The enlargement of surplus can be achieved primarily by two means: an increase in the production of absolute surplus value and/or an increase in the production of relative surplus value. The former is achieved through "the prolongation of the working day beyond the point at which the labourer would have produced just an equivalent for the value of his labour-power, and the appropriation of that surplus labour by capital," while the latter refers to "the surplus-value arising from the curtailment of the necessary labour-time." (Marx 1990: 221).

> the labour process under capitalism has specific characteristics, the most significant being the transformation of labour power into actual productive labour. In the employment relationship, control, therefore, emerges as a means to transform the capacity of work into profitable production.

Control over the labor process becomes a necessary priority for management so that they can ensure that workers' capacity to work will be realized in the most profitable manner.

The theory that described this capitalist logic necessitating managerial control over the labor process, as well as analysis of how control can be achieved by the employer, was initiated by Harry Braverman's (1974) seminal text, *Labor and Monopoly Capital: The Degradation of Work in the Twentieth Century*. Braverman argued that as early industrial capitalism advanced to monopoly capitalism, employers could increasingly wield power over workers as they gained control over the labor process. Braverman provided an analysis of the dominant managerial control methodology used in the early twentieth century, Frederick Winslow Taylor's scientific management, which was most popularly adopted by Henry Ford. As part of the scientific management approach, which I describe in more detail in Chapter II, workers' skills and their ability to have control over the full production of their crafts was replaced by the division of labor, or the breaking down of each step of the labor process and assigning a worker to only one of those steps. With the rise of factory work, those who were formerly expert craftspeople now sought jobs in the industrial factory, relegated to adjusting widgets on an assembly line, a process that Braverman called the degradation of work and the deskilling of the worker. Braverman argued that the capitalist mode of production, grounded in the logic of capital accumulation (profit expansion), "systematically destroys all-around skills where they exist, and brings into being skills and occupations that correspond to its needs" (Braverman 1974: 82). The division of labor and the deskilling of workers developed alongside the physical and ideological separation of production, or a distinction between the workers who physically assemble goods and management, those who were seen as intellectually fit and authorized to organize the production process. Importantly, the separation of conception from execution was a gendered, racialized, and classed process, with particularly immigrant women of color working the most routinized and lowest paid factory jobs (Barrett 1999). Gender and racial segregation are also present in high-tech work, being one of the most male-dominated occupations (BLS 2020). As I discuss in more detail in Chapter IV, the masculine organization of tech work has implications for gender and racial equity within teams, but also operates as a strong motivator for resistance and action in the workplace.

Ultimately, Braverman's seminal text "stimulated numerous empirical and historical studies, the majority of which have addressed one or other of its two central themes: deskilling and strategies of management control" (Knights and Wilmott 1990). As part of the labor process theory tradition, scholars debate methods of management control, challenging Braverman's claim that the capitalist labor process *necessarily* leads to the deskilling of labor through the control strategy of scientific management to instead show other possibilities (Wood 1982). Edwards (1979), for example, outlined three different types of control: simple, technical, and bureaucratic. He showed that managerial strategies for control may vary depending on the type and complexity of production that workers are involved in (e.g., assembly line work versus office work). Similarly, Friedman (1997) offered an alternative model of managerial control strategies to Braverman's narrow focus on direct Taylorist forms of control, through a conceptualization of a spectrum of strategies that management can choose from, from what he called direct control to responsible autonomy. Where direct control is achieved through close supervision and tight mechanisms of control, much like the scientific management method under Taylorism, responsible autonomy refers to management's ability to "harness the adaptability of labour power by giving workers leeway and encouraging them to adapt to changing situations in a manner beneficial to the firm" (Friedman 1977: 78). Ultimately, this dissertation primarily engages with and applies Friedman's framework to analyze the ways that direct control and responsible autonomy are elicited to improve productivity over the software labor process.

The Struggle for Control

Though control over the labor process is a managerial imperative in the sense that employers under capitalism must develop strategies to enlarge surplus through the disciplining of labor in order for their firms to survive, managers do not necessarily make decisions from a position of rationality.[7] The expectation that managers develop a single and rational strategy for securing profit, Storey (1985: 194) claims, has been a weakness of labor process research, which he argues has tended to "adopt a theoretical stance which is essentially structurally based and determinist." Labour process theories are subject to determinism when they assume that the capitalist drive for profit necessarily engenders coherent systems of control. Instead of assuming that managers take a rational,

[7] P.K. Edwards (1990: 128) offers a helpful explanation of the employer imperative to enlarge surplus or profit: "capitalism is a mode of production in which the purpose is not just exchange as opposed to use, nor even exchange-value as opposed to mere exchange as in barter, nor yet maximum exchange value as against some exchange value, but production for capital accumulation. 'Capital' is not the sum of individual capitalists but is the social force stemming from the mode of production: there is a constant pressure to produce, and this requires investment in productive capital, the production of exchange value, and the reinvestment of the surplus as accumulation advances. Surplus labour (that over and above the labour necessary to reproduce labour power) in capitalism appears as profit."

calculated, and fully coherent approach to control, Hyman (1987: 30) argues that "the key to any credible treatment of strategy within a Marxist analysis is surely an emphasis on contradiction" (Hyman 1987: 30). In other words, in order to understand the particular strategies that managers employ to gain control over the labor process, we must recognize a number of contradictions within the capitalist production system that managers may try to overcome. The contradiction within capitalism should not be viewed as "a conflict between two opposing statements," but "a tension inherent in a mode of production" (Edwards 1990: 135). For example, though capitalism is

> characterised by the drive to accumulate: it is a dynamic system and needs to continue to expand...competition between firms tends to drive down the rate of profit, and this is in contradiction to the need for growing profits arising from the drive to accumulate.

In the context of knowledge work, management aims to overcome the tension between "*knowing* [sic] as an active, lived experience" and "*knowledge* [sic] as a commodity within firms and markets." As a result, "Management becomes the focal agent in attempting to integrate two divergent sets of social practices; on the one hand, the human actions involved in producing and applying knowledge and on the other exploitation of such actions for economic ends" (Scarborough 1999: 6). In this sense, contradiction and antagonism is a fundamental aspect of the capitalist labor process, or the means of accumulating profit.

As labor process theorists have pointed out and criticized Braverman for ignoring, another key tension in the capitalist labor process is the labor-capital antagonism, or worker and employer (Clawson 1980; Edwards 1979; Stark 1980). The relationship between capital and labor is necessarily a relation of conflict, not in an interpersonal sense, but in a structural one. As Hyman (1987: 35) argues, "the capitalist labour process is at one and the same time a cooperative and conflictual activity." Burawoy (1979:30) most popularly cited for analyzing the role of conflict in the labor process in his ethnography of workers in a Chicago factor, arguing that the labor process "must be understood in terms of the specific combinations of force and consent that elicit cooperation in the pursuit of profit" (Burawoy 1979:30). Burawoy showed that while conflict may come in the form of resistance from workers to the capitalist production process, conflict may also be established horizontally among workers. Moreover, the labor process includes eliciting consent from workers, which Burawoy argues is established, in part, by workers developing strategies to minimize their exploitation without calling the capitalist labor process as a whole into question.

Still, the necessity of the capitalist to control the worker in order to accumulate surplus, and the desire of humans to exert autonomy and discretion, creates a structured antagonism in the relation between capital and labor (Edwards 1986). This structured antagonism increases the likelihood of resistance from the worker, as Storey (1985: 196) explains:

> A key structural element of management is control. But because perceived interests are thereby potentially threatened, workers do in varying degrees resist this control both individually and collectively, passively and actively. This dynamic of contestation constitutes the basis for a dialectical interplay between control and resistance. The means of control which are in consequence actually emergent have to be understood, therefore, as products of this [conflictual] process and not either as Management Science idealisation nor as abstractions of the global functions of capital.

In other words, in order to engage in labor process theory to identify evolutions in control strategies, scholars must acknowledge how worker resistance shapes work arrangements, and vice versa.

Importantly, because managerial strategies are not necessarily the result of coherent and rational choices by employers, but responses to a number of internal and external contradictions and tensions, managerial control is not usually exerted in a singular manner (Callaghan and Thompson, 2001). As Sturdy et al. (2010: 114) suggest, “the early work of outlining the historical evolution of distinct forms of managerial control has been moderated to show the hybrid nature of control in practice.” Rather than impose a singular or rational approach to labor control, management is “an active process,” in which there is never “one best way” to manage (Friedman 1977). In other words, managers always choose, at times in contradictory ways, among an array of potential control strategies. In this dissertation, I adopt a framework of control strategies as blended and hybridized in order to analyze the software labor process. In the sections that follow, I discuss more specific control mechanisms that have been applied to different kinds of workers, leading scholars to suggest that control over knowledge-based work like software requires forms of control distinct from the traditional scientific management methodologies implemented in the early industrial stages of capitalism.

New Control for the New Economy

Now that we have a framework for understanding control over the labor process, we turn to the problem of control over knowledge-based work in the new economy. The “new” knowledge-based economy is often conceptualised in dichotomous relation to the Taylorist or Fordist production processes described by Braverman, and is said to require new forms of management strategies to secure profit.[8] The labor process under Taylorism

[8] While scholars generally agree that, particularly beginning in the post-World War II era, production processes have increasingly incorporated new technologies as well as information and knowledge, critics of the knowledge economy thesis argue that, first, the assumption that Fordism dominated production is an overestimation and, second, the fact that monopoly capital and

and Fordism is well documented as resulting in workers' experiences of boredom, alienation, low-self-esteem, and dissent (Smith 1997), experiences of work that are not well suited to encouraging knowledge-workers to be innovate, creative, or apply their expert skills. Instead, new organizational structures emerge in order to respond to the "human" factor of labor as well as harness the power of new technologies to control work. In this section I will describe the problem of control that knowledge-workers appear to post to management and explore what are regarded as some of the new strategies to control knowledge workers.

The Problem of Control over Knowledge-Based Work

Control over knowledge work is presumed to challenge traditional managerial control strategies and require new forms of control as a result of its discontinuous, unpredictable, and rapidly evolving nature (Alvesson 1993; Atkinson & Court 1998; Castells, 1996; Drucker 1979; Drucker 1999; Florida 2002; Gephart 2002; Locke et al. 1995; Powell and Snellman 2004; Prusak 1997; Lloyd and Sveiby 1989). The old Taylorist modes of control are seen as incongruent with new knowledge-based work, as top-down, rigid, and inflexible organizational structures limit knowledge-based firms'

knowledge-based commodities may present new challenges for accumulating surplus value does not necessarily mean that management strategies in the "old" economy are "replaced by a clear alternative" (Thompson 2003: 362). Thompson argues that management-worker relations in the 'new' economy are dependent on a bargain where workers are given more responsibility in exchange for autonomy and opportunities to develop human capital skills through "training, enhanced career structures, job stability and performance and skill-based reward measures" (2003: 363). Yet, as Thompson argues, employers are not able to hold up their end of the bargain because while they can see the strategic value of 'flattening' hierarchies and giving workers more responsibility, accumulation can be most securely extracted through disciplinary mechanisms.

ability to be flexible and innovative, and to harness a knowledge-based, rather than a manually based, dimension of work.[9] As Kelloway and Barling (2000: 293) write:

> It is relatively easy to coerce and control physical labor that by definition is observable and measurable. Indeed, by applying the appropriate levels of job design and control, an employing organization can fairly easily ensure that employees are operating at "peak efficiency." In contrast, knowledge work is fundamentally unobservable - one observes the outcomes not the process of knowledge work. As a result, the organization cannot impose external controls. Rather, the organization must focus on creating conditions for the enhanced performance of knowledge work.

Because knowledge work is not physical or not immediately visible to the employer, in the sense that the employer is not able to see the cognitive processes of knowledge production in the same ways that they can observe workers fitting metal to a car, new strategies are needed to make sure that the firm is getting the most out of their knowledge workers.

Antikainen & Lönnqvist (2006: 2) similarly argue that "since knowledge work differs so drastically from manual work, the attempt to apply measurement and management methods used in traditional industries will not lead to an improvement in productivity of knowledge workers." This claim has been made of software workers specifically, with Andrews et. al (2005: 71) writing, "the method by which surplus value is extracted and accumulated holds little resemblance to nineteenth or twentieth century manufacturing. This is probably due to the unique nature of the commodity, software, which is the result of knowledge work." Thus, scholars have established what appear to be strong distinctions between manufacturing and knowledge-based work. Firms that are increasingly dependent upon knowledge workers to create innovative and value-added

[9] Fordism, attributed to Henry Ford, is considered one of the dominant organizational structures in manufacturing-based work, associated with the assembly line, interchangeable workers, and a steep top-down hierarchy between executives, managers, shop floor managers, and workers.

products seem to require new organizational structures and managerial strategies to compel workers to transfer their knowledge to the company in order to expand profits, abandoning the old methods of control found in Ford's factory. Moreover, knowledge workers appear to "represent a new breed of worker with different needs, values and motivators from traditional workers," (Kelloway and Barling 2000: 288) such that managerial strategies of top-down or direct control may just lead knowledge workers to seek employment elsewhere. As part of these concerns, organizations have adopted alternative structures, emphasizing a less rigid and hierarchical organization referred to as the post-bureaucratic organization.

The Post-Bureaucratic Turn

As early as the 1950s, scholars challenged if tight bureaucratic organizational structures were most suitable for organizing work (Blau 1995; Merton 1949), though the "new management revolution" calling for post-bureaucracy became most popular in the 1980s and 1990s alongside the production of new technologies (Hecksher 1994; Johnson et al. 2009; Bolin and Harenstam, 2008). Post-bureaucratic organizations aim to "flatten" rigid hierarchies to mitigate differences in authority between managers and workers by grouping organizational members into teams, and offering workers creative control, autonomy, training, professionalization opportunities, personal responsibility, and stock owning options (Castells 1996; Maravelias 2003; Tschang 2007; Raelin 2011). Post-bureaucratic firms are associated with High-Commitment Management (HCM) approaches, in which workers are "highly involved in such matters as making day-to-day decisions, scheduling their work, solving their own problems, and supervising their own activity" (Barker 1993: xi; Walton 1987). Workers displaying high commitment to their organization benefit firms as they are observed to work more efficiently and productively (Bashaw & Grant 1994) and even come to work more regularly (Hackett, Bycio, & Hausdorf 1994). Firms who benefit most from post-bureaucracy are those where scope or product flexibility are required or desired (Youndt et al. 1996) and have been associated with high-technology and knowledge-intensive firms (Alvesson and Deetz 2000).

A critical perspective on the post-bureaucratic turn argues that new managerial practices emphasizing autonomy *replace* rather than *displace* control in the workplace (Briand and Bellmare 2006; Ezzamel and Willmott 1998; Barker 1993), operating as forms of “soft despotism” “normative” or “ideological” control (Clegg and Courpassen 2004; Barley and Kunda 1992; Kunda 1992; Willmott 1993), or what Freidman (1977) called responsible autonomy. Normative controls in post-bureaucratic firms arise from the development of a strong corporate culture, which operates to “exploit workers not through traditional methods of direct coercion, but via the indoctrination of shared corporate beliefs, norms and values” (Sturdy, Fleming and Delbridge 2010). For example, Baldry, et al. (2005: 168) show that rather than internalize the goals of the firm, software workers internalize their roles as experts and entrepreneurs, engendering “commitment to the job and to the professional identity to which the job bestows.” Especially for young tech workers, Baldry et al. (2005) find that responsible autonomy increases personal expectations to produce a quality product in a timely fashion. Workers who do not submit to non-standard and long working hours are “interpreted as individuals not being committed to work and not being serious about a career” (Rasmussen and Johansen 2005: 105). Thus, professional identity operates as a mode of control over the speed and quality of software production.

Teams, too, operate as a form of control in post-bureaucratic firms, allowing management to establish accountability mechanisms that reconfigure the locus of control from supervisors to team members, obscuring hierarchical power relations in the workplace without fully replacing them. Team-based control mechanisms include the development of team commitment (Kunda 1992), peer surveillance or peer scrutiny where team members monitor one another's work (Barker and Tompkins 1993; Ellway 2013; Sewell and Wilkinson 1992), and concertive control, or establishing norms and disciplinary practices for breaking them (Barker 1993). Normative control also helps firms to elicit the kinds of emotional labor necessary for a particular labor process by encouraging workers to have fun, or "just be themselves" at work in order to take on the emotional disposition necessary to provide certain services to customers (Fleming and Sturdy 2009; Kinnie et al. 2000). Indeed, the project team literature broadly and software development team literature specifically emphasizes the impacts of interpersonal and collective empathy on improving productivity over software development work, leading to better project performance (especially in terms of team learning, time-to-market, and cost) (Rues and Liu 2004; Akgu¨n et al., 2015).

While "soft" or normative controls like responsible autonomy and personal and team commitment have been successfully adopted by knowledge-intensive workplaces, it is important to note that control strategies are typically adopted in constellations or are "blended" (Sturdy, Fleming, and Delbridge 2010: 114; Callaghan and Thompson 2001). As previously discussed, employers often pursue a "contradictory coexistence of enabling and coercive aspects" of management control (Alder 2005: 421), applying HCM schemes (a form of responsible autonomy) alongside more traditional and tight mechanisms of control called direct control. Thus, normative control should be conceptualized as not only eliciting the kinds of behaviors and dispositions necessary for workers to be more productive, but also a form of control that distracts workers from “the dysfunctions of *existing* technical, bureaucratic and conventional cultural controls” (Fleming and Sturdy 2010:177). In other words, normative forms of control like responsible autonomy have the dual function of instilling a sense of commitment in workers to improve productivity, while also obscuring direct forms of control that they might otherwise challenge. In the next section, I examine the roles of responsible autonomy and direct control over the software labor process in particular and discuss how scholars have theorized modes of control over software work.

Making Sense of the Software Labor Process

In 1998, Beirne et al. stated that “the nature of the work performed in producing and operating software should be a matter of greater curiosity to labour process analysis” (142). In the two decades that have passed since this statement, relatively little has been published on the *labor process* of software development, and the software production process itself remains somewhat mystified (Barrett 2005). Perhaps this is because as archetypal knowledge workers, the actual work that software developers perform may seem opaque and enigmatic, leaving little material trace in an otherwise digital world. The research that has been conducted on the software labor process is typically divided into three categorized: those who view software work as subjected to Taylorist task fragmentation, those who argue that knowledge-based work in general requires new forms of control, as has been discussed in this chapter, and scholars who have provided a more “mixed,” analysis, showing how direct control and responsible autonomy are used simultaneously to create a control regime (Ilavarasan 2007).

For some, the evolution of software development appears to support Braverman’s deskilling thesis, with scholars arguing that the introduction of coding language and structured programming simplified formerly complex skills needed to conduct software development work (Kraft 1977; 1979; Greenbaum 1976). Kraft argued that the creation and widespread use of coding languages by software developers made the need for advanced mathematical, and even computer knowledge obsolete, acting as a form of deskilling. Kraft also argued that structured programming became “the centerpiece of management efforts to de-skill programmers” (Kraft 1977: 99). He explains:

> Structured programming makes it possible to organize programming along the lines of industrial rather than craft production. In this case, what is standardized is not a material product like an automobile or a package of breakfast cereal or a bank statement. What is standardized is a mode of thought, a logic, a pattern of decision-making.

Kraft argued that the standardization of programming logic facilitates the division of labor, or task fragmentation, as some high-level workers are assigned more interesting work, while other, deskilled workers take on less-complex portions of software development work. In other words, structured programming provided managers the ability to “fragment the larger system into discrete components” such that “it was not necessary for individuals/teams to know how their component fitted into the system” (Barrett 2001: 25).

The view of software work as task fragmented and deskilled has been met with criticism, with some scholars arguing that Kraft and Greenbaum have misunderstood the software production process (Orlikowsky 1988; Glass et al. 1992). Contemporary labor process theory analyzing software work tends to take a less deterministic approach, showing how hybrid forms of control like project management have been particularly effective for organizing software work, especially as "software projects typically involve a combination of complex problem solving with the need for a certain amount of innovation, yet much of the software development is also routine and calls for dependable, predictable tactical performance (Constantine 1993: 41)." Scholars have noted the role of responsible autonomy or normative controls in the software context but have identified that autonomy may operate as more of an illusion than a liberatory aspect of work, maintaining a critical view of "flat" or post-bureaucratic structures used to organize software work (Hodgson and Briand 2013; Rasmussen and Johansen 2005.) Damarin (2006; 2013) reveals the tensions between and direct control over web development work, arguing that deadlines set by clients and bosses operate as a form of direct control for web developers, and that while these workers do enjoy somewhat high levels of autonomy over their concrete labor practices, they are only "free to do as they please...if they please clients (and collaborators and suppliers)." Barrett (2011; 2004; 2005), too, found that though software workers did experience some time and technical autonomy in their work, they were still subjected to the normative control of HCM, as well as direct control in the form of project deadlines, frequent goal settings and meetings, and product specification. These studies confirm Briand and Hodgson's (2015: 301) argument that "control must be analyzed as a dynamic matter" in which control and autonomy are two sides of the same coin, as long as autonomy represents a method by which firms are able to align the disposition of workers with the goals of the firm, namely, improving productivity. This dissertation extends the hybrid approach to labor process theory analysis of software work by drawing upon managerial control strategies used in manufacturing-based work and showing how they can be applied to software work.

Project Management as a Hybrid Control Regime

An increasingly pervasive post-bureaucratic tool for coordinating, organizing, and intensifying complex work that relies on both direct and normative controls is project management, with more and more organizations identifying as project-based (Hamilton 2001; Hobday 1998; Pakendorf 1995; Morris and Pinto 2004; Whitley 2006). Part of the post-bureaucratic movement, project-based work and project based-organizations appear to be especially suited to manage complex, turbulent, creative, knowledge-intensive, and high-technology driven production processes (Ekstedt et al. 1999; Hobday 2000). Projects can be defined as a way to organize work, which "involves teams, are temporary in nature, focus on a specific task and are scheduled to be completed within some defined time, cost and performance standards" (Thomas 2006: 91-92). Project management functions as an attempt at organization rationalization or improving the efficiency of the production process (Hodgson 2002) through the "breakdown of an overall activity into an articulated life-cycle [the length or duration of the project], and the detailed specification of tasks and outcomes for each stage and sub-stage of the life-cycle" (Gleadle et al. 2012: 166).

Firms use projects to clearly define the time and costs necessary to produce products for clients (Samset and Volden 2016: 298) in order to minimize waste associated with uncertainty and rigidity. As Hodgson (2004) notes,

> Project management has been put forward by many as a "tried-and-tested" package of techniques able to cope with discontinuous work, expert labour

> and continuous and unpredictable change while delivering the levels of reliability and control of the traditional bureaucracy.

The field of critical project studies illuminates the ways that project management operates as a hybrid control system, containing attributes of "both top-down bureaucratic control...and also autonomy" (Gleade et al. 2012: 166), to achieve the "close control and monitoring of labour resources, costs and time constraints" (Metacalfe 1997: 305). Thus, project management has dual benefits for firms trying to manage knowledge work, in that "it offers techniques to render a discontinuous, relatively unpredictable activity more controllable and at the same time to discipline employees working in such environments by rendering them more accountable" (Gleadle et al. 2012: 166). In other words, project management relies on normative HCM control strategies alongside tighter, more traditional control mechanisms in order to rationalize work, making it more efficient, productive, and flexible.

In the context of project work, flexibility operates as a form of control in that it often requires employees to improvise in order to meet project demands (Lindahl 2006), which can lead to expectations that workers attend to project requirements outside of standard business hours or overwork (Perlow 1997; Upadhya 2009). Commitment is further supported by the notion that projects will only last for a short period, despite workers' need to engage in project after project (Lindgren and Packendorff 2006). Unsurprisingly, workers' continuous engagement in projects, and at times, multiple projects at once, can increase psychological stress, lower productivity, and even leave workers with a sense of hopelessness and worthlessness if they fail to meet project demands (Zika-Viktorsson et al. 2006; Cicmil et al. 2016).

While projects offer firms a methodology for organizing complex work, research shows there is still quite a disparity with project management in theory and the effective implementation of project management in the workplace (Williams 1995; Belassi and Tukel 1996; Morris et al. 2004). The managerial response to this problem has been to emphasize making improvements to project management methodologies (Hodgson and Cicmil 2006), an approach that I argue has helped to construct the emergence of Agile project management. In other words, I suggest that Agile is best understood as part of the rationalization process *within* the project management movement, the result of an effort at continuous improvement over project-based work.

Agile Project Management: A Better Methodology for Control?

Agile is an iterative, team-based project management methodology that was developed as a response to Waterfall project management, which generally dominated software production processes until the 1990s. Waterfall is a hierarchical, linear, sequential and rigid production process, with each step of production depending on the success of the former step. Because of Waterfall's inflexible nature, it has been associated with the rigidity responsible for weakening the competitive position of knowledge-intensive firms, resulting in high costs and waste, and viewed by managers and workers alike "as cumbersome, ineffective, and extremely bureaucratic" (Cobbs 2011: 5). So, following the broader trend of replacing bureaucratic organizational forms with post-bureaucratic ones, "adaptive" software production became popular in the 1990s culminating in the 2000 Agile Manifesto, a guide to what appeared to be a more flexible yet still controllable project management methodology. A more detailed discussion of Waterfall and the transition to Agile is developed in Chapter II.

In this dissertation I follow Barrett's (2001: 31) reminder that "many aspects of the new can be found in the old if you only look in the right place," to show that while Agile is a relatively new methodology that relies on the normative controls of post-bureaucratic HCM regimes, Agile also applies processes that emerged in automobile manufacturing in the 20th century. As Morien (2005: 1) explains, Agile can be understood as "an approach to managing software development projects that can be seen to have roots in lean manufacturing thinking and practice, derived in many respects from what is known as The Toyota manufacturing process." The role of lean production in the project-management literature broadly and the literature on software production in particular remains underexplored and undertheorized. Thus, while we may know that project-based work relies on responsible autonomy *and* direct control strategies, there is little research exploring how direct control unfolds, transforms the production processes, and even impacts the strategy of responsible autonomy, as I explicate in Chapters II and III. Illuminating Agile's lean roots complicates common assumptions that knowledge-intensive work like software requires *new* methods to improve productivity and reveals how evolutions in project management are being pursued by looking to the past with an eye on manufacturing processes to create a regime of Taylored Flexibility, as I develop in the remaining chapters of this dissertation.

Methodology

In order to develop an analysis of the software labor process, I conducted qualitative research with workers and managers who worked at firms that use Agile. Research for this dissertation was conducted in two phases: 1) an investigation of the impacts of Agile on the software labor process and 2) an investigation of workers' collective organizing attempts, through a pilot study of the Tech Alliance.[10] In both phases, this dissertation research included a combination of in-depth semi-structured interviews (n=45) and content analysis of online materials.

Recruitment

In order to analyze the impacts of Agile on the software labor process, this research took Agile project management, rather than a singular worksite, as the target of investigation. Recruitment of workers and managers employed at firms or on teams that use the Agile methodology consisted of cold-messaged participants on the website Meetup.com. Meetup.com hosts meetup groups, or interest groups, that folks typically join in order to attend in-person meetings. While Meetup.com does facilitate networking, it's primary purpose is not necessarily a social networking site, but a "platform for finding and building local communities. People use Meetup to meet new people, learn new things, find support, get out of their comfort zones, and pursue their passions, together."[11] There are many software meetup groups on Meetup.com, as well as specific groups for Agile project management or Agile software development. I narrowed the geographic location of potential participants to San Francisco, CA, Portland, OR, and Seattle, WA capturing three major tech cities in the United States West Coast. I cold-messaged 93 people on meetup and 22 responded. I snowball sampled from those participants to recruit another 11.

I also recruited participants (n=12) from the Tech Alliance, who worked at firms who use Agile as well, meeting the criteria for both the first and second waves of recruitment. In the spring of 2019, I attended a Tech Alliance meeting in Portland, OR.

[10] Tech Alliance is a pseudonym.

[11] https://www.meetup.com/about/

This was considered one of the first official meetings of the Portland chapter, and lead organizers invited three Tech Alliance Seattle members to join and educate members on how the organization typically runs. Seattle members noted that each Tech Alliance chapter is unique, and that while chapters are typically structured around shared principles--democracy, horizontal and grassroots organizing, free membership, etc.--each chapter is free to develop their own internal structure. After establishing some trust with members at the Portland chapter I was invited to participate in the Tech Alliance Slack channel, an online communication platform that most chapters participate in. In February 2020, I attended the Seattle chapter general meeting, taking in-depth field notes during and after the event. I had intended on participating in Seattle chapter meetings each month throughout 2020, but as the COVID-19 pandemic grew more serious in March, Seattle cancelled all of their in-person events and opted not to hold online events. However, by being a member of the Slack channel, I was able to cold message participants to recruit them for my research. Ultimately, I interviewed two members of the Portland chapter, four members of the Seattle chapter, two members of the Minneapolis, MN chapter, two members from the New York, NY chapter, two members of the San Francisco, CA chapter for a total of 12 interviews. I also attended a virtual meetup in San Diego, CA and one in Boston, MA.

Sample Demographics

This research includes 45 interviews with workers (n=22), leads, junior managers, or team coaches (n=11), managers with hiring and firing authority (n=9), and professional Agile Consultants (n=3), conducted over an 18-month period. All interviewees identify as working in occupations that were "involved in the software development lifecycle." Participants' occupations were wide-ranging, as there are not necessarily systematic titles given to tech workers, an issue that will be explored in more detail in Chapter IV. Some examples of titles include the following: Quality Assurance Scientist, Data Architect, Senior Project Manager, Associate Developer, Software Development Engineer, Web Developer, Full Stack Developer, Principal Software Engineer, Agile Coach, Scrum Master, Senior Software Engineer, Technical Product Manager, etc. Participants' ages ranged from 23 years old to 60 years old with an average of 34 and a mode of 34. Participants had a range of salaries, from $54,000 to $185,000, with an average of $122,800. The majority of participants were full-time employees (n=39) and 6 were independent contractors.

In cold-messaging, I made efforts at a purposive sample in order to oversample for women and Black, Indigenous, and people of color. In 2020 in the United States, women made up 25% of Computer and Mathematical occupations, whites 65%, Black/African American 9%, Asian 23%, and Latinx workers 8% (BLS 2020). To purposely sample, I messaged people that appeared to represent a variety of racial/ethnic backgrounds and genders. In this study 21 participants identify as male, 4 as non-binary, 18 as female, and 2 chose not to disclose their gender. 26 participants identify as white, 12 as Asian, 1 as African American, 1 as half Indonesian and half white, 1 as half Hispanic and half Asian, 1 as Latina, 1 as Latino, and 2 chose not to disclose their race/ethnicity.

Interview Design

Interviews with participants were semi-structured and open-ended, lasting between 45 minutes and 2 hours, though most typically lasted an hour and a half.

Interview took place in person (18), over-the-phone (20), and over video (7). For face-to-face interviews, I conducted interviews outside of the workplace, most commonly at a coffee shop or a park, so that participants had the opportunity to talk more openly and honestly, which they may avoid if they fear being overheard by a supervisor or coworker (McDowell 1998; Quinney, Dwyer and Chapman 2016). I conducted all phone and video interviews in my home and recorded them using a phone recording app or an online voice recorder. To be sure, phone interviews have been criticized for presenting barriers to developing rapport, limiting the possibility of identifying visual cues, and for generally impeding the ability to gather rich data (Novick 2008; Holt 2010). However, qualitative researchers have begun to document the benefits of telephone interviews, which are many: they provide greater scheduling flexibility, they are associated with lower costs, they overcome geographical boundaries, and they improve interview safety and security (Cashia and Millward 2011; Carr and Worth 2001). Moreover, phone interviews may be beneficial as participants may feel a greater sense of anonymity which encourages more frank responses, allow researchers the opportunity to take notes throughout the interview without feeling self-conscious, and may minimize distraction as interviewees can choose the location and time of the interview (Lechuga 2012; Stephens 2007). Interestingly, Irving et al. (2013) even found that interviewees participating in phone interviews were more likely to ask for clarification on questions and confirm that their answers were adequate. In my experience, phone interviews did not yield noticeably different results from in-person interviews. I did not find it difficult to establish rapport over the phone, and, as a young woman, ultimately felt more secure conducting interviews with strangers who were men over the phone.

Rather than take on a rigid role as a researcher, one who just wants to extract data from the participants, I aimed to build “conversational partnerships” (Rubin and Rubin 1995). I did this by being “warm and responsive,” and sharing some details of my own life with respondents. I also emotionally engaged with respondents, sharing feelings of frustration and empathy especially as they described painful experiences, and, in one case, crying along with a participant. Participants were asked questions about their job duties, relationships with their teammates, relationship with their managers, processes by which they were assigned particular projects and project tasks, and any issues they had at work that they wanted to or had even attempted to change. Participants were also asked questions about the ways their teams use Agile, including questions about how daily meetings are structured, the layout of their office spaces, their knowledge of Agile principles and methodologies, etc. As I began to be more comfortable with Agile jargon, I was able to ask more and more specific questions of participants. Still, I often asked participants to unpack and explain any jargon they were using in order to ensure I was able to compare experiences accurately. As interviews progressed, I also requested feedback from respondents on the best way to ask particular questions such that they would be more intuitive or make more sense. For example, I might ask the participant “did I ask that question in a way that makes sense?” adjusting any confusing questions as the data collection process progressed. Interviews with participants in the Tech Alliance included questions about Agile as well as questions about Tech Alliance. Participants were asked some of the following questions: How long have you been involved in Tech Alliance? What is your experience with labor organizing or labor organizations? What are the goals of this Tech Alliance chapter? How does the organization meet those goals?

How would you describe the structure of the organization? What kinds of people come to Tech Alliance events? All interviews were recorded and fully transcribed by me.

Interview Analysis

Qualitative analysis tends to be driven by an inductive approach, in which the categories and themes of analysis emerge from the data, rather than using interviews to test hypotheses that come from theory or literature (Strauss and Corbin 1998). However, as many of us who have conducted qualitative research know, the researcher can never enter the field as a fully objective scientist. We enter the field with questions, ideas, interests, and life experiences, which shape the ways we design, conduct, and ultimately, analyze qualitative data. Esterberg (2002) explains this well:

> In social research, humans are the researchers as well as the objects of study, which means that pure objectivity is impossible. We have a vested interest in what we study…challengers to traditional ways of doing social science argue that all knowledge is created within human interaction. Who we are shapes the kinds of theories we create and the kinds of explanations we offer.

Ultimately, I approached the field as a feminist critical labor scholar. The theory of Taylored Flexibility that I develop in this dissertation research is the result of empirical qualitative data collection or the stories, experiences, and insights shared with me by those who actually *do* the work of software production, alongside a theoretical framework that has been outlined in this chapter, namely, a labor process theory approach. As a feminist critical labor scholar, I analyzed qualitative data with a theoretical framework that highlights power differentials, structural inequality, and structured antagonisms that are inherent in the capitalist production process. However, as a labor sociologist and most certainly not a software developer (I have no experience with computer coding besides a 12-week undergraduate course in web design), participants were experts on their work arrangements. Thus, I engaged in a process of iteratively developing my theoretical analysis as I gleaned new information from participants (Srivastava and Hopwood 2009). In fact, it was my second interviewee who explained to me that Agile is a derivative of the Toyota Production System, initiating the line of inquiry that I pursued for the rest of this research project.

This research was also undoubtedly informed by my positionality as a young white woman. Because I recruited participants in person or through an online platform that included my picture, it is safe to assume that participants likely read me as white and a woman, which may have impacted the kinds of rapport I was able to establish with participants, the kinds of stories they were willing to shares, and the ways in which they framed their experiences. Research shows that participants are typically most comfortable talking to researchers who may have shared experiences with them (Cabrera and Nora 1994; Fries-Britt and Turner 2002). I did not find nor interpret interviews with participants to be less transparent than those conducted with white women and felt that I was able to develop strong rapport with all of my participants. However, I want to acknowledge that I may have had difficulty recruiting more people of color as a white researcher, and that my positionality most certainly impacted the kinds of information that people of color felt comfortable sharing with me. I did find that people of color were much more open to discussing issues of race than white participants, though there were some exceptions. Women and non-binary people tended to be most comfortable discussing issues of gender compared to men. This is likely because categories like race and gender are more salient for those who are in marginalized positions, rather than those who are in positions of power. Additionally, those in positions of power may be reticent to cite privilege as part of their experiences of work. Moreover, at times, I got the impression that white men withheld or at least were not as forthright as others with critical or negative information about their jobs. Though this was not the case with all white men I interviewed, I found myself having to probe quite more than with participants from other racial/ethnic and gender identities, particularly around their concerns with workplace arrangements. This is perhaps a result of masculinity, in which male participants could have felt more strongly attached to narratives in which they held prestigious and positive job experiences than workers with more marginalized positionalities.

In general, participants viewed me as a student, and, rightly so, one with hardly any experience conducting software. However, I developed an expertise in Agile throughout the research process, purposely watching videos and reading management books on Agile to situate myself to the field in order to ask more appropriate questions. This knowledge of jargon, coupled with my positionality as a researcher interested in learning more, helped to facilitate relationships in which some workers were able to "teach" me about the software labor process. Workers were sometimes surprised that I was interested in what they considered to be the "boring stuff," and my fascination with the details of their work as well as my knowledge of Agile jargon provided space for them to go into depth about their work arrangements. Ultimately, the themes that emerged from the interviews that form the basis of this dissertation are themes regarding managerial control strategies, organizational structures, team-based relationships, and motivators and methods of resistance.

Online Materials: Agile

In addition to conducting interviews, I have collected data from 36 Agile-related videos. 22 of these videos came from a company called Agile Transformation, and 14 came from a company called Scrum Alliance. Analyzing free Agile videos on YouTube is relevant to this research because, as I discovered through in-depth interviews, participants rarely receive formal Agile training in their workplace. Participants are "trained" on Agile through word-of-mouth, receiving books from their manager upon employment (though any participant who received a book reported never reading it), or watching free videos online. Additionally, because this research aims to compare Agile as a theoretical methodology to Agile as it is practiced, it is necessary to collect information on "proper" usage of Agile from a source other than participants.

Because Agile is so popular there are many videos on YouTube related to it. I began my search for appropriate Agile videos by going to the Google search engine and searching for the following phrases, 'most viewed agile videos' and 'highest viewed agile videos.' In both cases, a website with the URL 'agilevideos.com' appeared first. Agilevideos.com is a website connected to the company Agile Transformation, which provides training, education, and consultants to companies who are looking for professional assistance in transforming to an Agile Project Management Organization (PMO). Users can pay for access to individual courses, access to a 12-week lesson plan, and/or a training course in the Project Management Institute's Agile Certified Practitioner exam.

Agile Training Videos has the highest number of subscribers of any Agile channel on YouTube and provides the largest library of Agile-focused videos. In addition to offering paid services, Agile Training Videos provides 50 free videos that cover a range of topics targeted for a range of audiences (like managers, team leads, team members, etc.). On their YouTube account, where all their free videos are located, there are 25 videos grouped into the following seven playlists: “PMI Agile Webinars,” “AgilityHealth Videos,” “Real Case Studies,” “Intro to Scrum Videos,” “Agile Simulation Videos,” “Agile Teams,” and “High Performing Teams Videos.” There are 25 free videos that are not associated with playlists.

I watched, transcribed, and coded videos. First, I opened a video on YouTube alongside a google document. I used the Voice Typing function on google docs to transcribe videos while I watched and took notes. I then watched the video for a second time, fixing any transcription errors as I watched. Transcription accuracy from Voice Typing varied depending on the quality of the video, the speed at which the narrator was speaking, and the number of speakers in the video. After fixing transcriptions, I coded videos using a codebook of 50 codes. Initially, the codebook had 30 codes that derived from the in-depth interviews, and I added 20 new codes based on topics that came up in the videos. I also watched, transcribed, and coded all free videos in the Scrum Foundations eLearning Series (n=14) provided by ScrumAlliance.org. I decided to include these videos in the data set because Scrum is the most widely adopted type of Agile. Overall, because I wanted to be able to compare Agile in theory to Agile as practiced, as well as develop a greater understanding of Agile to facilitate more in-depth interviews, it was necessary to collect data from Agile training videos and official Agile guides.

Online Materials: Tech Alliance

In addition to interview data on Tech Alliance participants, I draw upon Tech Alliance's publicly available internet resources. This includes descriptions of 174 events from January 2015 to January 2019, 10 blog posts, 25 "Learning Club" posts and materials, and two videos of panel talks posted on the Tech Alliance Facebook page lasting 1 hour and 54 minutes and 1 hour and 19 minutes, respectively. Data was converted to word documents and uploaded to Dedoose, a qualitative analysis software program. I coded for themes such as organizing focus, goal, or issue and audience. I argue that internet-based information is, in and of itself, a *type* of organizing strategy that typically aims to educate, connect, and spread awareness about a particular issue.

Chapter Outlines

This initial chapter has provided a framework and context for why one might engage in labor process theory research on the software development process. I have shown that the idiosyncratic and immaterial nature of knowledge-based work appears to pose challenges to traditional management methodologies, which have historically relied on direct control strategies that leave workers with little room to exert discretion over the way work is produced. Given the expert labor that goes into software development work, direct control strategies appear ill-suited to the creative and innovative nature of software production. Responsible autonomy, the second control strategy outlined by Friedman (1977; 1979), may operate as a more effective control strategy, where workers are given more autonomy and responsibility over the labor process such that their goals are aligned with those of the firm, and their work habits help the firm to expand surplus. However, as I have shown in this chapter, control strategies are typically adopted in constellations. Project management and project-based work has been a strong example of how to apply direct control strategies alongside responsible autonomy to develop an effective control regime. Still, little labor process research on software work has been conducted, with few studies exploring the role of Agile, a popular project management methodology, in shaping the way software is produced.[12]

I extend the research on critical project studies as well as scholarship on the software labor process through an analysis of Agile, a lean project management methodology. Over the course of this dissertation, I will explain how Agile operates, how it functions as a hybrid control regime applying elements of direct control alongside responsible autonomy, and how Agile draws upon elements of management that emerged in manufacturing industries to construct the strategy of Taylored Flexibility: an attempt to improve flexibility and calculability over complex and turbulent knowledge-based work.

Chapter II offers a longer description of Fredrick Winslow Taylor's scientific management and its application in Ford's factories, giving the reader a better understanding of traditional direct control regimes. The chapter also explains the crisis of Fordism, which led to the need for more flexible production in order to lower overhead costs and enlarge profits. As a response to the crisis of Fordism, I discuss the emergence of the Toyota Production System and the popularity of lean production that Toyota is known for championing. I also offer an intervention in the literature by describing the distinctions between Taylorism, Fordism, and Toyotism, by critiquing scholarship that frames Toyota as a distinctive break from Taylor, showing how scientific management was adopted by Toyota and combined with new strategies to create lean production. Here, I show the necessity of understanding control regimes and production processes as

[12] For an exception see Hodgson and Briand 2013.

bundled or drawing upon elements of various systems. I extend this intervention to suggest that an understanding of the evolution of production processes that highlights continuity rather than distinction can help us make sense of control regimes over knowledge-based work like software, which also draws upon manufacturing processes. I suggest that an investigation of the evolution of software production processes reveals the role that Toyotism has played in control over software work.

Next, I describe Waterfall, the dominant software production system until the 1990s, and draw comparisons to the crisis of Fordism, and then briefly describe the rise of Agile. The rest of the chapter introduces empirical data to describe the Agile production process, analyzing how it is a method used to improve calculability over complex work while also remaining flexible: the goal of Taylored Flexibility. I discuss how Agile adopts a Just-In-Time approach, and then offer the implications of JIT on software production, suggesting that it reveals a trend towards skill saturation (limits on workers skills). I also explain how Agile improves surveillance over the software labor process by quantifying knowledge work using scientific management methodologies. In doing so, I provide data and analysis that contradicts claims that direct control strategies are unsuited to knowledge-intensive work.

Chapter III describes the ways that commitment as a form of responsible autonomy is elicited through Agile. I make an intervention in the literature, though, by suggesting that commitment is not only elicited through ideological or discursive means, as much of the existing research suggests. Instead, I show the ways that commitment is embedded into the very production process, such that workers really have no choice but to commit to the team in order to get their work done or receive favorable reviews. In this chapter I describe some of the different theoretical conceptualizations of commitment in the workplace. Again, I intervene in the literature by showing that most scholarship does not recognize the ways that Taylor and Ford were thinking about commitment and provide some evidence of this by drawing directly from Taylor's writing and Ford's practices. I explain that commitment should not be conceptualized as a new strategy emerging in the late 20th century, but rather that managerial strategies have evolved to improve methods for achieving commitment, a goal that has existed since the emergence of industrial capitalism.

I suggest that Agile provides an effective methodology for eliciting commitment by embedding it into the labor process in three key ways: 1) making workers mutually accountable to one another, such that a worker's performance is assessed as their role on the team and not simply an individual, 2) task interdependence, the result of Just-In-Time production in which workers must take on each other's tasks in order to finish their own, and 3) encouraging co-location, or the physical coordination of workers in an office (again, an undertheorized aspect of control over knowledge-work), which helps to establish strong bonds between workers. These bonds improve what is called behavioral empathy, or the willingness of workers to help each other with their work. I suggest that these mechanisms help the firm improve flexibility and calculability, as workers willingly share their progress with managers, make commitments to getting work done that controls the pace of production, and quickly take on unexpected or last-minute tasks so that the firm does not waste workers' labor or time.

In Chapter IV I extend Acker's (2006) theory of inequality regimes by applying them to Agile, an example of a flat organizational structure, to show that ambiguities and lack of hierarchy in the organization and on teams, an especially an emphasis on flexible and porous job duties, leads to women, non-binary, and people of color to take on extra labor that is often unrecognized and uncompensated by the firm. I begin by showing the ways in which labor markets themselves are gendered, and that the evolution of particular control regimes are the result of gendered and racialized process. As such, I argue that the emergence of Taylored Flexibility can be explained, in part, by the fact that tech work is a highly masculine and white occupation. Then, I show the ways in which inequality regimes under Agile emerge, specifically looking at how flexibility and commitment has gendered and racialized outcomes. I argue that an emphasis on flexibility and commitment leads primarily women, non-binary, and people of color to take on extra and uncompensated labor in two ways: 1) by taking on unexpected tasks or job roles, and 2) by developing strategies to mitigate gendered and racialized tensions on the team in order to make the work of Agile possible. I suggest that the extra labor done by women, non-binary, and people of color is the result of gender-neutral assumptions in the organization, and is beneficial to the firm in that they do not have to account for inequality regimes. Moreover, firms directly benefit from the extra work of marginalized works as their unrecognized labor helps to maintain a smooth and efficient production process.

Chapter V focuses on instances of resistance and collective action, highlighting that despite the influence of Agile on the software labor process, Agile was not a primary motivator for worker resistance and action. I discuss the most widespread attempt at collective organizing in the tech industry through an analysis of the Tech Alliance. Tech workers have historically been cast as difficult to organize for a number of reasons: their high wages, alignment with employers, and lack of experience with unions, to name a few. While most tech workers remain unrepresented by unions, this Chapter highlights the new organizing strategies of the Tech Alliance, an informal organization of tech workers and allies aimed at building worker power in the tech industry. Drawing upon interviews with members of Tech Alliance and analysis of internet-based data, this section of the chapter discusses the strategies that tech workers are using to tackle labor issues and beyond in a 21st century context. Tech Alliance members are primarily motivated to organize because they see the products of their labor as political. They articulate organizing in the workplace—including cross-sector and cross-occupational organizing–as necessary to resist the production of technology used to support imperialism, xenophobia, discrimination and exploitation. In order to organize, though, Tech Alliance tends to favor social movement and anarchist organizing models over traditional union models, viewing unions as potentially too conservative, top-down, and limited for the types of radical organizing they deem necessary for the 21st century. Tech Alliance's alternative strategies include peer-to-peer education that merges academic research with on-the-ground praxis, in-person and project-oriented general membership meetings, coalition building and organizing "for the common good," special events, and digital communication for organizing. In sum, Tech Alliance's work reveals that organizing efforts in the tech industry remain vibrant despite a lack of union representation.

This dissertation concludes with Chapter VI, where I provide an overview of the arguments made in various chapters to explain the theory of Taylored Flexibility as a mode of control over knowledge-based work software. I discuss some of the limitations of this dissertation research and suggest lines of inquiry for future research. More specifically, I reflect upon the ways that COVID-19 has undoubtedly transformed the software labor process, raising discussing what kinds of research is necessary to capture these changes and their implications for software workers. I conclude the dissertation with a discussion of alternatives to the Agile software labor process, with a particular interest in advocating for a labor process that is more liberating and less exploitative of workers.

CHAPTER II

DIRECT CONTROL OVER SOFTWARE WORK

> There is always a fundamental tension between the need to gain cooperation or consent from those who do the work, and the need to force them to do things they do not wish to do, or to be treated in a way which is against their own interests, in order that the goals of those "in control" of the labour process can be achieved.
>
> Andrew L. Friedman (1985: 11)

"On February 11-13, 2001, at The Lodge at Snowbird ski resort in the Wasatch Mountains of Utah," Jim Highsmith writes in an excerpt from *History: The Agile Manifesto*, "seventeen people met to talk, ski, relax, and try to find common ground—and of course, to eat. What emerged was the Agile 'Software Development.'" Though Agile software development first appeared as such in 2001 after the online publication of the Agile Manifesto, it is best understood as the culmination of a number of "lightweight" software production processes that were sweeping the software industry in the late 20th century.[13] Like most industries at the time, organizational anxieties regarding how to respond to the growing digital and knowledge-based economy, and the flexibility that it requires for industries to remain competitive, drove software firms to re-think their organizational structures (Cobb 2011). The response was largely a "post-bureaucratic" turn, in which organizations increasingly replaced the iron cage of bureaucracy with "flat" organizational structures in order to mitigate hierarchies between workers and managers, more flexibly and quickly facilitate the production of innovative commodities and attract the intellectual labor of technical workers by offering them greater autonomy

[13] The Agile Manifesto can be found at https://Agilemanifesto.org/history.html

at work (Alvesson and Thompson 2004; Reed 1996; Storey 2005).[14] As one founder put it, Agile software development would be the key to "succeed in the new economy and move aggressively into the era of e-business, e-commerce, and the web."

The promise of Agile lies in its ability to allow firms the flexibility to be competitive in the new economy, while also maintaining the calculability and predictability needed to maintain organizational and labor control. As stated in one Agile training video, "organizations move to Agile because they want to improve predictability, they want to improve time-to-market, they want to increase ROI [return on investment] and value delivered, they want quality to go up, and the ability to respond to change." Agile helps firms to achieve this illusive combination of flexibility and predictability, consent and force, by taking a project-management approach. Project management methodologies have become an especially popular post-bureaucratic method to improve the productivity of knowledge-intensive work (Hodgson and Cicmil 2006; Balch 1994; Hamilton 2001; Hobday 1998; Packendorff 1995; Morris and Pinto 2004; Whitley 2006). Project work's popularity stems from its ability to, from a critical project studies perspective, operate as hybrid forms of control over work by emphasizing rationalization (or the constant improvements in the production process), calculability, flexibility, and cooperative team relationships (Briand and Hodgson 2015; Gleadle et al. 2012; Hodgson and Briand 2014; Hodgson 2004). Projects can be conceptualized as hybrid control systems because they contain elements of traditional direct control through deadline setting, specifications, and frequent meetings, as well as forms of responsible autonomy (also called ideological, cultural, normative, or "soft" control), achieved through High-

[14] See Chapter 1 for more discussion of the post-bureaucratic turn.

Commitment Management (HCM) schemes like worker autonomy and team-based work (Hodgson 2002; Kunda 1992).

While projects offer firms a methodology for organizing complex work, research shows there is still quite a disparity with project management in theory and the effective implementation of project management in the workplace (Williams 1995; Belassi and Tukel 1996; Morris et al. 2004). The managerial response to this problem has been to emphasize making improvements on project management methodologies (Hodgson and Cicmil 2006), an approach that I argue has helped to construct the emergence of Agile project management. In other words, I suggest that Agile is best understood as part of the rationalization process *within* the project management movement, the result of an effort at continuous improvement over project-based work. While Agile is a relatively new methodology, it is modeled on much of the production processes that emerged at Toyota in the late 20th century. Agile is described by founders as “an approach to managing software development projects that can be seen to have roots in lean manufacturing thinking and practice, derived in many respects from what is known as The Toyota manufacturing process” (Morien 2005:1). Agile improves upon the methodologies of project management primarily by adopting elements of Toyota’s Lean production system, an undertheorized aspect of scholarship analyzing the software labor process specifically, and the project-management and post-bureaucratic literature more broadly.

In this chapter, I follow Rowena Barrett’s (2001:31) reminder that “many aspects of the new can be found in the old if you only look in the right place,” to briefly investigate Frederick Winslow Taylor’s scientific management methodology as well as the Toyota Production System to show the relationship between the two, and their influence on the contemporary software labor process. Using Taylorism and Toyotism as a framework, I put a name to the hybrid control regime used to organize the software labor process, arguing that Agile operates as a strategy I call Taylored Flexibility: an attempt to achieve combinations of flexibility and predictability in the most profitable manner. In this way, this research contributes to labor process theory approaches that urge scholars to analyze the complementary and, at times, contradictory managerial strategies that are simultaneously deployed in order to achieve control over work (Sturdy et al. 2010: 114; Callaghan and Thompson 2001).

Over the next three chapters I will explicate the elements of Taylored Flexibility, showing how it operates as a hybrid control regime that relies on both direct and normative aspects of control like responsible autonomy. Moreover, I show the ways that Agile relies on some of the principles of scientific management, to reveal continuities in the methods of control used to organize knowledge-based work and manufacturing work. In this chapter, I focus primarily on *direct* forms of control in the Agile methodology, leaving an analysis of the combinations of direct and normative forms of control to Chapter III and IV. I begin the chapter by offering brief histories of Taylorism and Toyotism, in order to make a case for use of Taylored Flexibility as a conceptual tool to analyze lean software development methodologies like Agile. I describe transformations in software development processes that have led to the popularity of Agile, and the implications of Agile on the software labor process.

Next, I draw upon empirical qualitative data to analyze the direct control strategies that Agile uses to achieve Taylored Flexibility, primarily through the reorganization of software production from a sequential to a Just-In-Time (JIT) process. I argue that Agile's use of JIT leads to highly fragmented work, increases task interdependence, and reveals a trend toward a saturation of workers' skills.[15] I also show how Agile uses elements of Taylor's scientific management to overcome the "invisibility" problem in software production work, or the notion that because knowledge-based work takes place "in the minds" of workers it remains insulated from managerial view, and thus outside the realm of direct managerial control. Ultimately, this chapter reveals the continuities between manual and knowledge-based labor, raising questions about the relationship between knowledge, technology, management, and labor in the Digital Economy.

A Brief History of Taylorism

One century ago, Fredrick Winslow Taylor published *The Principles of Scientific Management* (1911), one of the most foundational and influential texts to be written about management methodologies in the context of a growing industrial capitalist economy. In this seminal text, Taylor argued:

> The best management is a true science, resting upon clearly defined laws, rules, and principles, as a foundation...the fundamental principles of scientific management are applicable to all kinds of human activities, from our simplest individual acts to the work of great corporations (7).

As shown in the quote above, Taylor argued that the best way for businesses to improve productivity in the era of what Braverman called Monopoly Capitalism, was through the use of a scientific, rationalized, and profit-oriented approach to organizing work where efficiency and productivity could be improved by "breaking down each labour process

[15] Task interdependence refers to the presence of interrelated tasks necessary to complete a project (Kakar 2017) and the extent to which team members expect that they will personally benefit by contributing to team goals (van der Vegt et al. 1998).

into component motions and organizing fragmented work tasks according to rigorous standards of time and motion study" (Harvey 1990: 125).

Taylor's scientific management approach introduced four principles to make work more productive: 1) the use of science, or the replacement of managerial subjectivity by objective measures of work processes. 2) Harmony, not discord, emphasizing that the relationship between workers and employers should be cooperative, symbiotic, and harmonious. 3) Cooperation, not individualism and 4) Development of Every Person to his Greatest Efficiency, or the application to the scientific measure of the most efficient and effective work processes to each worker.[16] By applying these principles, Taylor aimed to remove the issue of "systematic soldiering," or the "deliberate and organised reduction of work pace by groups of workers" that had become a common worker strategy to limit their exertion of labor power as a form of resistance to the capitalist production process (Friedman 1977: 92). With these principles at the forefront of his approach, Taylor emphasized the grouping of individual craft workers into a factory setting, such that they could act as one machine of workers rather than disparate parts, resulting in tighter control by managers over the labor process.

Scientific management appeared as one of the dominant managerial strategies in the 20th century, solidified by its successful implementation by Henry Ford, who also introduced the moving, continuous assembly line. While Fordism has been most commonly characterized by this introduction of the assembly line, scholars have noted that it was, in fact, the introduction of "the complete and consistent interchangeability of

[16] Already in these four principles, especially principles 2 and 3, we see Taylor's interest in eliciting a cooperative relationship between workers and managers, which I analyze in greater depth in Chapter III.

parts and the simplicity to attach them to another" that set the Fordist production system apart from ones that came before him. By establishing interchangeable parts, Ford was able to produce goods on a massive scale, dropping the costs of production and allowed profit margins to soar (Womack et al. 1990: 27). Moreover, following Taylor's aim to simplify the "horizontal flow of production processes," Ford helped to degrade the skilled labor of craft workers as he "perfected the interchangeable worker" (30), a laborer who was tasked with simple jobs that demanded the worker stay physically put at one section of the line, who's pace was determined by the speed of the machine (Harvey 1990). In this sense, Ford was able to achieve the primary goal of Taylorism, "control over the labor process...into the hands of management, not only in a formal sense but by the control and dictation of each step of the process, including its mode of performance" (Braverman 1974: 100).

Yet, by 1955 mass production was increasingly adopted in countries across the world, threatening the competitive advantage of Ford's factories in the United States (Womack et al. 1990). By the early 1970s, manufacturing in the United States would decline by seven percent while employment in service sectors would account for half of the job growth between 1979 and 1989 (Plunkert 1990). Scholars generally agree that the Fordist model began to reach its limits towards the latter half of the 20th century, exhausting productivity gains, rife with conflict between workers and employers, and unable to respond to external pressures arising from the growth of new technologies and finance capital (Thompson 2003). These circumstances created the Fordist crisis, raising questions about the best methods of organizing production that would lead to the rise of more flexible and allegedly worker friendly “post-Fordist” processes (Piore and Sabel 1984).[17]

The Rise of Lean Production

The most infamous of post-Fordist production processes is the Toyota Production System (TPS), an alternative to Fordism that was initially developed by Taiichi Ohno in the mid-20th century. Toyota faced unique barriers to pursuing Fordist-style mass production, as Ohno would come to realize after surveying a number of Ford’s American factories. For one, Japan’s domestic market demanded a wider range of vehicles, but Japanese firms had limited access to capital and foreign exchange in the post-war era (Womack et al. 1990). The Japanese labor market also differed from the United States in that Japan employed few migrant workers, strengthening the bargaining power of domestic workers who were beginning to demand more favorable working conditions in exchange for the toil of manual manufacturing labor. In order to compete against American automobile corporations and attend to growing worker discontent, Ohno would implement a new way of producing automobiles and managing production which would come to be known as lean production (Holweg 2007).

[17] While some view post-Fordism as a particular break away from Fordism, characterized by the rise of more flexible production practices sometimes referred to as flexible specialization, others have noted that despite the emergence of new technologies, increased reliance on High-Commitment Management schemes, and more small-batch production, “it is more important to identify the underlying continuity than to over-emphasize any supposed history break. Failure to do this leads to a number of problems. One is an attempt to seek a single path towards which firms and sectors are developing, rather than seeing a variety of options from which they can choose according to the dictates of profitability” (Bramble and Fields 1992: 567). I discuss in the following section, and argue throughout this dissertation, I do not view production processes such as Toyotism as characteristic of a new mode of production, altogether distinct from Fordist practices, rather, I argue that new strategies emerge in hybridized forms, taking the old processes with them.

Lean encapsulates a suite of production practices that originated in Toyota's factory aimed at a primary goal: eliminating *muda*, or waste (Liker 2004). According to Ohno (1988: 19 - 20) there are seven types of waste:

> defects (in products), overproduction of goods not needed, inventories of goods awaiting further processing or consumption, unnecessary processing, unnecessary movement (of people), unnecessary transport (of goods) and waiting (by employees for process equipment to finish its work or on an upstream activity) emphasized low inventory, minimal waste, and total quality control.

In order to eliminate waste, TPS moved away from the high-inventory Fordist model and instead pursued a lean Just-In-Time (JIT) approach. Rather than invest unnecessary capital in purchasing large quantities of inventory that may or may not reach the customer, JIT is "a set of principles, tools, and techniques that allow a company to produce and deliver products in small quantities, with short lead times, to meet specific customer needs" (Liker 2004: 23). Rather than mass producing the same model automobile, Toyota would produce small batches of specialized products for niche markets, enhancing consumer choice along the way.

TPS also achieved minimal waste by implementing Total Quality Control (TQC) mechanisms that identify defects throughout the production process, rather than at the end of production once labor had already been spent. TQC can be achieved by the sixth principle of the Toyota Way, which states that "standardized tasks are the foundation for continuous improvement and employee empowerment." Thus, while TPS aims to embrace flexibility more than Ford's mass-production line, Toyota's workers still must follow standardized procedures to work efficiently and productively. Fujimoto (1999: 19) highlights three of these standard procedures for achieving TQC, which include routines for problem identification, routines for problem solving, and routines for solution retention.

Problem identification took a particularly novel form at Toyota, with assembly line workers having the power to "pull a cord just above the workstation to stop the line if any problem is found" (Womack et al. 1990: 79). Clear visibility over production and especially defects is a key element of TQC, as workers are instructed to diagnose and resolve production problems immediately in order to eventually see a total elimination of defects. In fact, the seventh principle of The Toyota Way states that one should "use visual control so no problems are hidden." Lean production aims to progressively refine the production process through iterative learning developed through these standardized TQC procedures. In other words, lean production relies on consistently determining best practices and adjusting errors immediately rather than waste materials and labor.

Despite much excitement about the novelty of Toyotism, Fujimoto (1999: 50) importantly points out that TPS was not an entirely unique system, but that "Toyota's production organization adopted various elements of the Ford system selectively and in unbundled forms and hybridized them with their ingenious system and original ideas." Because lean production is rooted in the expectation of iteration, change, flexibility, and specification, Agile, like Toyotism, is often not "purely original or totally imitative. It is essentially a hybrid" (Fujimoto 1999: 50). While the Toyota Production System has been heralded as a distinctive break away from traditional scientific management or Fordist production, TPS can also be analyzed as the resurgence and reinvigoration of forms of direct control alongside an emphasis on flexibility and a greater effort at establishing responsible autonomy. Though TPS was drawing most directly on Ford, the influence of

Taylor on Ford's own practices cannot be overstated, requiring a recognition that scientific management has had influence on lean production. Drucker (1999: 81-82) sums this up well:

> Every method during these past hundred years that has had the slightest success in raising the productivity of manual workers -- and with it their real wages -- has been based on Taylor's principles, no matter how loudly his antagonists proclaimed their differences with Taylor. This is true of 'work enlargement,' 'work enrichment,' and 'job rotation' -- all of which use Taylor's methods to lessen the workers' fatigue and thereby increase the workers' productivity. It is also true of such extensions of Taylor's principles of task analysis and industrial engineering as Henry Ford's assembly line (developed after 1914, when Taylor himself was already sick, old, and retired). It is just as true of the Japanese "Quality Circle," "Continuous Improvement" (Kaizen), and "Just-In-Time Delivery."[18]

Ultimately, lean production often maintains standardization and managerial control by narrowly determining the types of products that workers will produce and delineating clear expectations for when work should be completed. As Braverman (1974) noted about new management strategies,

> The reforms that are being proposed today are by no means new ones...They represent a style of management rather than a genuine change in the position of the worker. They are characterized by a studied pretense of workers "participation," a gracious liberality in allowing the worker to adjust a machine, replace a light bulb, move from one fractional job to another, and to have the illusion of decision making by choosing among fixed and limited alternatives designed by management which deliberately leaves insignificant matters open to choice.

For example, in his study of six US Manufacturing plants, Vidal (2006) finds that lean production itself can be successful in unbundled forms. In many cases, firms may determine that they are "lean enough," or able to implement elements of Just-in-Time and Total Quality Control that, while not fully representative of all practices under TPS, still

[18] Interestingly, this citation is listed on the Lean Agile Training's website resources of recommended articles https://leanagiletraining.com/lean-agile-scrum-resources/articles-we-recommend/

allow firms to enhance productivity (Vidal 2017). In his study of Japanese firms located in South Wales, Danford (1988) argues that Toyotist production practices are hardly a break from traditional Taylorism. He suggests that the workers at the lean production plants he studied "have become more completely subordinated to the supervisor, to the machine and to the intensified pace of production" (61). Especially when firms emphasize scientific rationality and productivity, instituting substantial workplace change where teams of workers are empowered to make decisions or enhance control over the labor process is unlikely (Vallas 2003).

In recognizing the continuity in production processes since the emergence of industrial capitalism, I suggest that an understanding of the principles of Taylorism and Toyotism can shed light on contemporary production processes used to organize knowledge-based work like software production. I dispel the myth that in the context of software work "the method by which surplus value is extracted and accumulated holds little resemblance to nineteenth or twentieth century manufacturing...due to the unique nature of the commodity, software, which is the result of knowledge work" (Andrews et al. 2005: 71) by offering an investigation into Agile, and its relationship with Toyota and Taylor. Indeed, I show that manufacturing processes have been central to the evolution of the production process used to extract surplus labor from knowledge-intensive workers. Below I describe a similar evolutionary process that took place in software development work to the transformations in manufacturing, namely, the transition from Waterfall to Agile.

From Waterfall to Agile

Up until the late 1990s, software development work was typically organized through a linear, hierarchical, and top-down project management methodology called Waterfall. Waterfall is considered to have been first used at IBM in the 1960s to develop the large-scale System/360 project, though it became more popular in software development work as software occupations grew (Cusumano and Smith 1995). Waterfall is named after its sequential nature, meaning that each step of production relies on the success of the former stage, much like the Fordist system. The products that workers produce are determined by strict and detailed contracts between the firm and customers, in which the parties agree upon a "fixed scope," or a final product outcome, while relying on flexible estimates for cost and delivery time. Waterfall projects can last from months to years and have ultimately been perceived as incompatible with the quickly evolving nature of software production, making it difficult for firms to respond to change or unexpected error that may lengthen the timelines or cost of production. Cusamano and Smith (1995: 3) detail this pitfall clearly, explaining

> In cases of uncertain requirements or fast-moving markets, if designers try to create detailed specifications for the product and its pieces too early in the development cycle, the project team will end up building a product that does not meet customer needs very well or that is out of date before it even ships. If developers try to make changes in parts of the product as they go along, for example, due to interim feedback from customers or evolution in particular hardware or software technologies, or even just to add a feature that a competitor has just introduced, then the project may end up with pieces that no longer fit together. The integration effort and system testing fail. The project team then has to rework the pieces extensively -- even though they thought they had finished coding. They may even have to throw much of the code away. With these types of difficulties, it is no surprise that so many software projects end up being late, over budget, and riddled with bugs, due to errors in the pieces as well as in how the pieces interact.

As described here, despite being the norm in the software industry for decades, Waterfall can be a particularly wasteful production process.

The interviews I conducted with software managers and workers reflected, at best, a disinterest in Waterfall and, at worst, a disdain for it. Linda, a Scrum Master, critiqued Waterfall's negative impacts on code quality, citing a similar issue that Toyota criticized Ford for committing: a production process that does not have an effective quality control component embedded within the process, leading to waste. As a sequential production process, quality control is an ad hoc component of production in Waterfall. Linda explained,

> Waterfall is just, write code after your plan and then release. And when you do that, generally what ends up happening is you didn't write half the stuff that they [the client] ask for, the other part you wrote wrong, and you might get one quarter of the features right cuz there's no check-in or looking back or iterating through.

As a result of Waterfall's sequential, bureaucratic, and rigid process, this particular project management methodology puts firms at high risk for causing client dissatisfaction and waste. By the early 1990s workers and management alike began to view Waterfall methods as "heavily laden with documentation and were perceived as cumbersome, ineffective, and extremely bureaucratic" (Cobbs 2011: 5), demanding a new approach to software production.

In resistance to the traditional Waterfall method, E.A. Edmonds proposed "adaptive software development" in 1974, which became increasingly popular in the 1990s as more and more "lightweight" software production methods emerged (Edmonds 1974; Fitzgerald et al. 2013). Lightweight software production, like lean manufacturing and other post-Fordist or post-Bureaucratic methodologies, emphasized eliminating waste and enhancing firm flexibility to respond to change at lower costs. The Agile Manifesto, which was published online in 2001, aimed to synthesize disparate lightweight production methods and provide an overarching philosophy of lean software development.[19] Lean software principles include the goal to "amplify learning, decide as late as possible, deliver as fast as possible, empower the team, build integrity in, and see the whole" (Poppendieck and Poppendieck, 2003).[20] Below, I draw upon empirical data to describe the ways that Agile incorporates a JIT production process through the use of "variable scope," "sprints," and the principle of "maximizing the work not done," which I define and explain in more detail below, and which have been absent from much of the recent literature on software production processes. In doing so, I hope to demystify the software labor process and dispel some of the myths regarding the organizational structures and managerial control strategies that firms use to organize the labor process and improve profits in knowledge-based occupations like software production. I show how these mechanisms can be understood as a form of Taylored Flexibility, or a combination of calculability and agility, which, I argue, leads to a saturation of workers

[19] Agile is often used as an umbrella term to describe lightweight methodologies, particularly Extreme Programming (XP), SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development and Pragmatic Programming,

[20] The twelve principles behind the Agile Manifesto, which can be accessed at https://Agilemanifesto.org/principles.html, read as follows: Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. Business people and developers must work together daily throughout the project. Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation. Working software is the primary measure of progress. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. Continuous attention to technical excellence and good design enhances Agility. Simplicity--the art of maximizing the amount of work not done--is essential. The best architectures, requirements, and designs emerge from self-organizing teams. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

skills, compromises in the quality of code produced, and heightened surveillance and calculability over software-workers' knowledge-based work.

Just-in-Time Production in Software

As introduced above, a primary way that Agile reorganizes the software labor process to be more flexible, calculable, cost effective, and productive, is by applying lean principles and processes to software production. Agile, like lean, makes eliminating waste a primary goal.[21] A primary way in which Agile eliminates waste is through the adoption of a JIT production process, in which teams deliver some working component of the final product to the customer *incrementally*. The production process is organized into project "sprints," two-week time increments in which the software development team is working on some portion of the final product. This completed portion of work is referred to as "Done" work, or a component of the product that is in usable condition for the customer. As Denise, a Director of Engineering explained, "It's important to know when your work is done. If it's never really quite done productivity is just going to go down." By delivering value incrementally, Agile allows firms to minimize the likelihood of client dissatisfaction, engaging in a flexible production process in which the goals can be adjusted from sprint to sprint. In other words, the sprint is one component of Taylored Flexibility, allowing firms to be both more flexible and predictable: if the customer decides they would like something different the software team can quickly pivot and tailor the product to more closely align with the customer's request, all while the firms

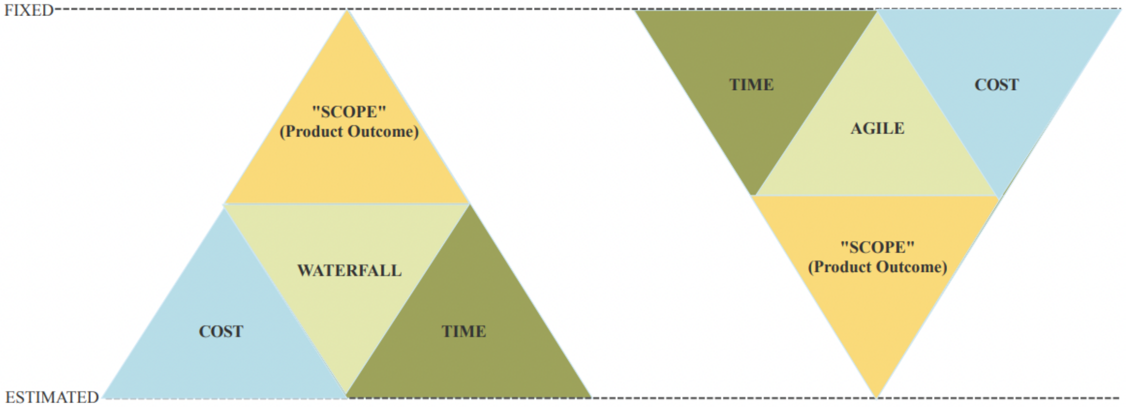
[21] Poppendieck and Poppendieck (2003: 3), a well-known Agile proponent and practitioner, applied Taiichi Ohno's seven types of manufacturing waste to software development, developing the following scheme: "overproduction = extra features, inventory = requirements, extra processing steps = extra steps, motion = finding information, defects = defects not caught by tests, waiting = waiting, including customers, transportation = handoffs."

can rest assured that the product team is never wasting more than two weeks on some component of the project that might be changed.

The team is able to change the product because in the Agile JIT approach, the relationship between cost, delivery timelines, and scope is transformed. To refresh, the Waterfall method uses fixed scope, or a determined final product, and adjusts deadlines and costs throughout the production process. In order to minimize the potential waste associated with missing or pushing back deadlines, or losing customers as costs rise, Agile uses a model in which cost, and deadlines are fixed and scope (the final product that will be delivered) is variable (see Figure 1). In other words, customers may not know exactly what product they will be getting at the end of a project, just that the product will meet their needs while remaining on time and on budget. As Varsha, an Enterprise Agile Transformation Leader explained, "If you want to have the same delivery date in Agile, the delivery model is that your resources and delivery dates are the same, that is fixed, but the scope is

variable."

Figure 1



Because agreements with customers under Agile mean that scope is variable, firms have some discretion in how to go about solving the software problem that the client has brought to them. Yet, as I discuss below, along with this newfound flexibility, Agile allows the firm to direct the activities of the team to heighten employer control over the production process, part of the regime of Taylored Flexibility. This is primarily achieved through the principles of maximizing the work not done, delivering as fast as possible, and through the process of breaking down the work.

Maximizing the Work Not Done and Delivering as Fast as Possible

Under Agile, software problems are solved by following the principles of "maximizing the work not done," encouraging the team to only build j*ust enough* to satisfy the customer. Steven, a Data Architect, described how these principles operate in the Agile production process by giving an analogy of workers trying to develop a vehicle for a client:

> So, an Agile methodology is the opposite of what we would call a Waterfall methodology, which has been the practice of age for many years. Which is to say we're going to start at square one. So [the client] wants a car. Okay we're going to build a car. And until we deliver the final car, everybody's going to be unhappy along the way waiting on the car to come through. And that's a big undertaking, to build a car! So Agile methodology will start with a skateboard. And we'll give [the client] a skateboard and say "how does that feel? how does it ride? Does it take too long? Let's go back to the drawing board and figure out how we can turn that skateboard into a bike." And at some point there, you might have a fit for need, right? The bike is good enough! The bike gets [the client] there on time; the bike can handle the road. And we have it *right now.* Stop process right here; we're good to go.

As Steven describes, implementing a JIT production process allows the scope of the project to vary, offering firms more flexibility to respond to changing client demands or new technology developments and to minimize waste associated with rigidity and sequential production. At the same time, firms are able to maintain predictability over the costs of the project, as those, along with timeline, are fixed.

One way that Agile ensures that workers are not overproducing cars when all the customer needs is a skateboard is by determining the average time that a team should be working on delivering a component of the final product, and taking a Minimum Viable Product approach. Minimum Viable Product came out of an organizational methodology called Lean Startup, which explicitly draws upon lean manufacturing methodologies and applies them to the startup sector. Agile project management also adopts the Minimum Viable Product approach, which means that software workers are encouraged to not create any extra features or components of the software that clients did not explicitly ask for, even if this could improve the product, as I discuss in more detail below in the section on skill saturation.

Along with the principles of maximizing the work not done, Agile also encourages teams to "deliver as fast as possible," which can include quickly delivering to the customer or being first to market. This demand to deliver as fast as possible, I argue, limits the amount of time a technology team can spend on a product, and ultimately the quality of the product that is released. As Nathan explained, firms reach a limit to widening their profit margins through the cost of labour and production and prices. Nathan argued,

> It's the technology that's giving us the competitive edge and in order to deliver those features, we have to build things really quickly and try to be ahead of our competitors because everyone is probably, has the same set of ideas but maybe just in a different order. It's like Apple or Samsung with the phones. Everyone has the same ideas. It's just whoever can get it out first is usually the winner and then you can refine it later.

Though Nathan initially argues that the technology or "concept or whatever makes you stand out" provides firms with a competitive edge, he goes on to provide a contradictory explanation that, in fact, firms often have similar ideas so quickness to market is where the real advantage lies. Blair, a Lead Product Manager, explained that if his team is assigned a project that has a 2-week time commitment, the team will determine "the very very very bare minimum that we can ship for this and actually meet our requirement or meet the goal. And then we plan for iterating it or following up later."

Through this JIT and fast approach, Agile helps firms to achieve Taylored Flexibility, such that they are able to meet the customers' needs while also building in the option to refine the product later. In this sense, teams prioritize quickness and simplicity over taking the time to build particularly innovative products. As Carrie, an Engineering Lead, put it, "the point is to meet the needs of the business problem that came in the door. Which may not be actually the best product they can make, just the fastest product they can make." Ultimately, working in this manner may lead teams to build up "tech debt," or, as Endah, a Programming Lead explained,

> something where you built something in the past and it basically comes back to bite you because you didn't implement it correctly or it just, sometimes it's just old and stuff stops working so it might be something like, you used technology that's no longer supported and now you have to upgrade or you um, did sort of a like, we would call like a "hacky" solution or a workaround where instead of implementing something the right way you implemented it the quick and easy way and now there's bugs or something.

Stewart, a Web Developer, argued that organizations that heavily prioritize the "deliver as fast as possible" element of Agile are characteristics of "bad" workplaces. He explained that while his current firm was willing to give workers the time necessary to ensure they weren't accruing technical debt, tech debt still happened as a result of team members who had previous experiences at companies who “did these practices,” creating a “hostile workplace.” Stewart explains,

> you can get a lot done really fast but you’re accruing tech debt so it’s stuff that will just cause problems later, or will just take somebody else longer to do later which should just be really simple. They were making a lot of tech debt for themselves that they would just have to clean up later, and essentially really only doing it to show off to their client which is necessary, but I don’t know, yeah.

As Hyman (1987) suggests, the goals of the enterprise are necessarily conditioned by capital, leaving organizations to primarily pursue short-term strategies. Thus, in order to remain both flexible and adhere to the demands of clients or customers, as well as quickly develop products, Agile adopts a production process that allows firms to secure capital driven by client satisfaction, as well as organize work in such a way that responsible team members add value to the final product by adhering to lean production methodologies, rather than use their creative skills.

In one case, a worker on a team was moved off a project as her passion for the product curtailed her ability to meet short-term deadlines. Because deadlines are fixed in Agile, not meeting a deadline can result in the loss of a client or increased costs. This is precisely the purpose of flexible scope, to allow the firm the ability to produce the product that requires the least labor time. Endah explained that on one project she was working on a teammate was consistently behind schedule for her daily tasks:

> Endah: we had a particular developer who was very very attached to one specific application [chuckles] and she like, [laughs] it’s one of those things where she was such a perfectionist about it and it was like, we were like, we just need to get this out the door, like you need to stop working on it. And we actually, she got pulled off the project at the very end because they were basically like ‘somebody else is going to finish this and get it out the door because you're spending too much time on it’
>
> Larissa: And do you remember how she responded to that?

> Endah: Yeah, she was not very happy
>
> Larissa: And what did you think about it?
>
> Endah: It’s funny now thinking back but at the time it was not funny. I mean, I think it was valid because like, I think it was valid for her to be angry, but I think it was also valid to pull her off because she really was like, it was kind of like one of those things where it was her baby and she was super attached and it was, we gotta just get this out like we can’t make it perfect. So, I see both sides, but I think maybe the way it was done was a little more like, surprising for her like she, I think she just felt like, “are all these people mad at me? And they don’t want me on this project? Like, what is going on?” I think it could have been done more tactfully.

While it became clear to the leaders on Endah’s team that her colleague must be taken off the project in order to meet deadlines, the teammate was confused about how her desire to improve the quality of the project could have resulted in her transfer to another project. This account reveals the short-term priorities that have become so popular especially in late and neoliberal capitalism (Harvey 2005), which have not escaped the logic behind the production of knowledge-based commodities, despite common perceptions that knowledge production requires a particular kind of innovation. While software workers certainly have highly developed skills to solve the problems required of their occupation, the drive for quick profit appears to take precedent over quality or creativity.

Importantly, readers should note that the emergence of programming languages in the mid-20th century has already been theorized as a methodology for degrading the software production process. As Barrett (2001:25) writes, the introduction of structured programming language meant that software developers did not necessarily need to know complex math, which contributed to the movement of software development “from being a craft activity to a more conventional industrial production process. As such it reduced the originality involved in software programming (Friedman 1990). It did this by stressing orderliness, simplicity and economical use of standard languages and code." As described in the following sections, this existing emphasis on orderliness through languages combined with lean production processes has particular implications for workers skills.

Breaking Down the Work

Another way in which Agile contributes to orderliness and the minimization of waste to achieve JIT production is by breaking down into its smallest deliverable chunks, determined through what is called "user stores" or "feature requirements." User stories determine the software features that will offer the most value to the customer, providing clear expectations for what developers should be aiming to produce. User stories typically follow the format, "as a [type of user] I want [some goal] so that [some benefit]."[22] An example of a user story could read, "as a student, I want to buy an online parking pass, so that I can park at school." The user story tells the developer that they need to create software that allows the user, in this case, the student, the ability to buy an online parking pass. User stories can offer software developers a wide range of control or limitations over how to produce software, depending on how strict or open-ended the user story specifications are written. While the sprint provides the firm flexibility, user stories provide the firm with the predictability it needs to achieve Taylored Flexibility, operating as a direct control strategy over the software team's labor.

Ultimately, the detail of the user story is part of a fragmented production process. One Agile video describes this process as part of the Agile lifecycle, or the flow of the production process. The speaker explains

[22] http://www.Agilemodeling.com/artifacts/userStory.htm#Introduction

> The very first part of our agenda is going to be Visioning so understanding what it is that we're trying to accomplish. What's the purpose of the project? And then we're going to go through and brainstorm the requirements, now that we've visioned, what are the actual stories that we have to deliver here? Then we need to slice some of them down. So, a lot of times we'll have stories, but they'll be "epics," so they'll be too big. So, we'll slice them and break them down into smaller chunks.

This process of fragmenting work was a key theme that emerged when I asked participants to generally reflect on how Agile organizes work. Carrie, an Engineering Lead, explained under the Agile methodology,

> you break coding problems into 16 smaller problems, and take those 16 problems and break those into 10 even smaller problems. So tiny machines are built correctly, and you put it all together and you get the big machine that solves the problem.

A number of participants reflected on how breaking down work into manageable chunks contains the amount of interpretation or uncertainty that goes into solving a coding problem. As I analyze below, a consequence of Agile's JIT production process is a closure in the spaces of play in the development of software. Closing spaces of play, as Aneesh (2001) argues, results in skill saturation, or a minimizing of the "extent to which the new skills have spaces for play embedded in them" (Aneesh 2001: 364). Thus, these findings raise questions about trends towards skill saturation of expert knowledge workers as a result of a JIT project management methodology.

Skill Saturation as a Result of JIT

While literature has shown that software workers do experience some autonomy in the choices, they make about how to assign tasks to one another and how they may go about solving a technical problem, these choices are often constrained by a company's demands, and may operate more as illusions than genuine autonomy (Damarin 2006, 2013; Briand and Hodgson, 2014; Barrett,2004). In the case of Agile, the desire to minimize waste acts as a constraint upon software workers' autonomy and operates as a closure of spaces of play in the technical skills needed to solve a software problem (Aneesh 2001). Waste in software production primarily stems from "gold plating," or overproduction, in which software workers build extra, unplanned, or unwanted features that increase the cost of production and extend the timeline for project completion (e.g., building a car when the customer could have been happy with a bike) (Poppendieck and Cusumono 2012). In other words, Agile narrows the amount of interpretation, uncertainty, or discretion that may go into solving a coding problem.

Melanie, a Software Manager, reflected on the relationship between firm constraints and worker autonomy when I asked her about how much autonomy teams had over technical aspects of work. She explained,

> So, kind of how that happens is like the big decisions trickle down to me and the product owner and we give the team chunks of things to work on. So, we would identify three pieces, say, and tackle them one by one. So, let's say we're building a textbox. The team would go off and choose the technology and there's like a designer who would tell them how it should look and work, but they would decide how the code should work.

Though Holly began by trying to highlight developers' creative autonomy over little decisions, she continued to reflect upon the limited choices that developers are able to choose from. She continued,

> but then like, even the technology decisions, to a certain extent, can be kind of constrained. Where it's like 'hey you can either build this in Ruby or Java.' You can't just go off and be like 'hey, I just heard about this brand-new programming language and I want to build it in that.' So, I guess even their decisions are constrained by the big, you know, company constraints.

Nate, a Product Owner and Agile Coach, also discussed the impacts Agile's JIT approach has on worker creativity, stating, "Agile is like you're really chunking up your project or your software into small pieces so you can deliver some software or feature or value to the business, so that means *you're kind of pacing the developer's creativity*." While Nate explained that he felt software workers still had some opportunity to exert creativity, his recognition that Agile attempts to pace developer's creativity reflects, I argue, a direct control strategy towards a closure of spaces for play.

Play closure is especially consequential for developers working on products for a client, rather than for their own firm. Endah, a Lead Developer, explained that "for some of the teams, especially if they worked on externally facing applications [for clients], they might have standard stuff for what they needed to use. Like color schemes or technologies, yeah." Managers in particular noted that software workers who desire to incorporate opportunities for play in the software development process could pose problems to the firm. For example, Nate stated further that 'a lot of times engineers just gravitate to gold plating. Where they're like, "what if we do this?" "what if we do that?"' Nate continued to explain that when engineers excitedly offer 'gold plated' coding options to solve software problems, he must remind them, '"well, the business stakeholders didn't ask for this." We really narrow down the scope of what we need to do for this feature, so it's just, it's just kind of controlling how much they have to create.' Sandy, a Senior Project Manager, also described the challenge of compelling workers to do work that was not particularly interesting. She explained:

> Engineers like technical challenges, they like working on things that they think are interesting. And what the org thinks is interesting is not always what's interesting to engineers. So right now, one of our things to do is really boring. It's like data entry but for programming. So, a lot of the time engineer's complaints are around boredom or repetition. But sometimes, the metaphor I use is eating your vegetables or brushing your teeth. Nobody likes it, but it's something you have to do.

While Sandy and Nate recounted having to align workers more closely with the goals of the firm rather than their own personal desires for more interesting work, Stewart actually stated that he appreciated the clear guidelines for product requirements offered through Agile:

> You have to be a lot less creative when you receive a very focused ticket [requirement] that says exactly what to do in basically every way. It really helps you work through things. Like, if you're working on a personal project, you can do whatever, but you're not. You're making this for a customer.

Stewart's statement reflects one of the consequences of skill saturation that Aneesh (2001: 379) describes, a "high level of predictability" that "affords greater control and ease for task execution." This statement also contradicts assertions that knowledge-intensive firms are particularly dependent upon their workers' creativity, creating a new mode of production in which employers can assert limited constraints on worker activity (Andrews et al., 2005)

Still, as Barrett and others pointing out, workers may experience *some* technical autonomy. Yet, as I have shown, the production process in which work is fragmented into short timescales and clearly defined features operates as a mode of direct control which limits the workers ability to exercise creative autonomy in the workplace. In fact, one worker, Alan, rejected the idea that software work is creative. When I asked the 25-year-old Software Engineer if he feels like he gets to be creative at work, he responded, "not really because it's not really my MO. Software doesn't have room for a ton of creativity. I think there are creative solutions to problems, but in a very binary process."

These findings raise questions about how the normalization of JIT production processes like Agile may contribute to new expectations about what the work of software development involves, and the role of autonomy and skill in that work. While scholars have noted cases of workers being resistant to project management, as it acts as a constraint upon the autonomy they expect to have as professional knowledge workers (Gleadle et al. 2012; Case and Piñeiro 2008), participants in this study did not overwhelmingly raise concerns about Agile's trend towards skill saturation, as illustrated above. However, participant's experiences did suggest some evidence that software workers may leave jobs to find more interesting work, with Arnold stating that he left his former firm because "they basically had us doing the same thing every single day, or not every single day, but on a cycle of two weeks...I wasn't learning anything and that was my biggest incentive for leaving." While current software workers can rest assured that their lack of market saturation still provides them opportunities to find new jobs, the growth of software development work may pose problems for workers' ability to exercise exit strategies as a response to uninteresting work and skill saturation being potentially intensified by Agile.

Controlling the Shadows: Tightening Surveillance

While the fragmented Agile production process acts as a form of direct control in that it sets the pace and required activities of work, a number of Agile practices aim to heighten visibility over production to ensure that teams are working efficiently and productively. Andrews et al. (2005) suggest that the materiality of the software production process is invisible, taking place "in the minds" of programmers, visible only through "shadows" of production like lines of code on a computer screen. The authors argue that

> the software development labor process in startups represents a new mode of capitalist production, accompanied by new forms of control and relations of

> production. We call this shadow production, because the commodity is only "visible", as it were, through the shadows of artifacts like comments, builds, and documentation (63).

Though the authors make an interesting point about the fact that knowledge-based work is particularly cognitive, the authors miss the fact that, without the instruments of production, namely, a computer screen, programming language, or code, the workers are not able to accomplish their work. A software worker is not able to actually do the work of solving a coding problem without manipulating code, which requires the translation of cognitive activities to a material item, namely, a computer. As Kraft (1977: 99) explains, "the logic, and decision [involved in software development] must take a palpable form...the final goal of all programming must be these concrete (or paper) products of the programmer's activity." In this sense, knowledge-based work like software cannot exist outside of the realm of the material world; software workers are not able to do their work solely in their minds, they engage in an interactive process in which solutions to software problems are developed by not only thinking, but typing out potential code. As Román-González et al. 2017 (678) explain, "computer programming is the fundamental way that enables CT [computational thinking] to come alive." In other words, managerial control over the "shadows" of work ultimately constitutes the most important or meaningful form of control, as the shadows cannot be separated from the work in its most productive form.

Agile offers firms' opportunities to tighten control over the shadows of production, making software production more calculable and predictable. In an Agile Training Video, the speaker informs the audience that Agile is

> really based on transparency, inspection and adaptation. So, transparency means that we need to have visibility on everything that's going on. That's

> why we have the planning meeting which provides visibility for the team what they're going to get done in the Sprint. That's why we have the daily scrum which provides visibility for tasks in terms of what's really happening with the task. And visibilities into impediments. That's why we have to have a final review to actually show and provide visibility with accomplishments, with results, progress.

Overall, Agile draws upon what is called the Empirical Process Control Theory, which mirrors, in many ways, Taylor's scientific management approach and his notion of the "one best way." Taylor writes:

> Among the various methods and implements used in each element of each trade there is always one method and one implement which is quicker and better than any of the rest. And this one best method and best implement can only be discovered or developed through a scientific study and analysis of all the methods and implements in use, together with accurate, minute, motion and time study. This involves the gradual substitution of science for rule of thumb through the mechanic arts.

As stated in the official Scrum Guide, "Scrum is founded on empirical process control theory, or empiricism. Empiricism asserts that knowledge comes from experience *and* making decisions based on what is known. Scrum employs an iterative, incremental approach to optimize predictability and control risk." In order to make decisions on what is known, Agile includes a number of processes to make immaterial and knowledge-based work like software production more visible. This is done primarily through a combination of in-person meetings where workers willingly report progress and the quantification of work tasks. In this chapter I focus on the quantification of work, analyzing in-person meetings as a form of surveillance in Chapter III.

Agile methods encourage the use of "visible information radiators," or physical or digital representations of progress. As explained in an AgileOnline video, visible information radiators like progress charts are used so that "when you walk into the team room or when you come into an area, you should see right there on the walls the status of what they're doing, how they're going… where are they at right now in terms of their tasks and deliverables.” Similar to Taylor’s time-motion studies, one Agile Training video encourages managers to collect data on team productivity: “the easiest way to do that is just go shadow someone, shadow someone and take a timer with you and time how long it takes for them to do something. Then define target measures.” Varsha, an Agile consultant, stated that when she begins working at a firm that is aiming to transition to Agile production, she first asks workers to report their own productivity and keep track of idle time. She stated that she instructs workers to

> list me exactly what you do from morning to evening when you step outside the house--inside the office. You’re looking at your personal email, you are taking a break, you are going for lunch, you are on slack, you’re on messenger…list me out what you do from the time you come and the time you leave.

As I discussed in the sections above, at the beginning of each sprint, workers determine a number of tasks that they are going to complete in the sprint. The tasks are listed digital on what is referred to as the Product Backlog. Following a post-bureaucratic approach in which teams are empowered to make decisions without going to red tape, teams typically decide together which items on the backlog they are most likely to be able to complete within the sprint. Again, completing an item in a sprint typically means that the item is able to be fully delivered to the customer at the end of the sprint period, most often 2 weeks. As explained in an Agile video,

> what happens next is we have points. Points means that every item has a specific size associated with it. Consider it complexity: an 8 is obviously bigger than a five is bigger than a three and these points are called Fibonacci numbers. Think of them as t-shirt sizes, small medium medium large large very large that's pretty much what they are.

As part of an attempt to make complex work more calculable, Agile encourages teams to quantify tasks by assigning them point values. As Jack, a Software Development Engineer explained,

> we use points, where 8 is approximately what one developer can do in a sprint. It feels more accurate. Using points, you know, somehow all the magic math of it calibrates to our scoping patterns and turns out to be pretty accurate.

Points become particularly important as they determine the team's velocity, how many points a team can typically accomplish in a sprint.

Velocity was a key way that visibility and control over the pacing of work is achieved. At the beginning of each sprint workers are assigned a list of tasks. At the end of each sprint, workers track how many points that they were able to complete. The rate of task completion within a sprint is called velocity. Teams are encouraged to improve their velocity with each sprint, essentially increasing the pace of production from week to week. In an Agile video titled "Team Tripled Their Velocity," an Agile Coach tells the audience how collecting data on the pacing of work from sprint to sprint allowed the team to have empirical evidence regarding their productivity. The Coach tells the audience that after the team was required to review their productivity, "they came with a plan to modify how they were working...this team over the course of three sprints quadrupled the number of points they delivered."

In some cases, tracking velocity offered workers the leverage to give realistic and reasonable timelines for the completion of a project, taking back control over the pace of production. When I asked Jack what he liked about tracking velocity, he responded, "I guess just being able to create reasonable expectations for our product owners and managers. So that, you know, so we don't take on too much or anything you know or like to promise deadlines that we can't actually achieve." Still, Stewart warned that tracking velocity may be risky from the point of view of the engineers if management decides to use this to increase the speed of production. He explained, "those are tools to like, attempt to determine how much work is getting done and that's very hard to figure out. And that's like, so there's a side of agile that's trying to, you know, delineate that, but that ends up being used *against* developers instead of *for* them. Instead of helping us get more done it's like, 'well you're not getting enough done' and why is that happening? It's because the scope of this is too large to begin with, probably. Your developers are not lazy, I promise you." Overall, managerial discretion had an impact on whether or not these visibility tools were used to enhance direct control over workers. Yet, as Stewart pointed out, the main function of visibility over velocity is to have some method for empirically tracking work that is done "in the minds" of developers.

Conclusion

While productivity over manual labor can be easily directed, the fact that knowledge work like software production takes place "in the minds" of programmers means that only the "shadows" of production are visible to management, posing a problem for managerial control (Andrews et al. 2005). This study provides empirical evidence that direct control mechanisms that derive from Toyota's production system are a key source for implementing control over software production. I find that through the implementation of JIT, Agile creates a highly fragmented and task interdependent production process, leading firms to tighten controls over what Aneesh (2001) calls the "spaces of play" in software development work. While I do not suggest that Agile leads to a complete closure of worker autonomy or a Bravermanian degradation of skill that early writers warned of in software work (Kraft 1997; Greenbaum 1999), I do argue that

Agile reveals trends in the field of project management *towards* a saturation of skills, precisely because of the nature of JIT or *just enough* production. Moreover, Agile complicates notions that knowledge-intensive work like software development requires novel forms of management control, showing how software work, like manufacturing work, can be subjected to JIT production and scientific management.

CHAPTER III

EMBEDDING COMMITMENT INTO THE LABOR PROCESS

> Precisely because capital has continually to revolutionise production and labour's role within it, it cannot rely wholly on control or coercion. At some level workers' co-operation, creative and productive powers, and consent must be engaged and mobilised.
>
> Paul Thompson (1990: 101)

"The first value is commitment," the speaker of a free Agile Training video explains about team-based work under Agile. "Commitment means that it's going to be hard for you to get anything done if you don't have anyone committed to it. So, the team members commit themselves to the work that they want to get done." The expectation that commitment is necessary to ensure workers get their work done, or, in Marxist terms, exert their labor power, reflects the assumptions of the Human Resources Management (HRM) movement, and especially the High-Commitment Management stream of HRM that gained popularity in the late 20th century. In general, the HRM movement emerged as a response to the crisis of Fordism, which had become associated with the alienation and isolation of the worker, and subsequent conflict between the worker and employer taking the form of shutdowns, slowdowns, walkouts, and strikes (Smith 1997). The HRM movement called for a new relationship between the organization and its members that could smoothen out antagonisms between capital and labor, conceptualizing organizations as "social systems," based on the belief that workers "have a deep need to belong, and to cooperate and communicate with other employees" (Beltrán-Martín 2006: 39). As Beltrán-Martín continues,

> The commitment model marked a movement away from these traditional approaches [of Taylorist methods], by promoting an alteration in the relationship between employer and employee. The main contribution of this [HCM] version of

> HRM is what it aims to achieve: mutuality of interests between the two parties and to increase employee self-direction and commitment. Broadly speaking, it is manifest in a rejection of employee compliance to the organisation's regulations and a greater emphasis on employee initiative and autonomy to manage their own behaviors.[23]

Ultimately, the goal of HCM is to smoothen out structural antagonisms between capital and labor by offering workers the appearance of more control over their work.

As discussed in Chapter 1, in order to adopt the High-Commitment Management (HCM) model, organizations began to explore alternatives to Taylorism and strict bureaucratic structures that would be more amenable to this approach. These new post-bureaucratic organizations aim to elicit commitment from the workforce by replacing the heavy hand of management (forms of direct control) with more self-management and autonomy, or what Freidman (1977) called responsible autonomy. Giving workers more autonomy can help the firm achieve greater flexibility: rather than having to go through the red-tape of the traditional hierarchical organization, organizations embracing HCM hope to improve their agility by empowering workers to make informed decisions that can more quickly respond to unexpected issues, a common occurrence in knowledge-intensive industries. Yet, while post-bureaucratic strategies like HCM allegedly liberate workers from the constraints of management, critical labor scholars have pointed out that managerial oversight in post-bureaucratic organizations, is, in practice, "less marked by a discontinuity in surveillance than by its displacement and intensification" (Briand and Bellmare 2006: 65).

A key strategy for eliciting responsible autonomy and reconfiguring "steep" workplace hierarchies has been the movement of workers into teams. Through the use of

[23] See also Wood and De Menezes 1998.

high-performance work teams, workers are presumably given increased autonomy and control over allocating tasks and job duties as hierarchies between workers and managers are "flattened" (Reed 1996; Storey 2005). Through the introduction of teamwork, firms move from a direct control strategy to responsible autonomy, as control is displaced from managers to workers themselves, allowing management to establish worker accountability mechanisms while obscuring power relations between capital and labor. While responsible autonomy is a common strategy for controlling knowledge-based workers, control strategies are rarely adopted exclusively, and tend to be blended in order to achieve the greatest control (Sturdy et al. 2010: 114; Callaghan and Thompson 2001). By using a framework that recognizes the simultaneous adoptions of various control strategies, namely, direct control and responsible autonomy, we are able to develop a more accurate picture of how responsible autonomy is deployed.

As described in Chapter 2, project-based work (typically done by teams) has been a popular way for firms to organize the labor process, as it draws upon both aspects of direct control and responsible autonomy (Hodgson 2002). Agile aims to improve upon the project management tradition by adopting a lean project approach, with an emphasis on minimizing waste, achieved through direct control strategies like moving to a Just-in-Time (JIT) production process and fragmenting work. As a result, workers experience constraints in their creative autonomy, with the production process leading to a saturation of workers skills. This approach operates as a strategy to achieve Taylored Flexibility, or an increase in predictability and calculability while still maintaining the flexibility to quickly and efficiently respond to changes. In this Chapter, I continue to flesh out the strategies for achieving Taylored Flexibility by focusing on how Agile elicits responsible autonomy, a complementary strategy to the direct control approaches discussed in Chapter 2, and a strategy that is particularly effective alongside direct control. I argue that direct control processes as well as transformations in the production process help to embed commitment *into* the production process, which becomes an especially effective strategy for achieving Taylored Flexibility, or, rendering the labor process more predictable and flexible. By arguing that commitment is embedded into the labor process, I suggest that commitment is not simply initiated through corporate discourses or organizational culture, as is argued in much of the literature on eliciting commitment in the workplace, but becomes embedded into the very fabric of the production process. Moreover, responsible autonomy under Agile is supported by the use of other direct control mechanisms, as described in Chapter 1, creating an effective hybrid control regime.

Part of the strategy of Taylored Flexibility, Agile employs three primary mechanisms to embed commitment into the labor process: task interdependence, mutual accountability, and co-location. Task interdependence necessitates cooperation between workers on teams as individuals rely upon one another to meet team-based goals and to complete their individual tasks. Teams are also encouraged to commit to one another through the establishment of mutual accountability, or the practice of reviewing a worker’s performance based on their performance as a *team member*, rather than just an individual. In doing so, team members work cooperatively as a result of external pressure and rewards. Finally, commitment and cooperation are achieved through a spatial reorganization of the workplace through what is called co-location. Co-location refers to the spatial organization of an office typical of post-bureaucratic workplaces, in which high-walls and individual cubicles are replaced by glass doors and shared workspaces. By encouraging workers to be physically present at work, Agile disrupts much of the trends towards remote work in knowledge-intensive occupations.

Through spatially coordinating the workforce, as I describe in more detail below, firms are able to elicit behavioral empathy (Akgun et al. 2015) or a willingness to help teammates solve their coding problems, or even take on their teammates' work to meet team-based goals. Moreover, mechanisms for embedding commitment also allow firms to improve surveillance over the labor process, as workers willingly and routinely participate in progress updates. Ultimately, these mechanisms for establishing commitment allow the firm more flexibility to respond to changes and challenges in the production process, while also ensuring that the production process does not become unpredictable or out of the control of management. Moreover, responsible autonomy offers a mechanism through which workers’ consent to the labor process is gained. In the sections that follow, I offer a framework for understanding commitment as a form of control, discuss the role of commitment and the rise of teamwork within the manufacturing industry, and discuss contemporary mechanisms of responsible autonomy.

Commitment as Control

To remind the reader, in the capitalist context, employers seek control or the “exercise of authority over workers” in order to realize the most surplus or profits possible from workers’ labor, identified by Marx as “variable capital.” Labor power is considered variable because it is ultimately up to the worker to decide how much effort they will expend such that their labor is profitable. As Braverman (1974: 37) writes,

> Labor, like all life processes and bodily functions, is an inalienable property of the human individual. Muscle and brain cannot be separated from persons possessing them...what the worker sells, and what the capitalist buys, is not an agreed amount of labor, but the power to labor over an agreed period of time.

In order for the capitalist to make the most of the variable capital they purchase from the worker, they must engage in a process of management over the worker so that they exert themselves in the most profitable manner, which is a process of control. To establish managerial control, Friedman argues that managers have relied primarily on two categories of strategies: direct control and responsible autonomy. Traditional (Taylorist) forms of control are typically referred to as direct control, and can be defined as a strategy that "tries to limit the scope for labour power to vary [or how productive workers are being] by coercive threats, close supervision and minimizing individual worker responsibility" (Friedman 1977:78). While direct control is considered the dominant strategy used to organize manufacturing labor, responsible autonomy is another control strategy available to employers. Responsible autonomy refers to managerial attempts "to harness the adaptability of labour power" by "giving workers leeway...status, authority and responsibility." Responsible autonomy has become a particularly ubiquitous strategy in white-collar workplaces. As Friedman (1977: 79) notes, both strategies of direct control and responsible autonomy "have characterised management throughout the history of capitalism, but generally the Responsible Autonomy type of strategy has been applied most consistently to privileged workers and the Direct Control type of strategy to the rest." Indeed, literature exploring control mechanisms in the software industry have emphasized the role of responsibility, illusions of autonomy, and commitment as forms of control (Baldry, Scholarios, and Hyman 2005; Barrett 2005; Damarin 2013; Hodgson and Briand 2013). This marks a key intervention of Chapter 2, which analyzed the *direct* control strategies being used to organize software work through Agile project management and their roots in lean production, which continues to be an undertheorized aspect of control over knowledge-intensive work. Nevertheless, Agile's reliance on direct control strategies have not been adopted *in lieu* of responsible autonomy, but alongside it, to develop a complementary hybrid control regime of Taylored Flexibility as I continue to analyze in this chapter.

Establishing Responsible Autonomy

Establishing responsible autonomy and commitment at work, Frenkel et al. (1995) argue, can be developed through four mechanisms: providing intrinsically satisfying work, the use of semi-autonomous teams, embedding norms into bureaucratic processes (e.g., merit raises), and a strong corporate culture. Corporate culture can be described as "instilling strong, common and organization-focused norms and values among employees in order to create a distinct, shared sense of identity and belonging" (Sturdy et al. 2010: 117). For example, Kunda (1992) analyzed a high-technology corporation to show how managers impose responsible autonomy through what he refers to as normative control. As part of this normative control strategy, management disseminates an organizational ideology that espouses self-discipline, hard work, moral and ethical conduct, teamwork, creativity and autonomy in order to "elicit and direct the required efforts of members" (11). The development of a strong corporate culture, Kunda argues, replaces micromanagement, rigid hierarchies, or other traditional direct control strategies.

Of course, as described in Chapter 1, management does not have to make exclusive choices regarding the adoption of either direct control or responsible autonomy, and, in fact, typically adopt constellations of control strategies. Project management has been a good example of this, employing elements of both direct controls and responsible autonomy, particularly through the use of project-based teams (Thomas 2006). Still, I argue that scholars writing about the role of commitment, loyalty, or the incorporation of the workers' subjectivity in the new economy tend to overemphasize responsible autonomy as a particularly new phenomenon. As I show in the next section, though the managerial emphasis on commitment gained traction in white-collar workplaces in the late 20th century, discussions of how to control the disposition of the worker have long been part of the development of managerial control since the rise of industrial capitalism. Thus, we must understand attempts at achieving control through responsible autonomy as part of the revolutions in the production processes, and identify the adoption of old models alongside the introduction of potentially new ones.

Manufacturing Commitment

A review of Taylor's *Principles of Scientific Management* reveals an early concern about mitigating antagonisms between capital and labor by developing a more cooperative relationship between worker and employer. Taylor (1911: 10) writes,

> throughout the industrial world, a large part of the organization of employers as well as employees, is for war rather than for peace...the majority of these men believe that the fundamental interests of the employees and employers are necessarily antagonistic. Scientific management, on the contrary, has for its very foundation the firm conviction that the true interests of the two are one and the same.

Taylor hoped that by paying workers through a differential piece-rate system, a wage system that financially incentivized workers to be more productive, he could show workers that they had a shared interest in enlarging the firm's profits. However, the toil of deskilled industrial labor proved a significant barrier to eliciting commitment and loyalty from the worker. This became especially untenable for Ford who, adopting Taylor's scientific management methodology, experienced an exceptionally high turnover rate of aggrieved employees:

> In 1913, Ford hired more than 52,000 men to keep a workforce of only 14,000. New workers required a costly break-in period...some men simply walked away from the line to quit and look for a job elsewhere. Then the line stopped, and production of cars halted. The increased cost and delayed production kept Ford from selling his cars at the low price he wanted (Downs 2013: 367).

As a result of the high turnover, Ford instituted the now famous 5-dollar-day, more than doubling his workers' wages. Yet, in raising the wage of the worker, Ford also engaged in a project of cultivating the disposition of the worker, going beyond simply subjecting the worker to a regime of scientific management. The $5 raise was not an increase automatically given to all employees, but was the result of a bonus for employees who qualified, based on how well they submitted to Ford's demands to create a "new aesthetics and psychology" of the worker, "a new type of worker and a new type of man" (Harvey 1990). As Worstall (2012) writes

> The $5-a-day rate was about half pay and half bonus. The bonus came with character requirements and was enforced by the Socialization Organization. This was a committee that would visit the employees' homes to ensure that they were doing things the "American way." They were supposed to avoid social ills such as gambling and drinking. They were to learn English, and many (primarily the recent immigrants) had to attend classes to become "Americanized." Women were not eligible for the bonus unless they were single and supporting the [paternal] family. Also, men were not eligible if their wives worked outside the home.

This attempt to cultivate the worker's disposition reflects a long-held tradition in management activities over labor, though techniques for cultivating workers' subjectivities have grown more central to management strategies. Today, a key strategy for ensuring the kind of worker commitment, loyalty, and responsibility desired by Taylor and Ford initially emerged at Toyota, with the introduction of teams. Through

Tachaii Ohno's reconfiguration of the assembly line to now include clusters of teams, Toyota would become a model for introducing responsible autonomy to the shop floor.

Team Commitment

Teams have been a particularly popular strategy for eliciting commitment from workers. The movement of workers from the assembly line to the team has roots in Toyota's Total Quality Management approach. Teams represent a new strategy to achieve what Taylor desired: a mitigation in conflict or antagonism between workers and managers. While the Toyota Production System (TPS) primarily emphasized minimizing waste, as discussed in Chapter I, "respect for people," was also a guiding principle, initiated by Taiichi Ohno to quell worker resistance that was beginning to arise in Japan's automobile factories and around the world. Like Taylor and Ford, Ohno sought management practices that could mitigate the slog of manufacturing work, adopting management practices to allegedly improve the workers' position within the firm.[24]

Unlike Ford's degradation of workers' skills through an interchangeable production process, Toyotist production practices grouped semi-skilled workers into self-organized and self-disciplined teams who engaged in face-to-face communication and collaboration with their teammates (Hutchins 1988; Liker & Morgan 2006). Ohno, in many ways, took a strong corporate culture approach to cultivating commitment, encouraging managers to empower workers to make mistakes and offer suggestions in order to enhance their commitment to the firm. Ohno (1982: 6) writes, "wouldn't workers be even more cooperative if when mistakes are made, they are not met with reproving

[24] In his 1982 management text, *Genba keiei* (*Workplace Management)*, Ohno includes a number of chapters that highlight new management practices such as, "Offices without supervisors," "Be a boss people can count on," and "Work is a contest of wits with subordinates."

looks, but with encouragement?" Throughout *Workplace Management* Ohno encourages management to develop relations with their subordinates that are not rooted in traditional top-down management styles, but relations where managers can cultivate productive, efficient, and committed workers. He explains, "I think--to exaggerate slightly--that what you have to work on is sort of character-building. You have to be concerned that the other people are sticking with you." Instead of pursuing a character-building project through the intrusive measures taken by Ford, Ohno reorganized the relationship between employers and workers in the firm. He achieved this by offering workers tenure and fair compensation packages, and also aimed to offer workers more interesting work, the ability to try out new ideas, the ability to stop production in order to resolve defects, and a sense of loyalty to the workers.

Yet, as we have seen from critical labor scholars, team-based work has not been a strategy for truly rejecting the structural antagonism between labor and capital, but another way to establish control. The team-based forms of control initiated at Toyota have become particularly insidious to post-bureaucratic or "participative" organizations (Barker 1993), as they establish forms of "soft despotism" (Clegg and Courpassen 2004), that can increase worker commitment. Teamwork allows management to establish accountability mechanisms that reconfigure the locus of control from supervisors to teams, obscuring power relations between capital and labor as team members surveil, scrutinize, and ultimately manage one another (Ellway 2013; Barker 1993; Sewell 1998; Kunda, 1992). In other words, "Responsible Autonomy does not remove alienation and exploitation, it simply softens their operation or draws workers' attention away from them" (As Friedman 1977:101). In this sense, responsible autonomy does not altogether replace more despotic or direct control regimes, but is a useful complement to direct control in that it can obscure conflict by generating consent to the labor process.

I argue that the strategy of Taylored Flexibility allows firms to successfully achieve combinations of direct control and responsible autonomy in order to both "capture the benefits particular to variable capital," using the responsible autonomy strategy, while also using direct control to try to "limit its particularly harmful effects," or worker discretion and resistance. Ultimately, the hybrid control strategy of Agile draws upon elements of Taylorism, Toyotosim, and High-Commitment Management in order to establish a strong control regime over the variable capital embodied in the worker. Moreover, I argue that because of the ways that commitment is embedded into the labor process through Agile, workers are less likely to view management as hypocritical, despite deploying both direct controls and responsible autonomy. Instead, commitment is normalized as part of the production process, and is not necessarily incongruent with other direct control strategies.

Cultivating Commitment under Agile

Agile teams are typically made up of less than 15 people, though large Agile teams can have 50 or more team members. Team roles usually include a team lead, a product owner, representatives from sales and marketing, and software developers of varying senior and junior statuses. Workers in this study did not report particular commitments to their companies, and primarily conceptualized commitment as team based. This finding is aligned with more recent studies analyzing the cultivation of software workers' commitment, which emphasize employees' commitment to their identities as successful, creative, knowledge workers and the products they develop, rather than commitment to a particular firm. For example, Barrett's (2004) analysis of managers and employees in three software firms shows that workers are more deeply committed to their work, rather than a firm *per se*. Like Barrett, Marks and Lockyer (2005) found that software workers have higher professional identification than organizational identification. This can actually contribute to employee turnover, as workers leave "boring" jobs for more challenging work.

Lack of company commitment is reflected in the high turnover rate in tech: a 2018 LinkedIn report showed that the tech industry had the highest rate of turnover compared to all business industries in the U.S., at 13.2%. As we have seen, turnover has remained a concern for employers, and eliciting commitment has become a strategy for compelling workers to remain at a particular firm. Yet, participants in this study recounted experiences where their employers did not seem particularly concerned about turnover, and in fact, embraced it. Steven, a white data architect, stated that during an onboarding meeting, his general manager:

> looked at me and four other people that were in my hiring cohort and said "if half of you are still here in one year, I'll be shocked. And it's okay. If I can help you get a dream job elsewhere that's fine, I'm going to ride your benefit as long as I can."

Many participants echoed Steven's experience with firms embracing turnover in tech. Max, a white quality assurance scientist, described how his discontent with his job, and a lack of a clear job trajectory, led him to arrange a meeting with his HR department. He said he told HR of the high turnover:

> Don't you think from an HR perspective that there's a problem when there's such a high turnover rate? That's an indication of an issue, like why are people not happy doing QA? Is it the pay? The management is messed up? Because of resources? Like why is it that? And you know HR was kind of like, "oh, yeah I understand, whatever" and it kind of came down to being accepted that QA is this

> high turnover rate thing and they’re not going to change anything because they’re okay with that.

Overall, the movement of workers from firm to firm, and even intraorganizational movement from team to team was common to participants’ experience, which undoubtedly, I suggest, makes the establishment of a strong corporate culture or company commitment challenging. Thus, I argue that the widespread adoption of Agile especially by technology firms may operate as a strategy to standardize and habituate workers to a particular software labor process in order to achieve Taylored Flexibility. In other words, by adopting a standardized production process, Agile, firms may be able to mitigate the costly effects of turnover, and rest assured that workers entering the workplace are already highly committed to their teams and to their work and understand how the production process is organized, requiring less effort on the part of the firm to establish commitment or train workers. While workers may only work at the organization for a short while, the Agile labor process is one in which commitment is highly necessary, because of the ways it is embedded in the production process, as I analyze below.

Conceptualizing Commitment

Across workplaces, commitment was a commonly discussed aspect of work. As discussed, commitment can come in the form of company commitment (now less common), commitment to the work, and commitment to the team. In the Agile context, cultivating team-based commitments are of the highest priority, as workers are expected to commit to each other to deliver value to the customer and get their work done. As explained in an Agile training video:

> Agile teams are empowered, empowered to make decisions, empowered to self-organize their work, and empowered to decide how they want to execute on the vision as long as they're within the boundaries. How do they self-organize around

> a task to really accomplish their goal...Agile teams are also highly motivated which means that they're inspired, they have the vision, they have a purpose, and they are really trying to get something done. They're not the, "let me sit back here and ask my boss what does he want me to do next?" it's more of "let's get this done, we committed to this iteration, we really promised the product owner [on the team] that we're going to get these stories done." So, what can we do together to really accomplish that goal?[25]

This excerpt reflects the responsible autonomy control type through a High-Commitment Management strategy, in which workers are given more control over how to allocate tasks and organize their work, such that heavy management intervention is not needed to ensure workers get their work done.[26]

Agile makes explicit the firm's relation to commitment, reflecting the notion that firms do not necessarily expect reciprocal loyalty between workers and employers. While workers are expected to be committed, the firm, especially upper-management and executives, are simply "interested." As a speaker in an Agile training video explains,

> We refer to the Team as committed individuals. Why do we call them committed? Because their neck's on the line. They have to get this project done. If the project falls apart, they are accountable for failure or success of the project. So, we call them our committed team.

The speaker goes on further to illustrate the difference between committed individuals (the team) and interested individuals (upper management and executives) by explaining what they call the well-known chicken and pig joke. The speaker describes,

[25] To clarify, the product owner under Agile is not the client, nor do they actually "own" anything. The product owner is member of the team who is not engaged in design or coding but is primarily responsible for ensuring work gets done: "The Product Owner (PO) is a member of the Agile Team responsible for defining Stories and prioritizing the Team Backlog to streamline the execution of program priorities while maintaining the conceptual and technical integrity of the Features or components for the team." https://www.scaledagileframework.com/product-owner/

[26] Of course, as shown in the previous Chapter, this adoption of HCM comes alongside more direct control strategies, creating a regime in which workers simultaneously experience more responsibility over their work, while also conducting their work in a more tightly controlled production process through the use of JIT work and the quantification of their tasks.

> You might have already heard about the famous joke within Scrum [a type of Agile], it's the chicken and the pig joke. So, the way that it goes is the chicken and the pig got together and they wanted to start a restaurant, a bed-and-breakfast, and they were wondering ‘what should we call it?’ And so, the chicken called, said ‘let's call it Ham and Eggs!’ And then the pig said ‘well, not sure I like this idea because I'm going to be committed [gestures cutting her own throat] and you're just going to be involved.’ So that’s sort of the background behind why we call people involved and why we call them committed [smiling].

The chicken and pig joke not only illustrates Agile’s own conception of committed versus interested individuals, but also, I argue, operates as a useful metaphor of the differential power relation between the employer and worker. While management literature may espouse commitment, autonomy, and worker discretion as evidence of a new relationship between labor and capital, the chicken and pig fable shows how neoliberal configurations of work continue to shift risk from the employer to the worker (Kalleberg 2009), making workers even more responsible for the success of the firm. Because of workers’ dependence on wage labor for their livelihood, they have little choice in rejecting the role of the pig.

I found that making workers more responsible at work is especially effective when commitment is embedded into the labor process. As I discuss in the sections that follow, Agile develops three primary mechanisms for embedding commitment into the software labor process: task interdependence, mutual accountability, and spatial coordination, in order to achieve Taylored Flexibility. The strategy of Taylored Flexibility is particularly effective because of its ability to combine complementary control regimes that rely on direct Taylorist control strategies as well as responsible autonomy, especially through teamwork. Moreover, eliciting responsible autonomy by embedding commitment into the production is especially effective, as commitment is no longer just a discursive or rhetorical tool by management, but a necessity of the production process.

Task Interdependence

Task interdependence is a key way that commitment becomes necessary, rather than a desired, aspect of the production process. As described in Chapter II, Agile's fragmented JIT process allows firms to achieve greater control over software work, in that they can be flexible to shifting demands while having more predictability over time, costs, and the work that teams are performing. The result of this fragmentation process is highly task interdependent work. Task interdependence refers to the presence of interrelated tasks necessary to complete a project (Kakar 2016) and the extent to which team members expect that they will personally benefit by contributing to team goals (Van der Vegt et al.,1998). Eliciting a cooperative disposition from workers, I suggest, becomes both more desirable and more achievable when interdependence is embedded into the labor process. Carrie, an Engineering Lead, interestingly pointed out how the interdependence required of software teams stands in stark contradiction to stereotypes of software workers as especially isolated or individualistic. She explained,

> Some of the cultural stereotypes are that there's the one genius coder that just bangs out a shit ton of work, but the truth is that sustainable, maintainable, good code is always written by teams...It happens in teams and it's collaborative and highly systematized. You can't make good software without doing the work on a team and that's just the bottom line. I don't know why the conception is so wrong. Probably Hollywood. It's more romantic to think about the one hero coming in and saving everything.

Because work in a sprint is so highly fragmented, teammates are intimately dependent upon one another to complete tasks in a timely fashion so that everyone can meet the final sprint deadline. Thus, the particular combination of highly interdependent work, the result of JIT, and organizational expectations for mutual accountability as described in the next section, help to construct a powerful control regime that improves firm flexibility and calculability, or Taylored Flexibility. If successful at establishing team member commitments, firms can have greater assurance over the likelihood that work will get completed in an efficient and productive manner.

Firms are especially able to maintain flexibility as self-organized teams are given the authority to adjust task assignments and priorities among themselves, in order to best meet the goals of the sprint, the strategy of responsible autonomy. For example, Jack, a Software Development Engineer, explained that at the beginning of a sprint, team members assign one another the tasks they are going to complete over the sprint cycle. Yet, assignments are not hard and fast, and team members must often help each other out in order to succeed in the project goals. In other words, through responsible autonomy, team members take on the responsibility of allocating tasks and addressing challenges without heavy oversight from management, helping to produce a more efficient production process that can keep the firm from wasting labor time. However, as Jack describes, helping teammates complete their tasks, or “unblocking” teammates when they are experiencing challenges, cannot simply be explained as a result of responsible autonomy, but is a *necessity* that emerges from the interdependence of work that is established by Agile’s JIT process. Jack told me:

> A lot of our work is dependent on each other, especially now. Scrum [a type of Agile] has increased that. If we’re all kind of siloed in our own project maybe one person being blocked on their projects doesn’t affect us as much. But [in Agile] if we're going to be taking up a task, it’s kind of, you know, it's related to our teammates’ task or dependent on it even. Then we're going to, then unblocking them unblocks us. I think there’s a lot more shared ownership there.

Through this example, we can see how shared ownership and commitment is embedded into the Agile software labor process. In other words, Jack is not simply helping “unblock” or help solve the problems that his coworkers are facing because he values commitment, because of company culture, or because he has the autonomy to do so, but because he would not be able to complete his tasks otherwise. In this sense, we can see that responsible autonomy may be particularly effective when combined with more direct control strategies that reconfigure the production process in such a way that commitment to helping teammates accomplish tasks becomes a necessary component of completing one’s work.

Mutual Accountability

While task interdependence is a powerful mechanism for embedding commitment into the labor process, workers in the study also expressed commitment to completing a project and helping their teammates out as a result of the Agile concept and practices of what is referred to as mutual accountability. Workers in Agile are not simply committed to completing work on an individual-level, but are mutually accountable to one another for completing the work of the project or sprint. Though individuals on teams assign themselves and one another tasks, ultimately, the sprint is only successful if all tasks have been completed *as a team*. This means that while one member of the team may be working efficiently, if all members do not complete tasks, the team ultimately fails. As one training video put it,

> Another characteristic of an agile team is that they are mutually accountable. There's no finger-pointing here. There's no “I didn't get the requirements clearly and that's why I didn’t code it right way” or “the developer didn’t code this correctly so that's why I found all these bugs.” Really, we are all in it together and we're all in it to get the story done and deliver value for the customer. So that's what Mutual accountable means.

By being mutually accountable to their teammates and committing to a particular amount of work to be accomplished in the sprint, firms are able to ensure that work gets completed at an appropriate pace. As we have seen in the previous chapter, the firm attempts to control the pace of production by recognizing the production process into two-week sprints. In combination with this reorganization of production that emphasizes these temporally short-term increments for completing work, the mutual accountability that workers adhere to in order to complete sprint goals also helps to structure an efficient pace of production. Workers are given the autonomy to determine how and when work gets done, as long as they are able to make the sprint deadlines. As Casey put it, “It’s not about how many hours your butt is in the seat, it’s that we’re able to get our commitments to each other done. Within the team.” Denise, a Director of Engineering, provided a critical reflection on the autonomy that workers have over the hours they put into work, and how this allows the firm to increase the rate of exploitation, especially as most workers are paid through a salary system. She explained that because software workers are not awarded overtime pay, they end up working long hours with compensation structures that are not necessarily as generous as they appear. She explained:

> It’s a dirty little secret in the software industry. Or I should say a thing that engineers don’t like to think about. Every software company and software engineers are exempt. They never get paid overtime. But if they’re in a startup or at any place where many hours are expected for a week, and in the software engineer they have a reputation of very high salaries, if they divide their high salary by how many hours, they work it doesn't come out that good. Then if they multiplied that by 40 hours a week they’d be like, “whoa that’s not so great, that doesn’t make my salary so great at all.” They don’t want to look at that. That really bugs them. So, it’s a mystique that’s built in the industry, at least in the U.S. Some of it has to do with capitalism, too. So, in other countries that are more socialist maybe, maybe it’s not like that. But it certainly is in the capitalistic, a country like the U.S.

Denise’s concern reflects what scholars have noted: in her study of contingent Microsoft workers, Rodino-Colocino (2007) found that knowledge workers are some of the most overworked employees, expected to work close to eighty hours without paid overtime. Through the expectation to work long hours, a JIT production that fragments work into two-week time increments, and an expectation of mutual accountability, software firms are able to ensure an efficient pace of production over work that cannot be paced in the same ways as manufacturing labor. In this sense, though knowledge-based work may pose some issues for implementing control over the pace of production that cannot be solved through the machine, Agile has provided strategies overcoming these barriers to control.

In one case, however, a sense of responsibility to the team lead Carrie, an Engineering Lead, to contest working arrangements determined by upper management. When I asked Carrie if she felt a responsibility to her teammates, she explained that she does in the sense that she wants to “protect them from the bullshit of the company.” When I asked her to explain what that meant, she stated,

> like the producer that I yelled at the other day [laughs]. You know, she did a very bad job of managing the project and she, she put undue pressure on the developers to deliver something that’s actually impossible. So, when I see things like that I step in, unless the developers ask me not to, and I step in and push back for them. I feel like a lot of times, especially for junior devs, they don’t know that you can say no, or they don’t trust their own instincts or get caught in things or political situations you can’t understand. So, I try and, I mean I’ve been doing this for so long. People make the same fucking mistakes over and over again so when I see it happen I kind of step in and I’m like “yeah, you don’t have to do what she says. Let’s take it back. Let’s fix this.” Cuz that’s bullshit, that kind of thing. Sometimes it works and sometimes I just make things worse with my strong personality.

Carrie’s discussion of protecting workers shows the ways in which commitment under Agile does transform from commitment to the organization to commitment to the team, which, in this instance, led to resistance to control. However, are Carrie mentioned, she was not always able to be successful with her attempts to push back, and noted that, at times, software developers were too afraid to "rock the boat" and remained silent about concerns.

Another way in which Agile organizations help to elicit mutual accountability is through performance reviews. While performance reviews have been theorized as a bureaucratic mechanism for eliciting organizational commitment and consent to the labor process (Frenkel et al., 1995), I argue that the particular expectations that workers' performance is assessed based on their collective performance shows a shift in the site of commitment from the level of the firm to the level of the team. As Steven explained, "in bonus measures and especially in conversations about raises they look at the whole person. That's the language they use: 'who is this person as a teammate? 'Do they feel good [to others] in the office?' and so forth." Other workers described a process of 360-reviews, in which teammates anonymously review one another. For example, Holly, a Project Manager, explained that in addition to individual reviews from managers, the company also performs "what they call 360 reviews. Where they send out a survey to 4-6 people and say, "tell me about Holly behind her back." Holly explained that she is currently compiling direct reports of the members on her team to upper management, explaining the process as follows:

> So, I send out a survey about a person and I'll take that feedback and I'll synthesize it and anonymize it, so it's not immediately identifiable from coming from one person, and I use that to give them their reviews. So, if everyone on the team says, "this person is super abrasive, I can't work with them and they don't get their work done," then I might, I might not [already] know that because all I see is, well, tickets are moving across the board [i.e., tasks are getting completed]. So, that might color my review of the person to say you know, on the surface it

> seems like they’re doing great work, but it doesn't seem like they're doing well with the team and you know things like that.

One worker, who felt that engaging in interpersonal relationships with team members was particularly challenging, expressed concern over the metrics used to evaluate her work. Erin, a Full Stack Developer, told me that her concern with reviews is that “it’s not objective, as far as I know, in the sense that they go through and analyze my code, that there’s this many bugs. They don’t do that. So, I think that it is at least somewhat subjective.”

In many ways, performance reviews challenge the assertion made by the HCM movement, in which the aim is “eliciting a strong commitment to the organisation, so that behavior is primarily self-regulated rather than controlled by sanctions and pressures external to the individual” (1995: 220). Tyler, a programmer, explicitly expressed a combination of extrinsic and intrinsic motivators for committing to completing work on the team, and helping the team meet their project goals. He stated that junior-level programmers may ask him, “how do we get around this [problem],’” and, you know, I just tell them, ‘oh yeah, I’ll get this done right now.’” When I asked Tyler to explain why he was willing to help junior teammates, he laughed and responded, “I value knowledge.” He then elaborated, “It looks good on me...If I increase the value of the company then I’m also increasing the value of how they interpret me.” As team-based commitments become a component of work performance, commitment *does* operate as an external pressure on the individual, having a direct link to the individual’s compensation and ability to succeed on the team.

Spatial Coordination

Spatial coordination has been an element of managerial control since the transition from craftwork to industrial capitalism. In order to establish control over workers, firms erected factories, grouping disparate craftworkers into one spatially coordinated setting. Though the rise of factories are typically associated with the introduction of mechanized labor, factories were also a primary source of discipline. As Friedman (1977: 87) explains:

> Both discipline and mechanization were the reasons for setting up factories. But to enforce discipline was the main reason at first. The rise of the factory system predated both the introduction of power to the labour process and any technical change in machine design or organisation of work for which factories were particularly suitable. In wool spinning, for example, the 'spinning jenny' was used without power in both factories and cottages...factory discipline meant that labourers worked more regular hours and longer hours in all.

While control over the labor force through discipline over the body (e.g., grouping workers under one roof) has been central to traditional managerial strategies, the scholarly focus prioritizing analysis of responsible autonomy in the knowledge-based workplace is a result of, I argue, a conceptualization of knowledge workers as disembodied, thus not requiring the spatial coordination of labor used in manufacturing work.

The view of knowledge workers as disembodied is immediately evident in the Bureau of Labor Statistics description of software developers as the "creative *minds* behind computer programs." Drawing on an academic example, Andrews et al. (2005) analysis of the software labor process led them to call this production process "virtual." This virtual labor process of software work includes four stages of production, which are neither linear nor mutually exclusive, much like the Agile production process. The authors emphasize the creative, iterative, and dynamic process of software production that often requires teams of workers creating, revising, and reconstructing code. They show that management faces limitations in being able to dictate this process, because the work largely takes place "in the minds" of workers. The labor processes exemplified through these stages, the authors argue, largely distinguish "programming as knowledge work in contrast to the physical work of manufacturing," and that "coding is a mental rather than a manual process which depends on developers' active creativity and decision making" (59). Though the authors are able to offer an important account of software production, they do so under a frame that dichotomizes mental and manual labor, erasing the materiality of workers' experiences.

Ursula Huws (2003: 139) provides a critique of this assumption, writing that software work "involves the body in a series of physical activities, such as pounding a keyboard, that have implications for its physiological well-being." Following Huws, I argue that any analysis of the immaterial labor process of knowledge workers should not privilege the intellectual over the bodily components of work, discipline, and control. In order to fully develop a labor process theory of software development, scholars should turn their attention to recapturing the material and bodily experiences of knowledge workers. In doing so, we are able to see how disciplining the body in order to improve control over productivity returns as a managerial goal, even in a knowledge-based context.

In recognizing the role of discipline over the body in tech work, I found that while task interdependence and mutual accountability certainly facilitate an easier expectation of commitment by workers, Agile also reorganizes the labor process physically and spatially through the use of co-located workspaces. Though software work is highly amenable to remote work, Agile encourages face-to-face interaction and the close physical approximation of workers and managers to one another, in order to "reduce the cost of moving information" (Cockburn and Highsmith 2001: 301). This is done through an emphasis on co-location, or grouping teams together in a physical location in an office, typically an open-floor plan office.

In general, team members in this study held negative perceptions of open-floor office design, viewing them as loud and not conducive to productivity, established as a trendy and cost-saving option for firms. For example, when I asked Denise, a Director of Engineering, if she could change anything about her workplace what would it be, and she responded:

> Tables and chairs are so important. As a consultant [in tech] I went from place to place and we often got that lousy physical environment. It makes such a difference in people's productivity. They pay so much for software engineers and they give them the lousiest back aghast places to sit. Open office plans are the thing now, and they are so unproductive. There are three major studies that prove that the interruptions that you get in an open office plan break up the software engineer's mental you know, focus, through thinking, and they have to reset their context. But, like, it's the "hip" thing and they do it over and over again. It saves money because they fit more people in, so it saves money in the office layout. But they are losing money in productivity. It's terrible. Open offices are a terrible mistake. In fact, I was going to a chiropractor, I had gone to them off and on, but just before I left my job, just recently, I was going to the chiropractor every other week for about a year. And I told him I was quitting my job and he said, "good for you!" He said, "I wish that my other tech workers would do that. I bet you won't be coming in as often." He's right, I haven't been back to see him since. So that tells you something there. You should interview chiropractors![27]

[27] Interestingly, in a book edited by Ben Tarnoff and Moira Weigel (2020: 134-135), the authors do interview a massage therapist assigned at a high-tech company. When asked to "tell us about the bodies," the massage therapist responded, "people worked long hours. They had postural problems, the sort you get from sitting at a desk all day. Many of them did not exercise or strengthen their body, so the wear and tear of being on a computer for ten to twelve hours straight was even worse because they had muscle weakness...mostly, though, people were just stressed out. So they had the muscle tension that comes with that."

Carrie, an Engineering Lead, also expressed disdain for open floor plans: "I hate them! I feel a rage about the open office plan." When I ask her why she thinks they're so ubiquitous, she responded,

> Because they look pretty. I think we're really service oriented and people who do marketing and run businesses want shit to look nice and really focus on the surface of things and if you're involved in coding or technology the surface is not what's most interesting about what's going on, so, I don't know. I asked for a cubicle and they were like, "nope!" Cuz it would wreck the floor plan, you know? Whatever. We're organized by team. We're close, close, like right up on top of each other.

Holly, too, found open office plans to be "loud and distracting," stating "it doesn't work great for me at all." When I asked why she thought open floor plans are so common, she responded:

> It's cheap. It's really cheap. You know, I would fit three people, maybe four, on each side [of the table] so at least 6 people sitting in a small space. When I first came into tech and was in a cubicle, it would have been this entire corner, four desks each, one facing a corner, for four people, you know? And so, it's just so much more cost effective to throw people in a room and say that it works even if it doesn't.

Like others, Linda, a Scrum Master, recognized the trendiness and cost-savings factors of office floor plans to be the primary reason they are used, but also identified socio-technical benefits. When I asked Linda what she thought of co-location and open office plans, she responded

> I think that it's very fashionable and I think it's less expensive to do it. I think it does extend some measure of additional collaboration to the teams because they're right there. But it's also very noisy and very disruptive. We use Slack [an online communication platform] a lot to communicate, and we meet face-to-face fair enough. And they pair programs [simultaneously code together] quite a bit and that's very helpful. They just get a room and they just swarm on something.

Though co-located workplaces provide clear benefits for firms in terms of cost and improving surveillance, co-location was also used to elicit behavioral empathy, which I suggest should be conceptualized as part of the responsible autonomy strategy.

Behavioral empathy is defined as “the ability of team members to respond to the feelings of others within the team...engagement in helping behavior” (Akgun et al. 2015: 249). The project team literature broadly and software development team literature specifically emphasizes the impacts of interpersonal and collective empathy on the rationalization of software development work, leading to better project performance (especially in terms of team learning, time-to-market, and cost) (Rues and Liu 2004; Akgu¨n et al. 2015). Though participants did resent the distractions that can come with an open office plan, they also noted how co-location did help to cultivate strong team relationships. As Agile proponents like Cockburn and Highsmith (2001: 301) suggest, talking in-person to a coworker that sits in the same space as you is not only faster than typing out an e-mail or direct message, but it also helps to “improve the team’s amiability so that people are more inclined to relay valuable information quickly.” Through co-location and interdependent work, I found that firms were able to be especially successful at eliciting behavioral empathy, which, I found, improves firm flexibility and surveillance.

Sam, an Agile Consultant, explained how co-location could positively impact communications streams between teammates:

> As far as sitting together I personally am a huge proponent of that. The hallway conversations that you have, the differences between what you say in an email and the ability for that to be misinterpreted, you cannot collaborate in a one directional communication medium like email, and video is limited in intentional time value. Whereas sitting next to you has an unlimited amount of time value.

By saying that sitting next to someone has unlimited time value, Sam means that workers are able to call upon one another to help finish tasks in a more streamlined and flexible manner than through trying to communicate through digital communication platforms. Sam continues her positive view of co-location by saying that the spatial coordination of workers can help to elicit behavioral empathy, or a willingness to help one another complete work, which can bring benefits to the production process and the firm:

> If I know that your child is sick because you told me on the way to lunch, and you can't get your work done, it's less likely I will automatically resent you for having to do your work. 'Cuz I now I have to do it 'cuz you didn't finish it and we're on the same team.

Agile's emphasis on cooperation and behavioral empathy operates as a control strategy to allow the firm to be more flexible and leaner, as they do not have to provide rigid processes for assigning extra work, or compelling workers to help one another meet team goals.

In fact, an emphasis on team relationships was seen as better for productivity over all. For example, Jennifer, an Engineering Lead, explained that team meetings begin by "going around the table saying how was your weekend? What did you do? Or show each other pictures. And paradoxically we get through our agendas better if we take five minutes to be humans together!" When I asked Jennifer to reflect upon why being human together may be beneficial for productivity, she explained:

> I think part of it is that it just sets, helps calibrate us, or keeps us, I don't know. I love the word grounded. It keeps us remembering that everybody on the team is a person who has feelings and desires and other things going in their life and well for one thing if things are a little tense, we can be reminded it may not be related to this code we're writing at all. Like I may have something going on in my personal life, but my teammates will know that.

Jennifer’s answer reflects what Fleming and Sturdy (2009: 7) refer to as neo-normative control, or the efforts of the firm to elicit normative diversity. Where normative control seeks to establish normative uniformity, in that all workers uniformly develop values that are aligned with the firm, neonormative control “invites and supports the outside of work self (‘warts and all’) and not only the preferred ‘front stage’ corporate (specific) self.” In doing so, neo-normative control allows the firm to incorporate the entire worker into the production process, drawing upon workers’ feelings and experiences as an opportunity to improve team collaboration and efficiency. Like Jennifer, Sophie noted the importance of building relationships through being human at work as particularly useful for productivity. She stated that developing interpersonal relationships with coworkers is a “huge part of feeling free to interrupt someone if you have a question. If you have bonded in other ways, when you’re vulnerable with people, that builds trust. Teams that have that work better together and solve problems faster.” Workers on Agile teams demonstrated a willingness to help one another in order to meet collective deadlines and goals, espousing behavioral empathy towards struggling teammates. For example, Tyler, a Full Stack Developer, described his willingness to take on undesirable tasks, stating,

> If there’s a task that still needs to be done, I don’t think anyone wants to do, or even necessarily knows how to do it, they kind of want to push it on me because I know how to do it. Even though I don’t really want to, I’ll do it for the team.

The behavioral empathy that Tyler and others describe leads them to complete project goals more efficiently. This practice is also present in Just-in-Time workplaces “indeed, many of the less experienced members are reliant upon their fellow workers for help. Of course, while acting informally to help out their teammates, the operators are enlarging their own tasks...in line with the company’s requirement” (Delbridge 1995: 812). While literature clearly shows a correlation between productivity and positive team relationships (Akgun 2015), analyzing Agile illuminates how co-location and interdependence act as important scaffolding to support the cultivation of behavioral empathy.

Making the Invisible Visible

While mechanisms for eliciting commitment, as described above, help to construct a more flexible labor process in which workers have the autonomy and motivation to get work done (through the embeddedness of commitment in the labor process), cultivating strong team relationships, behavioral empathy, and commitment also helped to improve surveillance over team progress. As discussed in the previous chapter, Agile imposes direct control strategies to take software work "out of the shadows" through processes of quantification and performance tracking, or velocity. While these strategies operate as effective tools to track the progress of complex work like software, Agile *also* imposes responsible autonomy to improve surveillance over the workplace, another way of engaging in an effective hybrid approach to achieve Taylored Flexibility. Through the coupling of responsible autonomy as well as a strong sense of behavioral empathy, Agile is able to effectively impose mechanisms for improving surveillance that do not necessarily rely on peer scrutiny, as observed on many high-performance teams (Barker and Tompkins 1993; Ellway 2013; Sewell and Wilkinson 1992). Through peer scrutiny may allow firms to tighten visibility without needing the heavy hand of management by having team members scrutinize, surveil, and ultimately manage each other, behavioral empathy benefits firms as workers willingly report progress to the firm. What's more, as discussed in detail in Chapter IV, behavioral empathy compels workers to take on extra labor, sometimes that of middle management, without receiving extra pay or appropriate compensation. Thus, behavioral empathy is a useful disciplinary tool that deepens coordination *and* compels workers to pick up unexpected or unassigned tasks to improve flexible production, all while obscuring power relations in the workplace, as I discuss more below.

In the Agile context, the necessity of collaboration, cooperation, and commitment that results from interdependent work on Agile teams allowed firms to achieve greater surveillance over the production process, as workers routinely engaged in activities that made their work more visible. The Agile methodology includes a number of "ceremonies" in which workers provide progress updates. One popular ceremony is the "daily standup" meeting, which quite literally requires workers to stand during the meeting. Having workers physically stand was aimed at improving the speed and efficiency of the meeting. In one Agile training video, workers simulate an "unhealthy" standup, where one team member participates in the meeting while lounging in a beanbag. When discussing what made the standup simulation unhealthy, the Agile trainer asks the audience, "So what do you guys think is going on right now? What are all the dysfunctions that are happening?" and explains, "first of all, we've got we've got Mr. Comfortable over here who's just pulled up his entire beanbag, so that's obviously not going to work." While lying in a beanbag was joked as particularly egregious behavior in standup meetings, the trainer also pointed out the dysfunction of a pair of team members, asking the audience,

> did you also notice that both of them are sitting down in the standup?" The trainer follows up by asking, "Is it okay to lay in our beanbags? Should we stand up? You guys know why we stand up? Cuz it gets done faster. when we get comfortable, we tend to talk more. Stand up.

In addition to standing, the daily standup requires workers to answer three questions: What did you work on yesterday? What are you working on today? What roadblocks [issues] are you facing? Drawing upon Toyota's practice of having workers "stop the line" to immediately identify production defects (Womack et al. 1990), software workers are instructed to report defects or production impediments during these daily meetings. This mirrors the visibility that was implemented on Toyota's shopfloor, where defects are made visible to all workers so that "every time anything goes wrong anywhere in the plant, any employee who knows how to help runs to lend a hand" (Womack et al. 1990). As Meena, a Senior Product Manager put it, standup meetings "open the lines of communication and visibility."

Agile theory argues that by reporting roadblocks during daily standups, workers are able to get suggestions from their teammates for how to fix a problem so that they do not waste time unsuccessfully solving the issue on their own (Cockburn and Highsmith 2001). When a teammate identifies a roadblock, there is an expectation that either the team lead or other team member will help solve the problem quickly before moving on with their own tasks, as Jack provided an example of. Participants recognized that their interdependence upon one another to complete sprint goals required them to communicate, cooperate, and make their own progress visible. For example, Winston, a Software Developer, admitted that daily meetings may be useful on some days more than others, but highlighted that when tasks are more fragmented and interdependent, meetings become crucial for coordination. Winston stated, "when we have sprints [two-week delivery periods] that have a lot of smaller tasks the daily standups become a lot more important to knowing where everyone is at." When I asked Tyler, a Full Stack

Developer, why he was willing to share roadblocks in the daily standup meeting, he responded, “If I’m having a problem, I’m going to bring it up because if it’s not known it just going to either a) make me more stressed or b) make the problem worse.”

In some cases, workers who were not able to complete tasks within a sprint were reprimanded. Leslie, a Project Manager, found standups at her firm frustrating, reporting that she and her teammates would, indeed, indicate roadblocks to the Scrum Master in charge of running daily meetings. Yet, Leslie explained that her team’s Scrum Master had not been trained for the role, and was not effectively able to help developers solve any issues. This led to the team missing important deadlines set by executives and upper management, and team members were reprimanded by the disconnected boss of her team’s department for not working quickly enough. Rather than disciplining the “incompetent” Scrum Master, team members bore the brunt of higher management discipline.

Of course, co-location also improved visibility over production, as managers and workers sat next to one another. When asked about her interaction with teammates, Melanie responded,

> I sit next to them every day. I don’t have an office; my desk is right next to my juniorist programmer. Every morning we have a 15 minute stand up where they bring me up to date on what’s going on...every day for those 15 minutes it’s really dedicated to everybody being on the same page.

As Klein (1989) notes, workers in self organized teams who must constantly report progress are seldom able to “create ‘idle’ time, or to ‘hide’ work for ‘rainy’ weeks, etc.” Agile method also encourages the use of “visible information radiators.” As explained in an Agile training video, visible information radiators are used so that

> When you walk into the team room or when you come into an area, you should see right there on the walls the status of what they're doing, how they're going. What are their team norms, where are they at right now in terms of their tasks and deliverables? Everything is very visible in front of the team.

Co-located teams engaged in daily progress meetings offered firms the ability to improve upon project's management promise for greater surveillance, predictability, and calculability over otherwise "invisible" work.

Conclusion

This chapter has shown that alongside reconfigurations in the production process to establish direct control as discussed in Chapter 2, Agile has developed mechanisms to embed commitment into the software labor process, improving the ability to achieve Taylored Flexibility. In other words, I contribute to understandings of how firms elicit cooperate behavior among teammates, showing that mechanisms for achieving responsible autonomy such as a culture of support, helpfulness, and collaboration may be especially achievable with some of the scaffolding that Agile provides: namely, highly interdependent work, and emphasis on mutual accountability, and co-located workplaces. In this sense, workers are not motivated to cooperate solely through a strong corporate culture or an internalization of company values, or even extra-organizational values, but by the very structure of the production process itself. The combination of neo-normative control alongside the establishment of commitment through Agile mechanisms, I found tech firms to be particularly successful at eliciting hybrids of direct control and responsible autonomy without strong resistance from workers.

For example, co-located workspaces, where workers and managers work next to one another or share desk space, improve opportunities for workers to develop emotional and compassionate relationships to one another. Moreover, software workers' recognition that interdependent work increases the necessity for collaboration helps to cultivate their engagement in behavioral empathy, where they are willing to help take on their teammates' tasks even if it was not originally assigned to them. I also found that an emphasis on collaboration and behavioral empathy led workers to engage in Agile methodologies to improve surveillance over productivity, especially daily meetings, without engaging in especially scrutinizing behaviors over one another's work like has been found in other settings (Barker and Tompkins 1993; Ellway 2013; Sewell and Wilkinson 1992). Unlike peer scrutiny, however, behavioral empathy also had the impact of obscuring sources of control, as participants did not frame meetings as particularly intrusive, but often necessary for "staying on the same page."

Overall, through a reorganization of the labor process in a JIT manner, Agile suggests a need for firms to engage in both direct control and responsible autonomy in order to engage in the most cost effective, flexible, and predictable production of software, or, to achieve Taylored Flexibility.

CHAPTER IV

UNCLEAR, UNRECOGNIZED, AND UNCOMPENSATED: GENDERED COMMITMENT UNDER AGILE

> Involvement and empowerment are at the core of the new model of work, yet workers are compelled to engage in the organizational mechanisms by which they are achieved, by means of a multifaceted and decentered system of control...traditional control is coupled with a heightened job requirement to be more involved and responsible, often without added compensation or resources.
>
> Vicki Smith (1997: 333-334)

So far, this dissertation has extended labor process theory research on software work by showing the mechanisms through which employers aim to achieve Taylored Flexibility, the elusive combination of predictability and adaptability needed to make software work more productive. I have done this by analyzing how Agile, a lean project management methodology, reorganizes the software labor process. In Chapter II, I showed how Agile relies on aspects of traditional direct control strategies by adopting a Just-in-Time (JIT) fragmented work process, rendering knowledge work more calculable and thus trackable, and posing limits on the "spaces of play" in which software workers employ their discretion when solving software problems. In Chapter III, I analyzed the ways that Agile combines aspects of direct control with responsible autonomy, a control strategy based on eliciting the commitment of workers. I argued that commitment cannot only be understood as an ideological, discursive, or cultural strategy on the part of management, but that through the reorganization of the production process, namely, to JIT production, commitment gets embedded into the very fabric of production. Furthermore, through the cultivation of mutual accountability, where workers are assessed based on their team and not just their individual performance, co-located workspaces, and behavioral empathy, the organization is able to cultivate team-based commitments, resulting in workers' consent to the labor process, particularly regarding their participation in self and team surveillance and helping colleagues complete their work duties.

In this chapter, I continue to analyze the mechanisms of achieving Taylored Flexibility by examining the ways in which combinations of commitment and flexibility have gendered and racialized outcomes. I begin this chapter by discussing the ways in which the development of labor markets has been gender and racialized, specifically looking at the ways in which high technology is a white and masculine industry. Then, taking a feminist labor process theory approach, I argue that the project of Taylored Flexibility is the result of gendered and racialized evolutions in control over work, in the sense that the disproportionate representation of men in tech influences the kinds of control strategies used over tech work. Then, I discuss the ways in which inequality regimes, or the "loosely interrelated practices, processes, actions, and meanings that result in and maintain class, gender, and racial inequalities within particular organizations" (Acker 2006: 443) in Agile unfolds.

Specifically, I argue that an emphasis on mutable, porous, and flexible team assignments and job duties and a reliance on team-based commitments lead workers to take on extra and uncompensated work. This happens in two keys ways: 1) as new job duties and roles emerge in ambiguous and unclear ways, women and non-binary people disproportionately take on these positions, without being compensated by the firm, and 2) women and non-binary people take on the unrecognized but necessary labor of mitigating gendered and racialized tensions in the team in order to ensure the seamless flow of production. In some cases, workers bring issues of equity to the organizational level, but even in doing so, find themselves having to take on the extra and uncompensated labor of making the organization recognize and address gendered and racial inequalities. These findings complicate questions about equity in the flat organization, namely, that eliminating steep hierarchies can result in greater equality between workers in the firm (Acker 2006). Instead, I argue that the ways in which Agile organizes the labor process reflects gender-neutral assumptions, in which the firm does not recognize and directly benefits from inequality regimes in the labor process.

Tech and Labor Market Segregation

A historical account of the emergence of segregated labor markets, and the inequality regimes that unfold in particular workplaces, can help to frame the development and gendered outcomes of Agile and the software labor process. Following feminist scholars like Joan Acker (1992:112), we can analyze gendered outcomes in the organization of work by first acknowledging that "industrial capitalism is historically, and in the main continues to be, a white male project, in the sense that white men were and are the innovators, owners, and holders of power." Acker shows that two historical processes in particular, the race and gender-segregated labor force present at the emergence of industrial capitalism and the dichotomization of waged labor and reproductive labor, reveal how "capitalism as an organization of production and distribution is gendered and racialized" (Acker 1992: 111). For example, one can look at the ways that deskilling and degradation of work has unfolded in gendered and racialized ways. Alongside the process of deindustrialization, as briefly described in Chapter I, the United States economy has been increasingly bifurcated between "good" jobs and "bad" jobs (Kalleberg 2011). Stable, middle-class jobs seemingly began to disappear in the U.S. towards the turn of the century, replaced by professionalized high-wage work and routinized low-wage wage, reflective of "high-road" and "low-road" strategies. The high-road strategy has included the specialization and professionalization of work, requiring high levels of educational attainment and certifications, and offering high-wage, prestige, and benefits to workers. Low road strategies, in contrast, emphasize "deskilling," using little educational requirements to, in part, justify low compensation and typically no benefits to workers.[28]

Importantly, the processes of bifurcation has been racialized and gendered, with low-wage work typically relegated to women, Black, Indigenous, and people of color (BIPOC), migrant workers, and the elderly (Kalleberg 2011; Michel and Ben-Ishai 2016), while professional high wage jobs are disproportionately occupied by white men. Crompton and Jones (1984) and others have called this process the feminization of certain kinds of work, in which more routine, clerical, and deskilled jobs have been relegated to women, and have reproduced the idea that these jobs are women's work.

This feminization process has resulted in sex segregation, or the separation of men and women within and across industries, occupations, and workplace activities. Gender segregation can have many negative impacts on women particularly in male-dominated occupations, such as lower pay, being sexualized at work, being treated paternalistically, and being discriminated against and bullied by men (Padavic 1991).

[28] This has become particularly detrimental to workers as neoliberal restructuring has resulted in the disappearance of public goods and services, with migrant workers excluded from receiving state-subsidized resources altogether, despite being concentrated in low-wage occupations (Kalleberg 2013).

Particularly as jobs become segregated, work done primarily by women and BIPOC is devalued, with these jobs providing lower wages than those primarily occupied by whites and men (Cohen and Huffman 2003; Huffman and Cohen 2004). As part of this bifurcation process, tech work, a professionalized industry requiring math and computer science skills, has become one of the most male-dominated industries (BLS 2020).

Though high-tech industries are often touted as the forefront of innovation and progress, they lag in racial and gender equity. Within the technical industry, technical and non-technical occupations and managerial and non-managerial roles are divided along gender and racial lines. Drawing on data from the Bureau of Labor Statistics, in 2020, women made up 47% of the workforce in 2020, they only made up 25% of all computer and mathematical occupations (BLS 2020). Looking even closer at specific computer and mathematical occupations, gender inequalities emerge in 2020 women made up 45% of Web and Digital Interface Designers, an occupation with a median pay of $77,200 a year, while they made up only 19% of Software Developers, who earn a median pay of $110,140. Racial inequalities are also present for Black and Latinx workers: despite making up 10% and 8% of all computer related jobs, respectively, Black and Latinx workers each made up only 6% of software development jobs. Asian workers are particularly overrepresented in tech jobs, making up 23% of all computer related occupations despite making up only 6% of the workforce overall. However, it's important to note that while Asian workers are overrepresented in tech jobs compared to all other jobs, they are more likely to be foreign-born citizens holding H1-B visas, the visa that allows employers to temporarily employ workers without U.S. citizenship in specialty occupations. As Rodino-Colocino notes, "Indian H-1B workers are concentrated in the information technology (IT) sector, with 92 percent of H-1B petitions from 2001 in "computer related or engineering occupations" (Hira, 2003a: 14)." Though the H1-B visa program is meant to allow employers to hire the most qualified workers from any country, in practice, it allows employers to drive down labor costs. Rodino-Colocino (2007) found that "nationally, H1-B holders in the computer industry are paid 20 percent less than the average wage" (2007: 214). Based on interviews with 121 Indian H1-B workers, Chakravartty (2006) found that Indian H1-B workers employed through temporary agencies were most likely to experience longer hours and lower wages than American citizens. Thus, despite the overrepresentation of Asian workers in tech compared to the overall workforce, we can still understand that access to benefits and wages in the tech industry are segregated by race, citizenship, and gender, with American white men occupying more privileged positions. Moreover, the concentration of white men in professionalized occupations is intimately tied to the development of the labor processes within industries and control regimes within tech organizations, as we will see throughout this chapter.

The Feminist Critique of the Labor Process

Until now this dissertation has offered a labor process theory of software work by primarily identifying how Agile operates as a hybrid control regime that relies upon combinations of direct control and responsible autonomy. However, labor process analysis is incomplete without a conceptual framework that incorporates the gendered and racialized evolutions in the development of production processes, and the gendered and racialized ways that control regimes take shape to create particular labor processes (Smith 1995). Because capitalism itself is a gendered and racialized project (Acker 2006;

Robinson 1983), gender and race are necessarily components of the labor process. This feminist approach to labor process theory was sparked by "Braverman's silence on gender," which "opened up a rich area of research and debate about how the nexus of control, skill, consent, and resistance is gendered" (Smith 1995: 408).[29] Feminist scholars have pointed out that feminized work in particular has been subject to direct and despotic forms of control, while work that is considered more masculine is typically allowed more autonomy and involvement in production (Gottfried 1991; Tancred-Sherrif 1989). In other words, when Friedman writes that responsible autonomy has been more often applied to privileged workers and direct control to the marginalized, he hides the gendered and racialized character of modes of control, failing to identify that the "privileged" workers are particularly white and male, and the relegation of direct control to underprivileged workers is part of the process of feminization and racialization of low-wage work. It follows, then, that one way to make sense of the high wages and prestige found in software, and the reliance on a high-commitment model, lies in the fact that it is a masculine industry. In other words, a feminist and critical labor studies analysis of the software labor process must recognize that the masculinization of the occupation operates as an explanatory factor in the development of Taylored Flexibility, or a kind of control model that relies heavily upon commitment and responsible autonomy.

Complementing feminist research that criticizes labor process theory for its inattention to gender and race, I draw from the work of Joan Acker, (1990; 1992) one of

[29] The early feminist responses to the gender analysis gap within labor process theory was to develop a research agenda specifically looking at the experiences of women in the workplace (Lamphere 1985; di Leonardo 1985). While empirical studies of women workers were absent from the labor process theory literature, these early approaches still treated gender in an ad hoc manner, not identifying the ways that gender is embedded *within* the labor process, and shapes and is shaped by control, consent, and resistance at work (Smith 1995).

the key scholars analyzing gendered and racialized processes within organizations, who rejects the notion that workplaces are gender neutral. Acker argues that the reason white men have been particularly successful in highly paid professional occupations is because work itself has been centered on the notion that the worker is unencumbered, or a male worker whose primary loyalties lie with the corporation instead of domestic work and child rearing, duties most often associated with women's work (Acker 2006). These kinds of assumptions make up organizational structures, which are gendered in the sense that advantage and disadvantage, exploitation and control, action and emotion, meaning and identity, are patterned through and in terms of a distinction between male and female" (Acker 1990: 146). For example, a gendered lens may code aggressive men as "go-getters" but aggressive women as "bossy" (Wajcman 1998). Ultimately, the overrepresentation of white men in high-paying tech occupations reproduces racial and gender inequalities in the workplace and reinforces a culture that often excludes and discourages the involvement of women, Latinx, and Black workers.[30]

Gendered organizations are embedded within broader cultural narratives that naturalize gendered organizational practices. In general, cultural assumptions in the United States around men and women's intellectual abilities usually assign math and science skills to men and language and writing skills to women (Andre et al. 1999; Kahle and Lakes 1983). These assumptions are constructed within the cultural dualism of man/woman, which are co-constitutive with dualisms of masculine/feminine,

[30] Importantly, male-dominated industries are not inherently masculine, but develop and reinforce masculine cultures through organizational discourse, procedures, and cultural norms (Britton 2000).

rational/irrational and logical/emotional, so that men are understood to be rational and scientific, and women are thought to be emotional and communicative (Plumwood 1993). These dualistic cultural narratives naturalize the overrepresentation of men in technical occupations like computer programming and software engineering, and the concentration of women in marketing, graphic design, or customer service jobs in high-tech industries (Massey 1995).

Gender also intersects with race in the workplace, such that race is gendered and gender is racialized (Browne and Misra 2003). Nkomo (1992) details how most organizational research has mischaracterized organizations as race-neutral, and that the role of race and ethnicity in organizational norms and culture is either completely understated or not seen as relevant. She argues that discussions of race in organizational literature are usually ahistorical and decontextualized, taking on a Eurocentric view that normalizes and erases whiteness and casts racial/ethnic minorities as “others.” When incorporating race, most organizational literature deploys a narrow focus on discrimination and assimilation and ignores the racialized component of organizations, using a colorblind (Bonilla-Silva 2003) framework that obscures how organizational structure and culture are racially constructed. Ultimately, the hierarchies of race and gender *are* maintained in the workplace through social interactions and organizational norms. Reitman (2006), for example, provides a case study of whiteness in high-tech companies in Seattle, WA, showing how tech-work itself is conceived of racially, where dress (i.e., casual attire/no shoes), language (i.e., using Caucasian instead of white), and managerial positions are coded as “neutral” but are actually representative of white culture.

Inequality Regimes in the Flat Organization

When exploring the ways that inequality regimes inform the experience of marginalized workers in organizations, little research has looked specifically at inequality regimes within flat organizations, despite their growing popularity in the last few decades. Acker suggests that flat hierarchies may have the potential to shrink racial and gender inequalities that persist in hierarchical organizations (Acker 2006). This has long been an argument by proponents of the Human Resources Management movement, but, as Dickens (1998) showed, despite the "implication that the rise of 'HRM' represents a potential advance for gender - and other - equality...the gender equality assumption in the HRM model is part of the rhetoric rather than the reality." For example, in their analysis of work and employment survey in the UK in 2004 and 2011, Davies, McNabb, and Whitfield (2015) find that "both men and women in workplaces characterised by the presence of HPWs [high-performance workplaces] earn more than those in workplaces where such work practices are absent or less well developed," and that while this is good for workers overall, "women were found to exhibit a lower premium in pay associated with employment in high-performance workplaces compared with men." Ultimately, Dickens (1998, 27) suggests that "developments in flexibility appear to have utilised and underpinned rather than challenged existing gender segregation by industry."

Limited research has been conducted on how teams in flat organizational structures may be gendered. Some research has even suggested that team-based structures mitigate gender inequality (Kalev 2009). Others, however, have shown that women experience disadvantages on teams, particularly when they are male dominated. For example, Williams et al. (2012) found that team-based work can obscure the contribution that women on male-dominated teams make, as they may have more difficulty promoting themselves and the work they have accomplished compared to their male counterparts. Similarly, Corrington (2021) found that women on teams who act in more individualistic rather than communal ways are perceived as significantly less helpful than male counterparts who act in a comparable manner. I contribute to this literature by showing that when flat organizational structures like those promoted by Agile do not explicitly attend to issues of gender and race, they reproduce inequality regimes reflective of assumptions that the labor process is gender neutral. Below I show that inequality regimes in Agile emerge through the ways flexible work gets taken up and cooperative team relations are maintained.

The Gendered and Racialized Outcomes of Commitment under Agile

Through interviews with men, women, and non-binary tech workers employed on Agile teams, I found that an emphasis on flexibility and commitment, and the reliance upon team member' collaboration in order to get work done, leads the organization to be dependent upon women and non-binary people for taking on extra work to ensure that the production process runs smoothly, but does not provide adequate recognition or compensation for this labor, operating as an inequality regime. I argue that inequality regimes in flat organizational structures like Agile result from informal processes, in which workers have little formal recourse for compensation and recognition. In other words, without a policy to point to, I found that some workers, particularly women and non-binary people, experienced challenges to get recognized for the extra labor that they were putting in for the firm. This, again, operates as part of the regime of Taylored Flexibility, as informal processes in flat structures help to promote firm flexibility. These informal structures are in the firm's interest not only as a way to expend less resources on the heavy hand of management, but to create ambiguous organizational structures that make it difficult for workers to be appropriately compensated for their labor. In many ways, the firm is able to effectively but legitimately engage in wage theft. The firm benefits from employing workers on a salary basis and assigning workers job responsibilities in ambiguous structures, as they are legally protected from wage theft claims despite workers taking on roles for which they are not paid.

I also found that organizations benefited from a gender and race-neutral approach to eliciting commitment and promoting collaborative team relationships, that is sourced not only from the commitment that workers feel to one another, but the particular sense of collaboration and collective responsibility that women are socialized to feel and act upon in teams (Eagly, 2007; Eagly & Johnson 1990). As I have argued throughout this dissertation, cultivating strong team relationships is central to doing Agile: "Agility requires that teams have a common focus, mutual trust, and respect; a collaborative, but speedy, decision-making process; and the ability to deal with ambiguity" (Cockburn and Highsmith 2001: 132) Following a feminist understanding of work that recognizes that gender and race necessarily shape workplace interactions, tensions arising from sexism and racism on teams are likely to impact the ability for workers to collaborate to produce the commodity needed in a timely and effective manner. However, instead of the organization intervening, the implicit assumption of gender and racial neutrality meant that primarily women and non-binary people take on the extra labor of mitigating tensions that arise from sexism and racism at work, reflecting the ways that Agile is gendered and racialized. I argue that this unrecognized labor does not simply operate to preserve women, non-binary, and people of color's safety or dignity, but functions as another way to mitigate potential disruptions in the flow of production. In other words, firms directly benefit from the unrecognized labor performed by marginalized workers to navigate hostile spaces in two keys ways: first, in that responsibility for addressing gendered and racist interactions can be transferred from the organizational level to the individual level, saving the organization money for training or keeping the organization from having to fire problematic (racist and sexist) workers, and second, in that women's work to develop collaborative team relationships despite discriminatory experiences keeps the flow of production moving, ensuring firms can realize profit from team-produced commodities. Below, I rely on the accounts of participants to examine the ways that flat organizations are gendered and racialized.

When Flexibility Means More (Gendered)Work

Within flat organizations overall, job descriptions and job duties are expected to be flexible and to change. More "fluid, general and adaptable" job duties are beneficial to the firm, as they provide "the organization with greater freedom to maintain the flexibility required to successfully respond to external challenges" (Beltrán-Martín 2006: 21). In the Agile environment, creating more adaptable job duties is reflected by an emphasis on transforming software workers into generalizing specialists. Scott Ambler (2003), an Agile proponent and practitioner, explains that "a critical concept is that agilists need to move away from being narrowly focused specialists to become more of what I like to call generalizing specialists." The concept of generalizing specialists was repeated throughout the Agile Training videos, with one video offering the following definition: "Generalizing specialist is an interesting word…generalist doesn't mean that everyone can do anything, but it also means that we're moving away from being specialists." This emphasis on generalizing specialists is part of a movement within and beyond lean production toward functional flexibility, or "the reduction or elimination of job descriptions and demarcations," as well as "job rotation and the ability to work a number of different jobs" (Moody 1997: 95). Though taken up as a lean practice, this kind of flexibility operates, in the Taylorist sense, as a process for deskilling workers (Braverman 1974).

Constructing software workers as generalizing specialists is especially useful in a flexible organizational environment, as it supports the movement of workers from project to project, team to team, and to take on unexpected tasks on a particular team. Moreover, the high turnover rate in tech may be an especially compelling reason for firms to ensure workers are able to move to different teams and projects and work a number of different roles. The notion that software workers in Agile become generalizing specialists, and that jobs duties, tasks, team and project assignments are flexible, were reflected in participant experiences. Workers described experiences where new team roles suddenly appeared, or work was needed in a particular area but there was no clear bureaucratic process for assigning workers to the previously undefined tasks. Ultimately, I found that it was primarily women and non-binary people who took on these extra and unexpected roles. For example, Riley, a white Senior Software Engineer, explained that her company decided to run a trial period where teams would be assigned Product Managers, a customary role in Agile teams. Rather than hiring a new manager or officially promoting team members, Riley's manager requested that someone on her team volunteer to take on this new position:

> So, the company promoted this idea of using engineers as part-time Product Managers [PM]. Which has some advantages and some tradeoffs. And they started that program at least, it started affecting us at least 6 months after the newly formed team came along. So, we all sat around the room and our manager said is there anyone who wants to be PM and none of us really wanted to. Because it seemed really hard, and it seemed like a lot of extra work. Like it wasn't pitched as a just team lead position but like, here's more homework for you to do and homework that at least the four of us at the time found really difficult.

Despite her reservations, Riley ultimately ended up volunteering for the position. When I asked if the position came with a pay raise, Riley responded:

> No, unfortunately not. Definitely more responsibility and uh, honestly more fun, once I learned how to do it my way. And a few more headaches, um, but I think more exposure in a good way. And I think, honestly, more like a career benefiting exposure. The advantages are, so a couple, one I really do feel like it's good for my career as a senior because the next step above Senior Engineer is Lead Engineer where you're making decisions that affect the company as a whole. And you know this new position is quite related to that because you're helping your team determine the road map, but you're doing so because you have all that context. You're going to all these meetings with the product people, the ones who made the wider company strategic decisions, and you're finding out what your priorities are and then thinking about it with the team. How does what we do, how do the next few things we think we should do fit into those big priorities. And that kind of exposure and that kind of, again that kind of like doing some of the work of a lead engineer, I think is useful for careers. And again, exposure like people know your name and if you're doing a decent job, they'll be like oh wow they're doing alright.

As described in this excerpt, Riley was hopeful that taking on this uncompensated role would lead to promotion opportunities later on, though this was not a certain outcome explicitly promised by her employer. I suggest this ambiguity around what a worker needs to do in order to receive promotions helps to elicit worker consent extra unpaid work.

Importantly, Riley took on her new role with a deep commitment to flat or "ground up" organizational processes, using organizational ambiguities to her advantage in some ways. For example, though her company usually determines project priorities at the executive and upper-management levels, Riley began to gather data on best practices from her team members and from members of other teams to influence executive decisions. Through her relationships to other Product Managers, Riley was also able to orchestrate the switching of duties with other team members in order to enhance productivity and to match assignments to projects that software developers were most interested in. While Riley felt that this was particularly beneficial to her team, ultimately the position required a lot of uncompensated extra labor with only the unguaranteed hope for a future promotion.

Similarly, Spencer, a white non-binary Junior Product Manager, took on a leadership position after a number of workers were promoted out of that role. Spencer described not wanting their teammates to experience difficulties making deadlines, so they were willing to take on a leadership role despite only being formally recognized as a Product Analyst (which entailed less compensation) for almost a year. When I asked Spencer how long they had been in their current position, they responded:

> Spencer: This is actually only my second month with this title. I've been [laughs] doing that sort of work for a lot longer. Not 40 hours a week of the extra work, but basically, I was doing my entire other role plus maybe 15 hours a week [in this role]. I would finish that as quickly as possible so that I could get onto this Product Management stuff. I was still getting pulled into meetings and presenting to all of the same people and making those decisions, but it was um, like not recognized with a title change.
>
> Larissa: And were you being compensated financially for the extra work?
>
> Spencer: Um, kind of. So um, I have gotten raises, but they haven't always correlated with when my responsibilities increased, if that makes sense. Usually there's a 4–6-month lag time where like, I don't know if I'm expected to prove it or if they're just slow to get their stuff together. I have experienced, um, issues like, not being recognized for those things. Especially when it's been just like, a very drastic increase in responsibility.
>
> Larissa: For example?

> Spencer: I started off at like an entry level support position and was like, after two months basically the most senior person on the team besides the manager, between people leaving and getting promoted and um, then like yeah I guess it was people leaving and getting promoted, but then we like lost funding, and there were times when, or we didn't lose funding, we were trying to get a new round of funding so we weren't hiring new people but like, I was doing maybe 2 or 3 people's jobs through that summer. It was still within a 40-hour week, but I was just doing it so quickly. And my manager was also getting married that summer, so she was out for two weeks and there was a big fiasco where they messed up like $100,000 worth of payments, it was like the manager and I were getting back those funds, and I got back like every penny except 3 or 4 thousand dollars, and there was just like no, there was nothing. Like people that were in the position to get raises were like "oh my god if you hadn't stepped up, we would have been so screwed." But there was no action. And I was like, also kind of an acting manager, I was doing a lot of onboarding, I was meeting with like the C-level folks about products and it [the recognition] was just not happening.
>
> Larissa: And how did you end up taking on those responsibilities?
>
> Spencer: um, *they were just things that needed to happen*. We didn't have enough people, I just, I don't know, with like a support email que it just gets bigger and bigger if you're not getting through it. And luckily, I had just gotten them to hire my sister as an intern, my little sister, and she just could type so fast so I would take all the escalations [high priority problems] and she would take all the easy cases [problems] but somehow, we managed to get through that.
>
> Larissa: And so, did you ever bring up that you were taking care of all these responsibilities without being compensated?
>
> Spencer: my support manager was like the biggest advocate and um, she was the only woman manager. And just like they didn't disclose any of the past salary bans to her and so she had thought she had been mailed the salary I was getting, and I think I was like I think I was compensated pretty well based on like the title, but it was, the frustration of not getting a title beyond, like even now should I list this intermediate role as a support role 'cos I was doing so much more than that. Like if I put it in on a resume and so it's frustrating to just like, career wise, have that stalled out for more than a year just 'cos they didn't want to lose, they're like someone who is producing so much on this support team. They didn't want to pay for the backfilling just using that experience and yeah it was just very frustrating, and my manager was doing everything she thought she could and probably could do. And it's just like, the executives just weren't budging.

Ultimately, Spencer began to record conversations with executives about raises in order to hold executives accountable to their agreements, which did result in Spencer getting a raise (though they were never able to secure back pay, effectively meaning that Spencer's employers engaged in wage theft for a number of months). They continued,

> I refer to email, or slack messages, or whatever it is as, being like, "this is an established timeline, and on this day, we discussed that at a 6-month mark such and such would happen." And then bring that back at the 6-month mark and say, "hey you agreed to this." Which is just a pain.

In Spencer's case in particular, we can see how combinations of consent and resistance shape work arrangements. Ultimately, the necessity of making sure "things get done" and a strong reliance on committed team members led Spencer to take on work in which the organization was not compensating them. However, we can see the ways in which resistance led to Spencer being compensated for the labor, though even the act of recording conversations to ensure they got a raise was extra labor the organization was willing to let Spencer take on.

Maggie, a white Quality Assurance (QA) Scientist, explained having to take on the job duties of her team's manager, who she claimed was not suited for the position. Her manager, Harold, was originally hired as a QA Scientist and was suddenly promoted despite not applying for or expressing interest in a management position. Maggie explained that Harold acted as "a friend rather than a manager," requiring her to take on additional organizational tasks and responsibilities. She argued that because of Harold's hands-off approach to management, the QA department could "literally run itself," but, in fact, it was Maggie who specifically kept the department running. After months of taking on Harold's responsibilities, Maggie's patience reached its limit when a colleague working at an office located on the East Coast, where executives were also located, received a promotion with no explanation. Maggie and her coworkers organized a meeting with Harold that resulted in their getting flown out to the East Coast to air their grievances with the company’s head of Human Resources. After the meeting, Maggie was formally promoted to QA Engineering Lead, which came with a raise. Because of the organizational ambiguity that led to her coworkers’ promotion, Maggie was able to leverage the irrationality of the decision to get recognized and compensated for her labour.

Ava, a Central Asian woman, too, engaged in extra work, and explained how she ended up giving formal bi-weekly trainings to her colleagues without compensation. When I asked how she ended doing the trainings, she explained:

> Ava: It basically emerged out of multiple different, you know, ambiguous and ambivalent desires to do better by the customer support engineering department, where I used to be, ‘cos they get all the tickets that filter in about networking issues. So, you know my boss was like “since you have that rapport with the department already, would you be interested in facilitating this training?” And what I had imagined for it was that, you know, maybe I would get some help from my managers and others in the engineering department to come up with topics or maybe do slides, but nobody stepped up to do any of that. Eventually it became “okay you present a topic, you create the slide, and you also write the outline later on.” And I was like, this seems insane. It seems like I’m doing 3x my work.”
>
> Larissa: do you think if it ended up being successful, is this something that could lead you into a path to promotion or a raise?
>
> Ava: um, weirdly enough that’s something that really gets me here, is that I don’t think people see that. And they just, I think it’s one of those things, that people never fail to underestimate training and the value that it gives to the whole organization. So, people here don’t really understand that that’s a thing that we should value. And no. I truly don’t think that’s going to lead to anyone recognizing me.
>
> Larissa: So, what would be the biggest reason you were even willing to do it in the first place?
>
> Ava: um, I guess the idea that you come for that product stay for the people kind of thing. So, the people are the reason that you do the things, they’re your friends, they’re your coworkers, you’ve been in that situation where you struggled, and you wish someone would have helped you. Like I don't want to see my coworkers struggle and I don't want to see my friend’s struggle. And at the end of the day, you know you’re not all going to sit here and die in your chair hopefully, and you

> go off to different places and you want to make sure you build good relationships and rapport with people.

In Ava's example, we can see how the strong cultivation of team relationships resulted in behavioral empathy, or the willingness of team members to help one another with their work or issues.[31] Still, once Ava realized how much extra uncompensated labor she was putting into the firm, she finally set boundaries and began to step away from the trainings. Through these excerpts we can see how organizational ambiguities and an understanding of roles within the organization as adaptable had positive impacts for the firm, not just in the sense of creating more flexibility, but in constructing (gendered) informal structures in which workers take on extra uncompensated work.

Making the Work Work

Another key way in which I found women, non-binary, and people of color having to exert extra, unrecognized, and uncompensated labor was through the ways they mitigated gendered and racialized tensions in the labor process. I argue that these strategies taken by individual marginalized workers to create positive, cordial, or professional relationships with co-workers, necessary to maintain the flow of production, supplement a lack of organizational responses to inequity, discrimination, harassment, and exclusion. I argue that team-based work needed to accomplish tasks under Agile would not be possible without the (unrecognized) labor of marginalized workers to maintain team cooperation, showing how HCM under Agile operates as a gendered regime.

[31] For more discussion on behavioral empathy see Chapter III.

As part of flexible workplaces, workers routinely experienced being moved from team to team, which required them to engage in the extra labor of building new rapport with teammates who, especially in the context of Agile, they were required to build cooperative relationships. For instance, Linda, a Scrum Master, described being moved to a new team despite having a positive relationship with the old team she was assigned. Linda expressed a sense of understanding that her job requires her to be flexible. When I ask her if she has had any decision in the process of being moved, she answered, “Not really, but that’s okay. That’s kind of my job. I should be able to go to any team and work with them. That’s really what a Scrum Master should be able to do.” This process ended up causing Linda to have to engage in more work to develop a positive relationship with each team she moved to, as she had to begin the process of establishing rapport. When I asked how she felt about her new team, she responded with *their* feelings towards *her*, saying they “begrudgingly tolerate” her. She described the work she needed to engage in as a team leader to facilitate collaborative relationships in the team: “I have one gentleman who’s a little pokier, little spikier about his responses and I do have to say to him on occasion ‘we’re trying to maintain a safe environment here for everybody to be able to speak.’” The extra work that Linda had to employ in order to smoothen out tensions caused by a man on the team were common experiences for women.

For example, Ellie, a white Software Engineer, described the extra work that was, at times, necessary to establish the kinds of collaborative team relationships that are expected within Agile. Ellie explained that on a new team she was assigned, she was able to establish a positive relationship with most of her colleagues. She explained, though, having difficulty reading a male colleague with whom she felt tension. I asked her how she navigated that relationship, and she responded by reflecting on the ways in which *she* made changes to her modes of behavior in order to keep production running smoothly. She explained:

> At first it was kind of challenging. I definitely had moments where I didn't understand what he wanted from me and, like I wanted it to be, wanted the interactions to be a little, like, warmer. 'Cos I'm feeling like, I'm new and feeling uncomfortable. And then it's funny how, over time, I feel like I kind of see things about him that, I definitely see a lot that I appreciate, and have figured out that if I make certain adjustments things go better. And there's a part of me that's like, why do I have to adjust? So, it is the way that I've seen things work better, so I do. And, and it hasn't been that difficult, um, to make those adjustments in our interactions and I have benefited a lot from it and gotten a lot of really good information, like, I've figured out the way to get information from him that works, and he does have like, a ton, and that's kind of what I want. I know I will be more successful at this job and this team if I have as much of the background information as I can get. I feel like this person has this information here, so I want him to feel willing to be generous with that info. So, I've figured that out, how that works with him and I. And then I'm grateful for that. 'Cos it's not *that* hard. But I definitely think about it sometimes as, like, it isn't ideal."

Though Ellie minimized the work that she needs to do to get crucial information from a coworker, which is necessary for her to complete her work, meet sprint deadlines, and make sure the team as a whole meets their goals, the firm is certainly benefiting from not having to intervene in the tension that Ellie feels with her male colleagues. Instead, acting as if it were a gender-neutral project, the firm relies on the cultivation of commitment to ensure team members are working together collaboratively, which inevitably places the burden on women to create positive and productive relationships.

In other cases, women had to engage in the work of asserting themselves in order to gain the respect and cooperation from male colleagues, which was particularly relevant for women in managerial roles. For example, Barbara, a white Director of Engineering, explained the sexist response of a male colleague and friend when she was promoted into a managerial position. She told the story:

> I had a guy who was a coworker, and he was a friend. A work friend, and we were very friendly. And then I got promoted to be his manager. And I asked him, "will you be okay with me being your manager?" The reason I asked him was because we were friends. But he said, "I don't know, because I don't know if I'm okay with having a woman as my manager." He said that! I was speechless [laughs]. So I go, "well you better think about it." And I thought about that, and I thought well, I'm not gonna have him work for me if he can't do it. But what will they do with me? Will they put me in another group or something? I don't know. Anyway. Anyway, he went home for the weekend and uh, talked to his parents I guess, and came back and told me "yes," he could work for me [laughs].

This sexist encounter is not the only one that Barbara has experienced, especially in a leadership role in high-tech jobs. She explained to me that when in managerial roles, she developed a strategy for earning the respect and cooperation of team members, which essentially required her to take on a traditionally patriarchal masculine position. She explained:

> Barbara: I often had to go in and establish myself as the alpha in a project. and that was harder for me because I was female, and they were mostly male. And so, I developed techniques to do that rapidly in the first meetings.
>
> Larissa: What were some techniques?
>
> Barbara: So, don't let myself get interrupted, I interrupt them. Take time to prepare an agenda, go through my agenda, don't let them knock it off. Hand out the agenda, say "this is what we're going to talk about." Talk tech so that they don't think I'm just a Project Lead who doesn't understand technical stuff. But the best thing after you do that, don't let them interrupt you. If someone talks about some technical thing and they get it wrong, you gotta be the one to get in there and go no [clap] that's not true, and then you're done. You did the other things, and then you correct [the workers], and then that's it. And it worked for me.

So, on top of taking on managerial responsibilities, Barbara found herself having to employ extra strategies to assert herself in order to be an effective manager and to ensure team members got their work done. This excerpt shows that in order to engage in team-based work as a woman, taking on masculine tropes may become necessary extra work.

Heather, too, explained that part of working in a male-dominated industry has meant being "the only woman in the room." When I asked her how she navigated that, Heather responded "I've got really sharp elbows and I don't mind using them to get my way in there and asking questions." The sharp elbows and alpha male strategies employed by Heather and Barbara employ is aligned with research that shows how women navigate male-dominated spaces, which shows that women often manage disadvantages by assimilating to masculine work culture and suppressing their femininity (Dryburg 1999; Faulkner 2006; Maupin and Lehman 1994). However, little research has connected the ways in which these strategies not only allow women to be more successful in their jobs, but in the context of team-based work, allow the firm to ensure that the accumulation of capital continues seamlessly.

In addition to suppressing femininity or taking on a masculine disposition, research has shown that to manage male-dominated workplaces, women may distance themselves from other women, identifying instead with men they work with and adopting an "anti-woman approach." For example, women may participate in or remain silent when sexist or racist comments are made, downplay their femininity by wearing less makeup and less feminine clothing, and make efforts to distance themselves from coworkers who identify as women and align themselves with men (Powell et al 2009). Carrie, like Barbara and Heater, found that a masculine disposition allowed her greater ease in her relationships with employees and teammates, though she described this as being more natural for her rather than a source of extra work. When I asked her how she thought her identity impacted her experiences as a manager, she initially responded "I don't know if it has, honestly," before proceeding to explain how a more traditionally masculine disposition may be associated with an easier time for her in tech. She told me:

> There are sort of two things. One is that I am not, and I hate saying like, a typical woman 'cos that's not what I mean either, but I'm very direct. I speak my mind. I will not be shouted down. It's really hard to shame me. And two, I'm very, my interests are very male focused. And so, it's really easy for me to both participate in rooms where I theoretically should be intimidated, and it's very easy for me to make connections with other male managers because I can talk sports ball and I can talk about the home improvement thing I did this weekend. So, I think it can be a little bit easier, it seems like it's been pretty easy for me to cross that gap, where I've heard for other people it's not.

In this excerpt, we can see that Carrie makes explicit that connecting with male managers is the result of more easily fitting into masculine cultural norms, noting that this may not be the case for all women. Carrie's experiences reveal that for those who are not able to as naturally or authentically navigate masculine spaces, making connections and "crossing the gap" between masculinity and femininity does require extra labor, which I have argued, is necessary but unrecognized by the firm. Moreover, being able to successfully reconfigure one's femininity or mitigate gendered tensions is necessarily mediated by whiteness, as it was primarily white women who found success in either reproducing masculinity or finding other ways to make the work work with men on their teams.

Because the labor process of Agile, and, indeed, most work arrangements, require marginalized workers to routinely navigate sexism and racism, some women and non-binary people pursued efforts aimed at making organization aware of accountable for inequities. Yet, I found that reacting to uncompensated work required in the labor process by taking it to the organizational level required workers to engage in even more uncompensated labor in the form of equity work. Coined by Adia Wingfield (2019: 34), equity work refers to “the various forms of labor associated with making organizations more accessible to minority communities,” which results from the fact that “organizations abdicate the responsibility for creating diverse institutions.” In this research, I found that equity work took the form of marginalized workers calling out the gender-neutral assumptions made by their organizations, in order to create more equitable workplaces for women and BIPOC who work there. These attempts to change the organizations were met with varying success, which, I argue, was related to how much the firm could financially benefit from taking on an organizational responsibility to address a culture of discrimination and a structure of inequity, versus the cost of the change.

Engaging in equity work in order to change organizational practices operated as a response to the routine extra work that marginalized workers engage in to mitigate gender and race tensions, or sexism and racism. For example, Bailey, a white non-binary person, expressed frustration over the lack of compensation given to workers who are putting in the extra labor of trying to make organizational-level changes. Bailey explains:

> One woman has organized a weekly diversity meeting where she’ll prepare all these materials to talk about different initiatives, or publish work, or videos, and like, it became something like everyone was really proud of as a company, but she was never recognized for it. It’s just [shakes head]. It’s just so much energy that goes into it.

They further problematize the lack of compensation for equity work, stating that that firm primarily places the responsibility for this kind of labor on marginalized people, which, especially without compensation, is just functioning as a form of overwork. Bailey states:

> What I’ve been frustrated with is people are like “only these folks from diverse communities can lead these projects [on diversity] ‘cos we don’t want them to think we’re overpowering their voices.” When in reality, it’s taking those people away from building the skills, the marketable job things they can put on their

> resume. And it’s taken a lot of people refusing to take on additional things for them to realize.

In this excerpt, we see how equity work simultaneously operates as a form of resistance to routine requirements to take on additional work, yet equity work also takes the shape of uncompensated labor as workers spend extra time to present the organization with issues and solutions.

In another example Erin, a white woman, described the extra work she engaged in to advocate for racial justice training, in order to address instances in which her colleagues of color had to individually navigate and address racist encounters. Erin explained that she faced a lack of enthusiasm when advocating for racial justice training to her boss. She believes this is, in part, because she pushed for an especially in-depth training which could be costly or time consuming for the company. She stated that because she perceives online implicit bias training to be insufficient at actually challenging discriminatory behavior, and is skeptical about most corporate diversity training available, she has spent a significant amount of time outside of work researching what she feels would be more substantial training options. She explained that when she met with her boss, “he said yes right away,” but “it turns out his idea of training was very different from what I would like to have.” The training she was interested in consists of two full days during the week and a Saturday morning training, through the People’s Institute for Survival and Beyond, a “national and international collective of anti-racist, multicultural community organizers and educators.” Her boss mentioned that he wants any training that they offer to be mandatory and felt that he would be unable to require employees to attend on a Saturday, and worried about the employees having to find transportation. Still, Erin was hopeful that after “the hump of this particular project he’s stressed about," she could remind her boss of the training, and is coordinating phone calls with some of the trainers. She even told me that in regard to her boss's concern for finding Saturday transportation, "I have my heart set on this so much that I would drive around early in the morning and pick everybody up." Ultimately, I suggest that the training that Erin was advocating for was not met with more enthusiasm by her boss because it would have required blocking out two workdays for the training, and potentially even the company having to pay for transportation, on top of the cost of the training. In this case, we can see how Erin was willing to engage in extra labor to make the work work better for BIPOC workers, but because this extra labor was not more directly beneficial to the firm, it continued to go unrecognized and unused.

While many participants shared concerns like Erin about their company's shallow or absent attempts at addressing "political" issues such as sexism and racism, white men in particular expressed feelings that though women and people of color may be underrepresented in tech jobs, they were still more interested in pursuing a meritocratic approach to recruitment, showing sexist and racist assumptions about how work is organized. For example, Steven, a white man, criticized his company's response to equity and inclusion, stating that they were only willing to give it "lip service" rather than make more substantial changes. He stated,

> if there's a shooting somewhere or if someone gets run over by a car at a rally somewhere there's lots of emails coming out from like way upper management from like corporate saying like, "this isn't us, this is who we want to be, if anyone has any suggest--" you know it's just. It's cool, glad you said it. What are we doing through, it doesn't really mean anything.

Still, despite Steven's critical assessment of the company's approach to issues of racism, he stated that "at the same time I don't necessarily feel, like, an urgency to get more diversity. I just feel an urgency to get like better and better people." Steven was unable to recognize gendered or racialized processes that may code white men as particularly qualified for the job, taking a gender and colorblind approach where he just wanted to promote the inclusion of a meritocracy, where qualified people got the job. Arnold, a white man, expressed similar sentiments as Steven:

> I feel like the person who should get the job should be the person most qualified regardless of their race, not to say that affirmative action doesn't need to happen sometimes, like, it definitely does. We just, we shouldn't only be interviewing people who are from exotic places.

Arnold's resistance to interviewing people from "exotic" locations reflects more explicit racism, xenophobia, and discrimination, which is certainly not uncommon in tech work (Daniels 2015; Amrute 2020). Importantly, I interviewed Arnold in a joint interview with his partner, a white woman, working at the same firm. Immediately after Arnold's statement, Maggie jumped in to say "well, we're not...I've interviewed, like, ten people and most of them were white guys." In comparison, James, a Japanese man, expressed a strong rejection of recruitment processes that reproduce the overrepresentation of white men in tech. When I asked James if there was anything he could change about his workplace, what would it be, and he responded with concerns around issues of equity and inclusion. He answered:

> So, right now, one of the people leaving our company is a woman and they started the hiring process like, a month ago, with an emphasis on hiring sort of more non-white guys, because there's one other trans woman there and then I'm Asian, and then everybody else is white and male. So, the person that they ultimately sent an offer to was another white guy, which was disappointing. And so, they, my bosses, said when this other front-end developer leaves that he'd really like to hire a woman for it, which was good to hear because I've been pretty much waiting to, well I'm having a meeting with them this week just to sort of like, check in to get a raise and for them to ask me how I'm feeling about everything. So, I've been waiting for that meeting to say I'm pretty disappointed that this is who you, like, to sort of say this is what you're looking for and the end result just be another

> white person. I'll say that in the meeting, that I'm disappointed that that's what happened, but I'll also say that I'm a little reassured that he'll make it a focus to hire a non-white guy to replace this next role. The industry is just dominated by white men and it's a little frustrating especially to be replacing a woman with another white male. Kind of the wrong direction.

As James and others expressed, deep concern over racial and gender inequities were primarily felt by women, non-binary, and people of color. Ultimately, I found that women were mostly unable to find organizational or institutional resources to address issues of equity and inclusion, as their firms relied on them to solely navigate and smoothen out tensions on the team that may cause problems in the production process.

Conclusion

In this chapter I argue that the flexibility and commitment that characterizes the Agile software development process leads workers to taking on uncompensated, unrecognized, and unexpected labor in gendered and racialized ways. I show that because of the firm's need to be adaptable, the organization structures itself in such a way that workers can be moved from team to team and project to project to respond to changing priorities of the organization, they miss opportunities for raises and promotions, experiencing only lateral, rather than upward, mobility within the organization. In this sense, the firm benefits from flexibility not only in the ability to shift the amount of labor allocated to a particular project or set of tasks, but to avoid compensation schemes in which they would have to pay their workers more money. Furthermore, this chapter argues that the regime of Taylored Flexibility is the result of gendered and racialized evolutions in the labor process. I have showed that as work arrangements have developed since the emergence of industrial capitalism, processes of deskilling, degradation, and devaluing of labor has been gendered and racialized, such that it is primarily women and people of color relegated to the lowest paid and lowest-prestige jobs, with white men primarily occupational high wage professional occupations.

I also show that control regimes are constructed through gendered and racialized processes, such that direct and despotic forms of control are more often found in feminized and racialized occupations, while responsible autonomy particularly in the United States is more often used to control what Freidman called “privileged” workers, namely, whites and men. In other words, Taylored Flexibility as a strategy that relies on forms of direct control coupled with responsible autonomy, especially using a High-Commitment Management approach, to organize software work can be linked to the fact that tech is a primarily white and male industry. I also argue that the Taylored Flexibility regime, and managerial strategies to elicit cooperation and commitment are assumed by the organization to be gender-neutral, in the sense that there is no recognition that gendered interactions may occur, impacting the ability of the team to work collaboratively.

As a result of the firm’s invisibilization of gendered interactions, women, non-binary, and people of color take on the unrecognized labor of negotiating and managing charged interactions with their coworkers, though it is primarily white workers who are able to be effective with these strategies, reflecting the ways that whiteness is entrenched in the organization. I show that firms benefit from this extra labor of marginalized folks in two ways: 1) the firm does not have to expend time or money to make interventions to mitigate sexist and racist encounters 2) the firm is able to ensure that the production process, which is heavily dependent on workers cooperation and collaboration, runs smoothly in order to ensure profit continues to be secured. Ultimately, women, non-binary people, and workers of color do make attempts to hold the organization responsible for addressing issues of equity and diversity, but, I found, faced challenges when the organization viewed these responses as potentially costly, and still required them to engage in extra work.

CHAPTER V

POLITICIZING TECH: ALTERNATIVE ORGANIZING BY THE TECH ALLIANCE

> Struggles are active and creative, in several senses. They represent the working through of structural influences, they mediate effects from outside the capital-labour relation, and they have a dynamic and history - indeed, a logic - of their own: as they develop, they create understandings about how work shall be performed.
>
> P.K. Edwards (1990: 129).

In this dissertation I have primarily addressed control over the labor process with a focus on managerial control in order to show how Agile operates as a control regime to achieve Taylored Flexibility. While managerial control has been a key aspect of theorizing for labor process researchers, the "'control of labour' has two sides -- the attempts at controlling labour's activities by capital and its agents, and the control exerted by labour itself over what takes place on the shop-floor and beyond" (Delbridge 1995: 805). The exertion of control by workers is typically referred to as resistance, or the effort of workers to mitigate or reject managerial control methodologies, which can take individual or collective shape (Hodson 1995). In this chapter, I discuss the motivating factors that encouraged workers to take individual and collective action in the workplace as they aimed to improve and gain more control over the conditions of their work. I do this by analyzing the resistance (and theorize the lack thereof) of workers to the Agile methodology, as well as the collective organizing strategies of the Tech Alliance, an informal organization of workers in and around the tech industry. Through a pilot study of the Tech Alliance, as well as a discussion of resistance for those not affiliated with Tech Alliance, I analyze the emergence of a perhaps unexpected motivation for collective action: the political implications of the products that workers are producing.

While labor organizing typically centers around "bread and butter" issues such as wages, benefits, and hours (Moody 1980), the Tech Alliance tends to focus its organizing efforts on developing ways to resist building technologies that reproduce and deepen exploitation, imperialism, xenophobia, and climate disaster. They aim to achieve this through alternative organizing strategies, where an emphasis on unionization in one shop is replaced by an effort to organize workers across occupations and organizations, such that they are empowered to withhold their labor to build projects for which they are morally opposed. I suggest that this distinctive motivation and style of organizing can be explained, in part, by the ways in which Agile's emphasis of commitment to the team and to the product, rather than the organization, discourages and leaves little room for resistance to the project management methodology itself, but opens up possibilities for resistance to the particular product being built, as well as a political, legal, and economic context in which unions appear increasingly unsuited to models needed for labor organizing.

Consent and Resistance in the New Economy

Worker resistance and managerial control are dialectically related, in that the exertion and methodology of one impact the development of the other. For example, we have seen that the widespread use of despotic control under Taylorism led to mass collective resistance by workers, which, in turn, encouraged the development of new forms of production such as Toyota's participation schemes. Importantly, though the worker and employer are situated within a relation of structural antagonism, capital and labor are also inherently dualistic, in that they are dependent upon one another to reproduce the capitalist mode of production (Cressy and MacInnes 1980). In this sense, worker resistance cannot necessarily be taken as given (Delbridge 1995). As P.K. Edwards (1986: 42) argues, “workers do not simply enter work and then seek means of resistance. Instead, they find means of living with the system as they find it.” To be sure, while means of survival may counter the organization’s desire for employee behavior and managerial control strategies, individual forms of resistance that do not disrupt the logic of accumulation may function more as forms of consent to the labor process than result in significant transformations in the relations between workers and employers (Burawoy 1979).

In the context of lean production, disruptive resistance has been particularly difficult for workers to achieve. For example, in his study of workers in a JIT plant, Delbridge (1995) found that although workers sometimes “misbehave” or engage in individual acts of resistance, he did not observe resistance that was beneficial or liberating to the entire team. He notes,

> Individual workers might exert some influence over their tasks, but this normally had a detrimental effect on another worker elsewhere on the line. There were clear norms that operators were expected to meet in the eyes of their co-workers. Anyone not meeting these would run the risk of being verbally abused and ostracized by other members of the line who had to work harder to cover them (813).

In my research on Agile, I found that an increasing interdependence of work tasks, as well as worker control over the allocation of tasks within the team, left little room for enacting resistance to the particular task or the pacing of production as it could result in negative impacts for the worker. Moreover, while scholars have found increased peer-surveillance to operate as a panoptic mode of control over workers, in lieu of direct management (Kinnie, Hutchinson and Purcell 2000; Sewell and Wilkinson 1992), I found that Agile’s allowance for workers to “be humans together” and develop a sense of empathy towards one another did not translate into peer scrutiny, but into what felt like to workers as peer support. As a result, workers were more willing to self-surveil without perceiving the visibility of their performance as particularly coercive or abusive and were ultimately willing to consent to self-surveillance in exchange for autonomy (Deetz 1998).

To be sure, scholars have observed stronger forms of resistance towards HCM regimes compared to lean production, particularly when management initiatives are perceived by workers as shallow, inauthentic, or hypocritical, and especially in periods of organizational transformation to lean production practices from other modes of production (Ezzamel, Willmott, and Worthington 2001). For example, Ezzamel and Willmott (1998: 358) found that an organization that attempted to transform to a team-based model led to "the unintended effect of fermenting hostility towards the managerial goal of making the teams fully self-managed." Because new participative or post-bureaucratic firms must "present an image of 'open' and 'flexible' workplaces," while still retaining tight controls over the labor process, the contradictions between rhetoric and practice become more obvious to workers (Upadhya 2009: 10). Vallas (2003: 304) too, found that the contradiction between management rhetoric and practice drew attention "to the limited authority that workers were actually allowed," and that "team systems tended to heighten worker suspicion and distrust to foster patterns of solidarity that were difficult for managers to control." When teamwork is combined with electronic forms of surveillance, distrust can be particularly heightened. In fact, Townsend (2005) provides evidence that the cooperative nature of teamwork can allow team members to develop collective strategies for resisting the managerial use of electronic surveillance.

However, in the context of Agile, like Delbridge, I observed little resistance from workers to the mode of production, and little collective action. Though there were some moments of resistance to the labor process, such as Carrie protecting her team from the "bullshit of the company," as described in Chapter III, and Riley organizing a swapping of task assignments across teams described in Chapter IV, I did not find evidence of collective resistance to Agile *per se*. The most common instances of resistance were in regard to pay, where workers either collected data to prove to their employers why they should be paid at a higher rate, or where workers exited the organization to find work elsewhere. Other forms of resistance as described in Chapter 4 were motivated by marginalized workers' concerns over workplace inequities.

Though I did not find workers to be particularly agitated around Agile project management, I follow Crowly, Payne and Kennedy (2014) in suggesting that acceptance of teamwork does not necessarily reflect that workers are "manipulated or subject to managerial co-optation." Rather, workers may accept teamwork

> because they have evaluated their circumstances favorably in comparison to previous experiences inside and outside the organization, as well as their perception of arrangements available to workers in general and have determined that returns to teamwork are sufficient to justify their embrace (500).

Moreover, I argue that workers may not be exerting forms of resistance found in other HCM settings, in which workers distance themselves from the values of the firm, because Agile does not necessarily demand commitment to the organization, instead emphasizing commitment to your team and commitment to the product that you are producing. Importantly, I argue that these particular work arrangements, including an emphasis on workers' commitment to the products they are producing, have influenced the mode of resistance that I found when analyzing the Tech Alliance, namely, a resistance to the production of workers' labor, as I discuss in more detail below. Before moving onto the motivations and strategies of the Tech Alliance, though, I discuss some of the other barriers to unionization as a particular mode of organizing by tech workers in this study who were not members of the Tech Alliance.

The Challenge of Unions in High-Tech

High-tech workers are presumed to be especially difficult to organize into traditional labor unions compared to other workers. In 2019, 3.8 percent of workers in computer and mathematical occupations (which include software programmers and software developers) were members of unions, compared to 10.3 percent of employed workers overall in the United States (BLS 2020). The most commonly cited example of high-tech workers successfully unionizing is the Washington Alliance of Technology Workers (WashTech), a union founded by temporary Microsoft employees in 1997, who affiliated with the Communications Workers of America in 1999. The majority of WashTech members are those who work as contingent contractors, with lower pay and more precarious working conditions compared to tech workers who are full-time employees (Van Jaarsveld 2004). WashTech relies upon a combination of traditional and nontraditional union activities, which include mutual benefits, collective bargaining, and political advocacy for legislation that benefits high-tech workers (Van Jaarsveld 2004).

WashTech was able to achieve a number of wins, most notably, supporting a $97 million Microsoft settlement with workers who were denied access to Microsoft's Employee Stock Purchase Plan and their Savings Plus Plan (Van Jaasrveld 2004). WashTech also successfully partnered with SPEEA, a professional union representing engineers and technical workers in the aerospace industry, to lobby the General

Accounting Office to “undertake a study of offshoring of IT and other white-collar jobs.” Their political campaigns have not gone without challenges, though. The power of high-tech big business over legislative and political campaigns have proved barriers to pro-labor laws and electing political leaders. WashTech has also had difficulty pursuing traditional collective bargaining strategies because they did not have enough members to certify a bargaining unit. Employers’ right to refuse to bargain with WashTech members has limited workers’ negotiating power. However, as Brophy (2006: 5) points out, “this structure allows people to retain their membership to WashTech across volatile employment periods and regardless of the company they work for.” WashTech members’ contingent status certainly calls attention to the potential ill-suitedness of unions (already constrained by an anti-worker political and legal system) to some of the tech workforce.

Still, WashTech’s mutual benefit activities, primarily offering important workplace information and job opportunities to website visitors, have helped to mitigate employers’ attempts to mystify and obscure working conditions for contingent workers. Digital information about employers can function as an organizing tool that can successfully inform workers of their rights and potential working conditions. Yet, as Van Jaarsveld (2004) notes, because members and non-members alike can access WashTech’s online materials they are also disincentives from official membership, namely, paying dues. Unsurprisingly, then, though WashTech had 18,500 members in 2006, only 1,500 of those paid dues, leading to WashTech's heavy reliance on CWA funding. WashTech’s partnership with CWA, or affiliations with national unions in general, may also pose challenges for labor organizing in the high-tech industry. For example, many WashTech members were hesitant to affiliate with CWA and formalize their organization as a union after learning that a number of unions supported a bill that exempted contract workers from eligibility for overtime pay (Van Jaarsveld 2004).

While WashTech and more recent attempts of contingent software workers to unionize have continued to be of interest to workers, there are very few full-time software workers who are pursuing unionization. Milton (2003) offers an explanation for why tech workers may not be compelled to join a union, all stemming from their disidentification with unions goals and organizing models. Milton suggests that tech workers may feel their working environments are mostly positive, perceiving long hours as a necessary concession for the long-term potential for high-paying and rewarding work. Receiving generous wages and autonomy presents one hurdle to seeing the need for a union, as well as a "pervasive culture of association with the employer," and even hostility to unions, which are frequently seen as "archaic, undemocratic, and generally undesirable" (Brophy 2006: 631). Indeed, Milton (2003) found that tech workers identified the turbulence, flexibility, and creativity associated with tech work as characteristics that are antithetical to unions.

Too Cushy for a Union

For participants in this study, views of unions ranged from supportive, uncertain, opposed, and critical. Workers in the Tech Alliance are best described as critical, as I analyze in more detail in the sections that follow. For those not affiliated with the Tech Alliance, a common explanation for lack of union representation in the tech industry centered around the notion that tech workers have "cushy" jobs that may not require a union, or at least help to buttress the perception that unions are unnecessary. For those who held positive views of unions, the idea of tech job "cushiness" operated as an obscuring factor of the genuine benefits of unions. Casey, for example, explained that "people in 'cushier' positions are more likely to think they don't need to be unionized, and that if they do attempt to unionize their entire branch or company could get shut down." Still, Casey felt that unions would probably be beneficial to more contingent tech workers, explaining that these tech jobs may be especially "vulnerable to salary pressure, market pressure, those types of things." Holly held a similar view, but revealed an implicit assumption that unions primarily help to mitigate the most dangerous working conditions like health and safety hazards. She explained that she thinks unionization has not come up in her workplace because

> you might need a union to like, dial back some really terrible working conditions. And in tech, in general, the working conditions are pretty good. You're not like, in an area that's unsafe or unclean. You might be sitting at a shitty table with a laptop on your knees or something but it's not like, crazy unsafe like working in a kitchen where there are open flames and OSHA violations.

Holly, like others, noted that in contrast to the "cushy" tech jobs, unions might be particularly suited to places

> like coding sweatshops, where they're taking advantage of people who need a foot in the door, and they're being asked to work, like, 80 hours a week. And you know being salaried is pretty scary when your employer can tell you at any time that you have to work 80 hours a week or you'll get fired.

Ava also noted that tech workers may not fully understand what unions can do:

> I think that people in tech, maybe a lot of people in tech, live cushier lives than most other positions or as workers of the global world. So, I think they think of themselves as not needing a union or like, even understanding what unions do. Like, unions gave us the 8-hour workday and all the things that you enjoy um, but like people don't actually recognize that that's what unions do. They're kind of like, you know, it's like people make them the devil in the media. So, I think that people are not aware of all the things they're missing out on.

The perception of unions as the devil that Ava described functioned as a barrier to workers' interest in unionization was expressed by some workers themselves. Steven and

Carrie, who had both previously worked at a unionized workplace in the health and education sectors, respectively, had negative experiences with unions that kept them from being interested in unionization in the future. Carrie stated,

> I hated being a part of a union. You can't fire people that suck. And so, we were worse at our jobs because people were protected and there was no incentive to be better and they just did a bad job, and then you couldn't even budge them out of that job.

Steven expressed a similar sentiment, viewing unions as a source of unnecessary antagonism in the workplace. Importantly, Steven was in a middle management position when he was employed at a unionized firm. He held mixed views, explaining,

> I was in management which meant I was not allowed to be a part of the union. I saw a lot of benefits for the union, and I saw a lot of like um, I don't know if they [the union] were a detriment to the employees. It was an interesting position to be on both sides of it. That's my one experience with a union. And overall, I would say like, man, it was tough to respect them to be honest with you. Individually. Like, just those individuals were tough to respect. They were very combative, there wasn't a lot of understanding of where the organization needs to be, like the mission, focus, and there was a lot of protecting of people that really should be let go. Like, this person, this person is never on time, and when they're here they're an absolute detriment, like I need to fire this person. I need to get rid of them they're a total detriment, and then it's impossible. And oftentimes the only way to get rid of them is to trick someone else into hiring them. I saw that happen all the time, too. But for the really good people, I saw one of my guys was having a really hard time getting his pay raised, like a comparable rate, and the union really went to bat and really drove the organization to do a reassessment of all salaries. And so that one man's issue turned out to be an issue for a lot of people and the union drove that conversation in partnership. So that was a really good example of how it could work."

Steven's perception of unions reflects the view that conflict can and should be mitigated when unnecessary, and union presence is primarily beneficial when there is what he views as more legitimate wrongdoing on the part of management. Other mixed views were that unionization would be, as Spencer called them, "a big risky move," especially for more vulnerable workers, and without "any amount of expertise," it was unclear what the drawbacks would be. Ellie, for example, struggled to take a position on unions, stating,

> Well, I mean, most of the time I think “oh they’re awesome.” But I have this one woman friend of mine who studies, I guess like, economics and thinks that the country, well she has a lot of things to say about how maybe they’re not as awesome. And I’m like, she’s really smart and really knows her stuff, and I’m curious about the fact that she would say that, and she has a lot of stuff to back that up, but I’ve been way too tired and overwhelmed to dig into it, so at this point my own personal experience working before at company with a union is like, they’re awesome! With the caveat that there may be other important key points that I'd love to look into some day.”

As described in these excerpts, participants’ opinions of unions reflected a range of positions, from more direct hostility, to abstract support, to uncertainty and confusion.

The Union Challenge in the New Economy

Confusion around the role of unions can be explained, in part, by the decline of union membership particularly over the last fifty years that has resulted from neoliberal restructuring. The 1930s saw one of the most politically and legally supportive environments for union organizing, which resulted from the mass collective resistance to workers particularly to Taylorist production processes (Smith 1997). After the passage of the National Labor Relations Act in 1935, which gave workers the right to unionize, hundreds of thousands of industrial workers pursued unionization (Voss and Sherman 2000). However, state-level changes to minimize the efficacy of unions grew sharply over the following decades, from the passage of the Taft-Hartley act of 1947, which limited the circumstances under which workers could strike, to 1981, when Ronald Regan famously fired more than 11,000 striking air traffic controllers (Voss and Sherman 2000). Between 1950 and 1990 the number of workers fired for attempting to organize grew nine times, despite this being a direct violation of the National Labor Relations Board

(Milkman and Ott 2014). By 1999, only 14 percent of workers in the United States were members of a union, and in 2021 only 10.8 percent of wage and salary workers were members of unions (BLS 2021).

Though anti-union laws can account for a significant explanation of union decline, union organizing models in the United States have also been criticized for their, at times, narrow and bureaucratic approaches to labor organizing. Business unionism models, which became especially popular in the mid-20th century, have been linked to worker empathy and disempowerment. As Lopez (2004: 131) explains, the business or service model of unionism is organized by a structure in which "paid staff perform all representation for members – lead[ing] to worker passivity and to the equation of a strong union with one that offers 'good service.'" Alternatives to business unionism models increased in popularity in the late 20th century, with the rise of social movement unionism, in which workers pursue organizing strategies to "translate a 'union' issue into a social justice issue" (Lopez 2004: 165). As business unionism can rely on particularly rigid hierarchies, exclusion of women and Black, Indigenous, and people of color, and conforming to a bureaucratic model, social movement unionism has appeared as a more viable approach to workplace organizing tactics.

Still, the growth of contingent workers–independent contractors, temporary workers, and part-time workers–limits the feasibility of unionization and has, in some cases, weakened the protections that union contracts typically aim to secure. For example, a number of unions have negotiated two-tiered collective bargaining agreements in which part-time workers do not enjoy the same range of benefits as full-time workers, deepening the inequities between full-time and part-time work (Hoeller

2014; Martin and Peterson 2017). Although independent contractors may, in some cases, be legally able to unionize, they do not enjoy the same protections under the NLRA as other unionized workers, forgoing the right to strike without retaliation and not being able to bargain with employers over all the issues that full-time union members can (King 2014; Jost 2011). Independent contractors can also have "up to five or six 'employers,' making it almost impossible to even identify a bargaining unit" (Fitzgerald 1999; Brophy 2006, 5)." These political and legal constraints may account for at least some of the lack of unionization in high-tech sectors, where a large number of contingent workers are employed. Ultimately, I argue that the emergence of the Tech Alliance's political organizing motivations and strategies, which I will discuss in more detail below, are the result of an anti-union legal context, critiques of the pitfalls of traditional union organizing, and an employment context in tech where workers are paid relatively high wages and good benefits under a control regime in which autonomy and commitment obscure power relations between workers and employers. Yet, the emphasis on commitment under the Agile regime may provide a basis for workers organizing for more autonomy over the output of production, as I discuss below.

Politicizing Production: New Motivations to Organize

Generally, labor process research on workplace resistance analyzes the ways in which workers take collective action within their own organization, typically emerging out of poor working conditions. Burawoy (1979: 4) importantly noted that "activities on the shopfloor cannot be understood outside of the ideological and political realms of production." Building upon Burawoy, Ezzamel, Willmott and Worthington (2001: 1056) argue that "workers may favour (or devise) lines of resistance that are shaped at least as much by investments in particular political or ideological values as by calculations about job security." While values of individualism, entrepreneurship, and a political skepticism of unions may have kept some workers from devising more individual or collective resistance to Agile, I found that it was precisely political convictions and the political context in which workers were building technology that motivated workers to organize through the Tech Alliance. When asked how they got involved in the Tech Alliance, the majority of interviewees identified the election of Donald Trump in 2016 as a motivating factor. A number of events hosted by Tech Alliance explicitly cite Trump's policies as a cause for organizing. For example, following Trump's Muslim Travel Ban, Tech Alliance San Francisco hosted an event titled, "SF tech workers say: No Ban No Wall, Sanctuary for All." Tech workers, including janitors and food service workers in the tech industry, walked out of work and rallied to call on "tech campuses to be sanctuary campuses—for all workers in Silicon Valley." In this event and throughout interviews members made explicit connections between political issues and workplace arrangements and experiences. Though workers noted that Tech Alliance offered an outlet to participate in activism, this activism is intimately rooted in a recognition of the conflict between capital and labor, a conflict that tech workers realized was imploring their firms to build products with racialized, gendered, and classed political implications for the benefit of high-paying contrast that would increase their profits. In other words, Tech Alliance engagement in activism around immigration, harassment, or imperialism is articulated as constitutive of workplace issues, issues of how much control workers have over the products of their labor.

For example, Kinley, a Tech Alliance Seattle member, shared that at some of the first meetings she attended in 2017, participants had conversations like, “this work we’re doing is connected to what’s going on politically and with the government *and* my workplace.” Kinley explained that one of the first “pitches” that an organizer at a meeting made for why tech workers should be organized is the need to address the question, “what would you do if Trump asked your company to build the Muslim registry?” Paul, a member involved in Tech Alliance Minneapolis who helped organize a chapter in Chicago, also stated that one of the first discussions at a Tech Alliance meeting in 2018 was around Trump’s use of Twitter and the responsibility of Twitter leadership. He explained,

> if Twitter is being used for inciting violence, clearly Twitter leadership isn’t going to be incentivized to stop it. Basically, it would be contrary to the business model to stop that kind of engagement on their platform. So where is that power going to come from in order to force an action like that?

Kinley went on to cite historical examples of tech’s entwinement in politics and fascism, comparing IBM’s decision to make technology used to catalog people sent to concentration camps to contemporary politics. She stated that at these first Tech Alliance meetings, “you had people talking about technology being used for oppression.” Similarly, Ivan, a Tech Alliance San Francisco member, explained that he first sought out the Tech Alliance because he was unsatisfied with his role in production. He stated, “I wanted to do more than create a product or improve upon a product that maybe doesn’t align with my personal beliefs.” To be sure, a key organizing motivation *and* strategy for Tech Alliance members is the politicization of production, a recognition of the intersections between worker autonomy, technology, profit, and oppression. Interestingly, there is a history in tech of workers protesting against the political implications of their labor. In 1969 the Computer People for Peace protested the tech company's involvement in supplying technology used in the Vietnam war:

> The group pressured the Association for Computing Machinery (ACM) to adopt an anti-war position and NYC members collected 320 signatures on petitions against the military's use of technology. They raised funds to pay for the bail of a fellow programmer, Clark Squire [sic], who was arrested for his activity with the Black Panthers.[32] The group also attended an event at NYU to support work and research stoppages in universities.

Tech workers in the 1960s and 1970s also staged protests against their company's stance and involvement in upholding apartheid in South Africa.[33]

The politicization of tech production has become a key organizing tool that has helped to build the Tech Alliance. In an interview for a podcast, one member of Tech Alliance San Francisco stated that when workers found out that many of their companies had donated money to oppose a proposition to fund homelessness, "that really agitated a lot of workers. They were like, 'why is my company spending money on this? I don't have a say in this and they're doing this thing I'm against.'" The participant explained that this recognition of a difference between company and worker interests allows Tech Alliance to "draw the lines of struggle between the workforce and management." The approach of disjointing the interests of labor and capital is an important strategy in the tech industry and professional organizations overall, as scholars have noted that association with employers is particularly salient in high-wage, creative, and knowledge-based occupations and acts as a barrier for organizing (Brophy 2006; Milton 2003). Thus,

[32] Though born Clark Edward Squire, the former Black Panther Party and Black Liberation Army member goes by the name Sundiata Acoli.

[33] A full archive of tech worker collective actions and other resources on organizing in tech can be found at https://data.collectiveaction.tech/. Collective Action in Tech is a project being maintained by a group of tech workers, organizers, and researchers including Ben Tarnoff, Clarissa Redwine, JS Tan, Kristen Sheets, Nataliya Nedzhvetskaya, and Sarvesh Rajasekaran.

at the same time that Tech Alliance agitates workers to recognize the political implications of both the commodities they produce and the surplus that is extracted from them vis-à-vis their company, the organization is reinvigorating the conflict between labor and capital that professional industries have for so long tried to obscure (Brophy 2008). As a San Francisco member put it in an interview for a podcast, "It isn't a particularly bread and butter issue, it has a lot of more political implications than you usually see in workplace organizing." As reflected in workers' stories, the political implications of production outside of the four walls of the organization was a key organizing motive for workers. I suggest that this is largely because working conditions in tech for full-time employees that workers typically organize around, such as wages, benefits, or dignity tend to be less of an issue for tech workers. In this case, I suggest that workers are able to take up larger organizing projects on issues that extend beyond the shop floor.

The Tech Alliance Strategy: Building Worker Power for the New Economy

The informal mission statement that appears on the Tech Alliance website encapsulates the organizations' nod to traditional labor organizing. The statement reads, "guided by our vision for an inclusive & equitable tech industry, Tech Alliance organizes to build worker power through rank & file self-organization and education." Throughout interviews with Tech Alliance members and Tech Alliance online resources, traditional union and labor organizing terms like "agitate,' "rank-and-file," and "organizing conversation," appear. In fact, one interview began with a Tech Alliance member telling me that she was happy to participate in the research, as being interviewed is less emotionally exhausting than having the organizing conversation, a 4-step labor organizing technique aimed to compel an uninterested worker to join a union. Most literature assessing the viability of unionization in the tech industry suggests that tech workers generally have had little contact with unions or lack a collective memory of union and labor struggles, which leads to their lack of consideration for the option of unionizing. Though some members of Tech Alliance had not had experience with unions prior to joining their organization, Tech Alliance was a clear conduit for resurrecting a collective and historical understanding of labor struggles, and the role of unions in those.

Yet, interviews with Tech Alliance members as well as observation of the Portland chapter meeting revealed that tech workers are, perhaps, skeptical and concerned about unionization as a goal, though perhaps not for the same reasons that literature would suggest or as other workers in this study expressed. Though Milton (2003) points out that tech workers may not see unions as the best solution to what they may perceive as individual problems in the workplace, Tech Alliance may not see unions as the most viable solution because members perceive unions as reformist, offering too many concessions to management and too little liberation for workers. To be sure, Milton (2003) and Brophy (2006) acknowledge that tech workers may see unions as too stagnant or not creative enough to solve the problems of the tech workplace. Moreover, lack of unionization could be more reflective of the fact that union models based on 20th century work arrangements may not be best suited to address contemporary labor struggles (Benner & Dean 2000). Labor scholars and activists alike have noted the decline of union membership in the U.S. across occupations and industries that have resulted from neoliberal restructuring, what has been characterized by weakened labor laws and changing, more precarious and flexible work arrangements that appear unsuited for to the union model (Brophy and de Peuter 2007). As Benner and Dean (2000) put it:

> In the present environment of rapid change and complex outsourcing arrangements, the dominant forms of employee representation, which are based on the post-World War II industrial relations system, are increasingly ineffective. Hence, there is an urgent need to develop new models of representation that are more appropriate for our altered economic structure.

Interestingly, many Tech Alliance members view unions not as inherently antithetical to tech work, but as potentially too conservative to win the kinds of workplace struggles that demand more radical action. As one member stated,

> We're in a unique opportunity to look at the problems of organized labor, and how organized labor has betrayed a lot of the workers: they sign no strike contracts, they're complicit with management, there's all kinds of problems so right now we're in a position to analyze that and create something different.

The framing of unions as conservative, or at least not the top priority to strategically build power in the tech industry, has heavily informed the organizational methodologies of the Tech Alliance. Tech Alliance's organizational structure operates as a strategic alternative to unions, one that members hope will allow for a more democratic and grassroots organizing model.

Tech Alliance has also explicitly drawn upon anarcho-syndicalism as an organizing strategy through one of their Learning Clubs. Learning Clubs, which I discuss in more detail below, are topical events where members read academic and non-academic writing about a particular topic. One topic for the Learning Club was titled, "Organized Labor in the Spanish Revolution," which included academic literature on anarcho-syndicalism in Spain, paired readings from the Anarchist Library, an open-source resource for anarchist writing. Consistent with Tech Alliance reflections of the relationship between contemporary politics and early 20th century struggles, one of the discussion questions for this event asked, “how does the history of the Spanish anarchist and syndicalism position reflect our own time?”

When describing the formation of Tech Alliance San Francisco, members held a common narrative that some of the original activists were compelled and aligned with anarchist politics. Because Tech Alliance members are largely influenced by an anarchist approach, they have made explicit decisions to keep their organization flat and flexible. Kinley, a member of Tech Alliance Seattle, explained some of the process around how members made choices about maintaining the group’s unstructured organizational approach:

> We had a little, one day mini retreat probably about six months in to try to figure out what we were doing. Some ideas that came out of that that we ended up not implementing, like, we wrote some bylaws and decided that that was a bad idea... Earlier on there were more big group discussions and trying to vote on things, and that was harder for new people who didn’t feel like they had a lot of context. We don’t have a definition really of who can vote except who is there. Our definition of membership is basically, “have you ever been to a meeting?”

She explained that the voting process seemed to be a barrier of entry for folks, and because membership is “flexible,” the group felt that they could not develop a clear and equitable way to determine who has the authority to vote on a particular issue. Kinley continued that during the process of trying to figure out how to run and structure the organization a few members attended labor notes, a conference for union leaders and labor activists. She explained,

> one thing that one of our members brought back from that was from another community group, the idea of having work-sessions during general meetings. So, we started doing that and that’s made a big difference in our effectiveness. Like we did have projects going on before, we’re kind of a

> project-based organization and trying to focus more on the projects and less on like, we're going to vote about what to do.

Kinley also explained that the Seattle chapter decided they would not participate in electoral politics, viewing endorsements as a shallow gesture to organizing that can be better replaced by work to support a political candidate—such as canvassing—if a member should choose to organize a Tech Alliance project in that way.

Benji, another member of Tech Alliance Seattle, explained the organization this way: "Tech Alliance is not a 501c3 or 501c4, kind of political action group, or any of these things. It's an informal group: We don't have elected leadership; we don't have bylaws." Noam, a member of Tech Alliance San Francisco, also viewed the organizations' reliance on self-organizing without a formal structure as a strength. He explained,

> We have this real DIY kind of approach, do it ourselves you know? And that's really important because it allows us to have our own autonomy and it requires a lot of motivation and relies on self-activity to get stuff done. And I think that's really important.

Similarly, in a podcast on Tech Alliance organizing, a San Francisco chapter member stated that Tech Alliance has

> a really strong emphasis on leadership development and also self-leadership development. And focusing really on our own mutual self-education because what we really aim to do is build a reproducible network of leaders.

Ultimately, Tech Alliance members do not necessarily find the traditional union structure to be compelling and view alternative structures to be better suited for leadership development and addressing the kinds of issues that their members face, namely, how to resist producing unethical technologies. Nevertheless, Tech

Alliance is actively invested in educating members on a range of organizing strategies, union and beyond.

Education and Consciousness Raising

Though the 21st century may bear resemblances to the 19th century, an important organizing tool for the Tech Alliance that is made much more possible in the digital age is access to information that can be used to reinvigorate a collective memory of struggle. Reigniting a narrative to frame the relationship between capital and labor as ridden with conflict is particularly important for 21st century organizing, as the last thirty years have been framed by what Brophy (2008) identifies as the liberal-democratic story of "friction-free capitalism" which confuses "labour struggle and organization in general with the difficulties of the established trade union movement in particular." Yet, rather than view the difficulties of trade unionism as a signal that labor organizing is unnecessary, Tech Alliance equips itself with the knowledge of historical struggle in order to determine potentially new and hybrid forms of organizing. In this way, Tech Alliance follows the model of many worker-centers, which aim to build leaders from the ground up and empower rank-and-file members to make decisions and take action. Tech Alliance's self-education model not only allows for a more grassroots organizing approach but instills historical memory of struggle in the collective.

One of the key organizing strategies of Tech Alliance is the Learning Club, a regular event in which members propose topics to share readings on and discuss. Any worker can propose a topic in an online form, and Learning Clubs tend to be facilitated by members, though some Learning Clubs have invited labor organizers from other sectors, allowing for coalition building. For example, one Learning Club was led by Fan

Shigang and Li Wen to discuss the book *Striking to Survive: Workers' Resistance to Factory Relocations in China*. Learning club topics cover a wide swath of issues, represented in some of the following discussion questions listed on the Learning Club web page: "What is Solidarity Unionism and how does it apply to the tech industry?" "How can we envision an economic system in the tech industry that centers the needs of workers, communities, and the planet?" and "What does it mean to be an anarchist in terms of coordination and organizing?"

Members explained that the Learning Club deeply informs some of the organizing strategies of Tech Alliance, as members aim to use research and history to shape their organizational model and approach. For example, Benji stated that

> Last year a small group of us did the learning club on Beverly Silver's book called *Forces of Labor*. Reading and discussing that material was, I think, foundational for a number of us, especially the way that we, like, analyzed power. In workplaces in the marketplace, some of us have, like, skills that are highly in demand and high salary and those are all different forms of power that Silver discusses in that book.

Dana, too, expressed the importance of Silver's book on her own thinking about tech work. She explained, "The biggest thing I read that really changed the way I thought about the industry was the book *Forces of Labor*. It talks a lot about a big picture history of labor militancy and the conflict between labor and capital."

Members of Tech Alliance also expressed how their involvement on Slack, an online communication and sharing platform, helped to cultivate a particular understanding of power, labor, and organizing. For example, Paul explained that he attended a meeting for tech workers in Chicago and heard about Tech Alliance through another participant, who invited him to join the Tech Alliance Slack channel. Once joining the channel, Paul explained that he "I was exposed to more discourse and reading articles that people would post and also me and one of the other people turned that first event into a continuing series of discussion groups in Chicago so we kind of spearheaded what became that chapter of Tech Alliance." Tech Alliance merges digital and in-person strategies for sharing information in order to develop a comprehensive approach to leadership development and education.

Conclusion

In this chapter, I have highlighted some of the organizing motivations and strategies being taken up by the Tech Alliance in their efforts to build worker power in the tech industry. Interestingly, workers in this organization see an intimate relationship between the technologies they are building in their workplaces, the business logic of their firms, and broader political movements that use technology for nefarious purposes: inciting violence, perpetuating bigotry, and deepening inequalities. In response to these concerns, tech workers in Tech Alliance come together through an organizing model based on social movement activism, working towards the common good, and rank-and-file decision-making and action. In this sense, Tech Alliance has generally avoided the pursuit of formal unionization. Though Tech Alliance members do not express anti-union sentiment, they largely see the 21st century as parallel with eras without union protection, offering opportunities to develop new organizing strategies. As one member put it in a podcast interview about Tech Alliance, tech is at "a starting place that is comparable to a lot of industries in the 19th century in terms of what shape it could take. It's very nascent. It will take a lot of experimentation."

One of the organizing strategies that Tech Alliance uses is the Learning Club, a regular meeting where members choose a topic to select readings on and have discussions about. Members also build coalitions with other organizations in order to support their resistance efforts. Tech Alliance tends to prioritize deep membership over the quantity of membership, preferring to have fewer actively involved members as opposed to many disengaged members. This means that Tech Alliance is still very much in the early stages of their organizing work, in a phase in which building trust and collaboration in an industry that can lend itself to more isolationist behavior is the primary organizing goal. The organizing strategies that Tech Alliance will deploy are, in some ways, yet to be seen, as developing an organizational structure fit to address the political implications of tech work is one of the first steps to organizing.

This chapter shows that in order to capture the new organizing strategies of workers in jobs like tech, researchers must expand their vision of what collective organizing looks like and where worker resistance takes place. Though I found that there was little collective resistance to the Agile project management method specifically, if it was the only site in which I looked for resistance, I would have excluded forms of collective organizing that extend beyond the workplace. In this sense, workers must be understood more fully as political actors, who may engage in forms of resistance to the political implications of their work.

CHAPTER VI

CONCLUSION

> We must reinsert human beings, in all their rounded, messy, vulnerable materiality – and the complexity of their antagonistic social relations – at the very centre of our analysis.
>
> Ursula Huws (1999: 52)

Software development is known for its enigmatic nature, "high uncertainty, high interdependence, and high complexity" (Alder 2005: 406), raising questions for managers and scholars alike about the most efficient organizational structures and strategies to make this work more productive. Traditional direct control strategies developed to organize manufacturing work such as routinization, standardization, and a reliance on top-down organizational structures, referred to as Taylorism or Fordism, have become viewed as unsuited to the production of knowledge work like software (Castells, 1996; Drucker 1988; Locke, Kochan, and Piore, 1995; Powell and Snellman 2004; Lloyd and Sveiby, 1989). This is not only because knowledge work is considered complex and immaterial or invisible, making it difficult to be routinized or directly controlled by a supervisor, but also because Taylorist forms of control seem to clash with the disposition of professional and knowledge-based workers who require autonomy to be motivated to work (Kelloway and Barling 2000). As a result, scholars and managers alike have begun to analyze the best strategies to make knowledge-based work like software more productive.

The question of how to make software work more productive is particularly relevant in the contemporary knowledge-economy, where knowledge-based work like software production has become a more and more central form of labor for economic growth. In fact, according to a study in 2020, software work accounted for $1.9 trillion of the United States' Gross Domestic Product (GDP), an increase of 17.1 percent in two years.[34] Thus, as knowledge-based work like software is simultaneously more central to the growth of the economy and viewed as more difficult to manage than other forms of labor, firms have spent the last few decades developing new strategies for organizing knowledge-based work, or, what Marx called revolutionizing the labor process. This dissertation has explicated an investigation into forms of managerial control over the software labor process by responding to the following research questions:

- Has capital become dependent upon the technical labor of knowledge workers like software engineers, shifting the balance of power in favor of labor?
- What kinds of demands can businesses make over the labor of professional knowledge workers?
- What organizational forms are needed to support the accumulation of value from the knowledge-based worker?
- How can capital best extract labor power from knowledge workers, or how can organizations make workers *do* knowledge work?
- How do workers respond to these managerial control regimes and organizational structures?

I have aimed to answer these questions through an investigation of Agile, a lean project management methodology that is widely used to organize software work. Drawing upon 45 interviews with managers, workers, and Agile consultants, I argue that Agile operates as methodology to achieve Taylored Flexibility: a strategy that draws upon elements of

[34] https://software.org/wp-content/uploads/2021SoftwareJobs.pdf

Taylor’s scientific management, Toyota’s lean production, and High-Commitment Management (HCM) schemes to establish a more flexible but still calculable and predictable labor process. Below, I summarize the findings of this research by describing the ways that Taylored Flexibility is established, and its impacts on the software labor process as argued throughout this dissertation.

Establishing Taylored Flexibility

Agile first emerged as an alternative to what is called the Waterfall approach to software development, a sequential and rigid production process that came to be associated with inflexible organizational structures, unsuited to the rapid and evolving nature of software production. Agile was developed as part of the movement towards improving upon project management methodologies like Waterfall that were failing to deliver results by drawing upon Toyota’s lean production practices to revolutionize project work. Despite existing for almost two decades, Agile remains largely absent from labor process analysis of software.[35] Because Agile itself has been understudied, the linkages between Agile and Toyota have gone without theorization in the labor process theory tradition, making it an exciting area for critical analysis of managerial control strategies in knowledge industries. Throughout this dissertation, I dispel some of the myths regarding how control is achieved over knowledge-based work, by showing the ways that managerial strategies used in manufacturing settings have inspired managerial control strategies over software. I show that in order to make software work more productive, Agile relies on a hybrid approach to managerial control, including elements of what Friedman (1977) called direct control and responsible autonomy. By taking a

[35] For an exception see Briand and Hodgson 2013.

hybrid approach to control, Agile helps firms to achieve Taylored Flexibility, or the creation of a flexible, predictable, and calculable labor process as a means to enlarging surplus.

I examine direct control strategies primarily in Chapter II, by beginning with an explanation of Taylor's scientific management methodology, considered one of the first dominant managerial strategies that emerged as a means to organize the labor process at the start of industrial capitalism. Scientific management was a system that arranged work by breaking down the production process into its smallest components, and having workers be responsible for only working on that component. The scientific management methodology also set standards for the pacing of work by timing how long it took the fastest worker to complete a task, and then held this as the measure by which other workers should labor. In this sense, work is managed *scientifically*, in that it is based upon empirical information that measures workers' productive capacities. This process was famously adopted by Henry Ford, who used scientific management to organize the labor process in order to engage in mass production. Yet, by the mid-1950s the profit margins that mass production could ensure in the United States appeared to reach a limit. Though the benefits of mass-produced goods using Taylorist methods included cheapening the cost of each unit of materials and providing direct control over the workforce, the process can be highly wasteful, and thus costly, because this methodology is particularly inflexible. As flexibility became central to maintaining a competitive advantage in an increasingly digitalized and knowledge-based economy, the rigid and hierarchical model of Taylorism proved a less and less suitable methodology. Moreover, Taylorist methodologies could be unbearable for the workers, who were subject to routinized manual labor and a deskilling of their formerly craft abilities. As a result, workers began to collectively organize to improve their working conditions, which largely resulted in the establishment of unions, initially threatening the stability of the Taylorist approach.

As Taylorist mass production methodologies appeared to be more and more problematic, Taiichi Ohno championed a more flexible, more participatory production process called the Toyota Production System, most commonly known as lean production. One of the central priorities of lean production is minimizing *muda*, or waste. This was primarily achieved through the introduction of a Just-In-Time (JIT) production process. Rather than keep large amounts of products in hand, as was done in the Taylorist or Fordist production system, lean production aims to minimize waste by ordering a limited quantity of inventory, and producing and delivering products as needed. Lean production also relied on workers "stopping the line" during the production process in order to identify defects immediately rather than at the end of production, as was the practice in Ford's factory. Still, as I argue in Chapter II, lean production is not a starkly distinct production process from Taylor's scientific management, but rather, can be considered a hybridized production process that relies on elements of scientific management alongside leaner production processes (Fujimoto 1999).

In understanding the principles of Taylorism and Toyotism, and the ways in which managerial strategies are employed in hybridized forms, we can see how (JIT) production in particular has had an impact on the way that Agile organizes the software labor process. In Chapter II, I analyze three ways in which Agile adopts a JIT production process in order to render software development processes more flexible and less costly:

through the goals of maximizing the amount of work not done, delivering as fast as possible, and breaking down or fragmenting work. I show that workers are instructed to produce only *just enough* to satisfy the customer, discouraged from adding "gold plating" or extra features to the final product that the customer could have lived without. I argue that this has the consequence of narrowing the "spaces for play" available to software workers as they go about their work, revealing trends towards a saturation of workers' skills. In other words, workers reach limits on how much creativity and innovation they can imbue the product with, as priorities of delivering just enough and as fast as possible can conflict with the extra time it may take to add more components to the product. Furthermore, I show how a reliance on JIT software production leads to a fragmentation of the production process, as work is broken down into its smallest working chunks. Finally, in this chapter I show how direct control is used to improve surveillance, particularly through the quantification of workers' progress and "visible information radiators," a methodology deriving from Taylor's scientific management approach. Ultimately, I argue that while software workers provide the necessary skills needed to develop software products, the logic of capital leads firms to prioritize producing software quickly and cheaply over relying on software workers' creative and innovative labor to expand profit. Moreover, I show that software work as a form of knowledge-production is not insulated from the direct control regime of management, and that even work that appears to be invisible can be measurable through the application of scientific management methodologies.

While direct control strategies help to elicit both flexibility and predictability, in Chapter III, I explicate the ways in which responsible autonomy is used to achieve

Taylored Flexibility. I argue that strategies for eliciting commitment through responsible autonomy work in tandem with direct control strategies, as lean JIT production processes function, in part, as a mechanism for embedding commitment into the labor process. As work becomes increasingly task interdependent, a result of the JIT fragmented production process described in Chapter II, workers are compelled to help one another and to work collaboratively because each worker's task is directly tied to the ability of another person to complete their tasks, and vice versa. Additionally, commitment is embedded into the labor process through two other mechanisms: the construction of mutual accountability and an emphasis on co-location. Where mutual accountability operates as a way to encourage workers to cooperate, by workers being assessed based on their team, rather than just their individual performance, co-location spatially organizes the software labor process so that workers and managers all labor in shared spaced. The centrality of spatial coordination in the Agile methodologies challenges assumptions that are commonly made about how software work gets done, especially since tech work is particularly amenable to being completed remotely. However, Agile specifically emphasizes in-person coordination, as it is seen as a way to ensure stronger team relationships in which workers are more willingly to share information with one another quickly and help one another meet the goals of the sprint.

Embedding commitment into the labor process helps the firm to achieve Taylored Flexibility by cultivating team cooperation and a sense of behavioral empathy, defined the willingness of teammates to help one another out on work that isn't their own, that results from a sense of empathy for teammates who may be struggling to meet deadlines or solve problems relevant to their assigned task (Akgün et al. 2015). Workers quickly respond to issues that may come up for teammates through the combination of a willingness to help one another out, and by having the autonomy to re-assign and take on extra tasks without having to go through bureaucratic red tape to do so. At the same time, I found that a strong sense of commitment and team collaboration also led to increased surveillance, in the sense that workers were more willing to share updates about their progress, making the production process more visible and calculable to the firm. This was primarily done in in-person daily standup meetings, in which workers answer questions each day about what they had worked on the day before, what they are working on today, and what barriers they might be facing to accomplish their tasks.

While workers take on extra work as a result of behavioral empathy and commitment, I also found that workers take on extra and uncompensated roles and duties as a result of gendered and racialized processes, as I discuss in Chapter III. In this chapter I argue that analyses of the evolution and development of managerial control regimes over particular kinds of work must recognize that these processes are gendered and racialized. In this chapter I discuss the development of the segmented and bifurcated labor market in the United States, in which good jobs with high wages, benefits, and levels of autonomy are primarily occupied by white American men, while jobs with low wages, limited to no benefits, precarious working conditions, and more routinized and direct control strategies are primarily jobs done by women and people of color (Crompton and Jones 1984; Cohen and Huffman 2003; Kalleberg 2011; Michel and Ben-Ishai 2016) The overrepresentation of white men in tech work in particular can be seen as a result of this bifurcation process, as skills like science, technology, math and engineering skills are typically associated with masculine traits, reproducing the notion that tech work is primarily men’s work (Andre et al. 1999; Kahle and Lakes 1983; Massey 1995). Thus, understanding why Taylored Flexibility is the particular strategy used to organize software work requires an acknowledgement that tech work is gendered as masculine work.

Knowing that Taylored Flexibility as a strategy used to organize software work is, in part, the result of the masculinization of the tech industry, this chapter explores the ways in which Agile organizes the software labor process in ways that are gendered and racialized. I argue that an emphasis on firms being adaptable and flexible and a reliance on cooperative team-based work to get things done, alongside a gender-neutral logic in which organizations refuse to acknowledge the ways that interactions and informal processes in the workplace are patterned by gender and race (Acker 2006), leads to primarily women and non-binary people taking on extra, unrecognized, and uncompensated labor. Moreover, I show that the firm relies on women and non-binary people’s unrecognized labor of mitigating gendered and racialized conflicts on teams, in order to ensure team relationships remain cooperative so that the production process continues to flow smoothly. In this sense, Taylored Flexibility is achieved in gendered ways, in which women and non-binary peoples’ extra labor allows the firm to remain flexible, as well as increase profits as work goes uncompensated.

In Chapter V, I explain that while I found workers do deploy moments of resistance in order to receive compensation, recognition, or to reorganize work in their favor, the overall combination of high wages, benefits, the ability to exit a job because of lack of labor market saturation, and commitment schemes that obscure control mechanisms, account for the ways in which workers consent to Taylored

Flexibility. However, though workers in this study did not develop collective organizing strategies to resist or confront Agile methodologies, I do argue that, despite common perceptions of software workers as uninterested in collective organizing, software workers *do* develop collective strategies to reframe and address workplace issues. Though some research suggests that software workers do not pursue collective organizing as a result of their ideological and political opposition to unions as a particular form of organizing (Milton 2003), software workers may also be resistant to unionization as a result of neoliberal transformations that make unionization untenable for reorganizing work in the new economy.

While most tech workers remain unrepresented by unions, this chapter highlights the new organizing strategies of the Tech Alliance, an informal organization of tech workers and allies aimed at building worker power in the tech industry. This chapter discusses the strategies that tech workers are using to tackle labor issues and beyond in a 21st century context. Tech Alliance members are primarily motivated to organize because they see the products of their labor as political. They articulate organizing in the workplace—including cross-sector and cross-occupational organizing–as necessary to resist the production of technology used to support imperialism, xenophobia, discrimination and exploitation. In order to organize, though, Tech Alliance tends to favor social movement and anarchist organizing models over traditional union models, viewing unions as potentially too conservative, top-down, and limited for the types of radical organizing they deem necessary for the 21st century. Tech Alliance's alternative strategies include peer-to-peer education that merges academic research with on-the-ground praxis, in-person and project-oriented general membership meetings, coalition building and organizing “for the common good,” special events, and digital communication for organizing. In sum, Tech Alliance’s work reveals that organizing efforts in the tech industry remain vibrant despite a lack of union representation.

Ultimately, through an analysis of Taylored Flexibility this dissertation shows that organizations continue to make demands over knowledge-based workers like software workers, in which the priority is enlarging surplus through minimized costs and greater control over the labor process. I argue that organizational structures that allow firms to achieve both predictability and flexibility, particularly through the use of a Just-in-Time production process, methodologies to quantify work, vertical (top-down) and horizontal (team-based) surveillance mechanisms, unclear job duties and descriptions, and autonomy for workers to determine how to allocate tasks, is especially effective at making knowledge work more productive while also mitigating conflict between workers and employers.

Limitations

While the methodological approach taken to conduct this research allowed me to get an overview of Agile practices and their impacts on software workers, future research might consider the limitations of this approach, and develop alternative routes to researching Agile. For example, given that Agile is adapted in different ways at different firms depending on firm size, project goals, etc., future studies examining the impacts of Agile on the software labor process may benefit from setting stricter parameters of who can be included in the study. In this study, I included workers and managers at firms in any industry and size, as long as the participant considered themself a software worker (contributed to the software development life cycle) and worked on a team that used

Agile. The limitations of this approach include difficulty making comparative cases across workers' experiences. Though this study is able to develop an initial analysis of the Agile software labor process, the fact that Agile may be implemented in varying ways across firms may lead researchers to discover different implications based on how Agile is adopted by particular firms.

Researchers who want to study Agile should also consider if they want to investigate entire organizations who operate on the Agile methodology, or the use of Agile on particular teams or departments within the firm. Scholars who have conducted research on Agile teams in the context of a firm that is not entirely Agile have shown the ways that contradictions and tensions emerge between the team and the larger organization (Hodgson and Briand 2013). Research on Agile organizations could provide important insights into the ways that the entire firm takes on a particular managerial methodology, and how these impact particular kinds of teams in the firm. Researchers may also find differences across firms of different sizes and maturity, so parameters could include setting a minimum number of employees that are part of the firm and including firms that have been established for a set number of years or range of years (e.g., 5-10 years). This study also included firms across three different regional locations but was not able to make comparative cases across regions as a result of small sample sizes. Future researchers may want to conduct studies of Agile in one particular region or develop comparative cases that can have larger sample sizes.

An unforeseen and unavoidable limitation of this research is that data was collected prior to the COVID-19 pandemic. While the data collected pre-pandemic may not be reflective of the ways in which tech work may be organized in 2021 and beyond, this study offers an important look at how Agile impacted the software labor process before working remotely became a new and widespread feature of professional and digital work. As I discuss in the sections below, I believe that the COVID-19 pandemic presents important opportunities to analyze transformations in the software labor process, and worker responses to changes in their working conditions in the context of a global pandemic.

Future Research: COVID-19 and Transformations in the Labor Process

Taylored Flexibility for Remote Workers?

As a result of the COVID-19 pandemic, workers, especially in professional occupations, have found themselves required to work from home. While tech work is compatible with flexible and remote work, this dissertation has argued that tech companies adopting Agile are increasingly encouraging their workers to complete work on-site. This benefits firms in that they are able to improve surveillance over productivity, and they are able to more easily elicit commitment from team members, leading to workers ultimately being willing to put in extra, unexpected, unrecognized, and uncompensated labor. In the summer of 2020, about four months into the pandemic, I was able to attend a virtual Tech Alliance meeting, where workers discussed their companies' responses to COVID-19. In these informal conversations, many of the workers explained that their firms were hesitant to move them to fully remote work, making no statements about work arrangements even by late April. One of the workers at the meeting explained having to take collective action with other workers at their company, in which they shamed executives for requiring them to come to work during a health crisis and were finally able to work remotely.

Future research examining the impacts of Agile on the software labor process is needed in the context of the pandemic, in which remote work has become a necessity to keep workers safe. Research questions that may guide future studies in this area may include some of the following: How have companies that use the Agile project management methodology translated this method to remote work? What strategies do employers use to manage the production of software technology? What experiences have workers had transitioning from in-person to remote work? Have changes in the software labor process that resulted from COVID-19 occurred equitably? How do strategies and mechanisms to ensure productivity in the context of remote work compare to those used in the office? How do axes of gender, race, and age impact software workers' experiences of remote work?

Though research on the adoption of Agile in the remote setting would undoubtedly provide insights to managerial control methodologies, and ways to achieve Taylored Flexibility, in the context of the pandemic, as people in the United States get access to vaccinations and "business as usual" appears to make a return, research is also needed on the transition from remote work back to the office. The following research questions would be particularly relevant for studies conducted after workers who were temporarily working remotely return to work: For tech workers who primarily worked in the office before the pandemic, has the transition to remote work impacted whether companies have chosen to return to office-based work? How has remote work impacted the organization of tech work post-COVID-19? Do companies that used Agile before the transition to remote work still rely on this project management methodology? What, if any, new managerial strategies emerged? Overall, as the COVID-19 pandemic has undoubtedly transformed working conditions, research examining the intersections of Agile, managerial control, working arrangements, and equity in the context of a global health crisis are necessary to continue understanding the software labor process, and control over knowledge-based production more broadly.

The Exodus of Women

As the COVID-19 pandemic has presented challenges and concerns for workers, workers exiting the labor force all together has been a gendered process: “since February 2020, women have lost over 5.4 million net jobs, and account for 55.0% of overall net job loss since the start of the crisis” (Ewing-Nelson 2021). Not only have women, and particularly women of color, been more likely to be laid off or furloughed during the pandemic, but women with children have also disproportionately left the workforce as access to childcare remains largely inadequate. Future research aiming to identify gender and racial disparities in the impacts of COVID-19 on work in the tech industry should explore how parenting workers are managing new care work demands as a result of the pandemic. Researchers may want to explore what kinds of care work demands parenting software workers are facing, and how tech companies are responding to these new demands. It would be especially interesting to conduct comparative research on the experiences of fathers and mothers in tech during the pandemic, to see if strategies to address care work are gendered. Future research is needed to explore how childcare demands that have resulted from COVID-19, and employer responses to it, impact the “leaky pipeline” in tech, or the notion that women leave tech occupations because of organizational practices and policies.

Implications and Alternatives

Overall, through a reorganization of the labor process in a JIT manner, Agile suggests a need for firms to engage in both direct and normative forms of control over workers in order to achieve the most cost effective, flexible, and predictable software labor process. While this study answers questions about the organization of software work and the strategies for combining control and flexibility offered by project management methodologies, it also raises questions about the future of software development work and knowledge-intensive work more broadly. As Agile continues to mature, and with the software labor market growing at an average faster than all other occupations (BLS 2020), scholars will be pressed to examine the issue of skill saturation in software work. Moreover, this research calls for pause when framing software production and other forms of knowledge work as an especially unique form of labor, distinct from manufacturing-based work, as manufacturing production processes like lean have been successfully implemented as a tool to make enigmatic work like software more calculable.

Imagining policy and other solutions to the issues of control over knowledge work is no easy task. How do we regulate the ways that employers accumulate profit through surplus labor, especially in the context of knowledge work? One obvious transformation needed to redistribute how much the employer and the worker benefit from the productivity of the worker’s labor is through increasing wages. To be sure, tech workers earn comparatively high wages to the rest of the labor force. However, we should be careful to compare tech worker wages with low-wage workers, rather than comparing all worker wages to owner wealth. As Kinley, a Tech Alliance member put it:

> I used to feel guilty about how much money I make. But I'm coming to understand that it’s more that everybody else doesn’t make enough money than I

> make too much. I still don't have access to all the value that I produce. Although I'm in the upper-middle class in one sense, I still have a lot more in common with a warehouse worker than I do with Jeff Bezos.

As Kinley so astutely recognizes, improving the conditions of professional and high-wage workers requires a transformation in our approach to wages more broadly. This year we have seen that worker organizing for a $15 minimum wage, hardly an unreasonable demand given the rate of inflation, did not pass in the senate, leaving the federal minimum wage at $7.25 per hour. Raising the floor of wages overall is necessary before targeting wages in particular industries. Moreover, wages must be more appropriately determined by cost of living: in San Francisco, for example, though the average income for a tech worker was $155,000, the cost of living scored 269.3 above 100 in the U.S. overall, with the average cost of a one-bedroom apartment at $2,693 per month, or $32,316 per year.[36] Moreover, workers could gain more control over their labor with a transformation of intellectual property rights. Especially as business has been conducted virtually as a result of the pandemic, organizations have easy access to knowledge workers' digital labor. Transforming intellectual property agreements would be a significant step towards workers getting more rights to and control over their labor.

As I have discussed throughout this dissertation, because software workers are also often hired on salaries, they have little recourse for receiving wages appropriate to all of the work they are doing. Some potential solutions to this issue include having clear and standardized job descriptions, at a minimum within firms, such that each worker knows exactly what wage they are being paid for the role they are doing, and what their particular job duties are expected in that role. If firms want to ensure that they can be

[36] https://www.bestplaces.net/cost_of_living/city/california/san_francisco

flexible such that workers are able to take on extra work, the firms should develop agreements in which the worker consents to the extra work, as the firm must also provide extra compensation for extra work. Though developing the paperwork or procedures necessary to track workers taking on extra work may be more bureaucratically cumbersome than that “flat” organization would prefer, the absence of these agreements results in workers taking on uncompensated work. Clear policies that indicate what kinds of work each team member needs to engage in in order to receive a raise would not only increase the equity of pay across workers but help to establish explicit understandings about what is expected from workers.

To be sure, the perspective taken in this dissertation is that the capitalist logic of accumulation necessarily requires the workers’ subjugation to capital. We must continue to engage in research to analyze whether capitalism is the most appropriate economic structure to support the development of innovative, socially responsible, and even transformative technologies, which, this dissertation has shown, it may not be. We know that as long as firms and clients have conceptual control over the kinds of software that is going to be produced and used by people in all levels of our society, then workers will continue to be required to build technologies that reproduce imperialism, white supremacy, sexism, and domination. Moreover, as labor scholars continue to produce research about new forms of work in the context of an increasingly globalized, neoliberal, class stratified, digital and knowledge-based economy, my hope is that this research inspires labor scholars to highlight continuities rather than prioritize differences in production processes used in manual and knowledge-based work, such that we are able to analyze similarities in worker experience and worker issues in order to develop more holistic responses, and to collectively imagine alternatives.

REFERENCES CITED

Acker, Joan. 1990. “Hierarchies, Jobs, Bodies: A Theory of Gendered Organizations.” *Gender & Society* 4(2):139–158.

Acker, Joan. 1992. “Is Capitalism Gendered and Racialized.” Pp. 125–33 in *Race, Class, and Gender: An Anthology*, edited by M. Andersen and P. H. Collins. Belmont, CA: Wadsworth Pub. Co.

Acker, Joan R. 2006. “Inequality Regimes: Gender, Class, and Race in Organizations.” *Gender & Society* 20(4):441–464.

Adler, Paul S. 2005. “The Evolving Object of Software Development.” *Organization* 12(3):401–435.

Ahmad, Aqueil. 1981. “Knowledge work: Its Evaluation and Reward.” *R&D Management* 11(2):69–78.

Akgün, Ali, Halit Keskin, A.Y. Cebecioglu, and Derya Dogan. 2015. “Antecedents and Consequences of Collective Empathy in Software Development Project Teams.” *Information & Management* 52:247-259.

Alvesson, Mats. 1993. “Organization as Rhetoric: Knowledge-Intensive Companies and the Struggle with Ambiguity.” *Journal of Management Studies* 30(6):997 - 1015

Alvesson, Mats, and Stanley Deetz. 2000. *Doing Critical Management Research*. London: Sage

Alvesson, Mats, Paul Thompson, and Stephen Ackroyd. 2004. “Post-bureaucracy?” Pp. 485-507 in *The Oxford Handbook of Work and Organization,* edited by Stephen Ackroyd, Rosemary Batt, Paul Thompson, and Pamela S. Tolbert. Oxford, United Kingdom: Oxford University Press https://doi.org/10.1093/oxfordhb/9780199299249.003.0025

Ambler, Scott W. 2003. “Generalizing Specialists: Improving Your IT Career Skills.” Retrieved Nov. 13, 2021 (http://www.agilemodeling.com/essays/generalizingSpecialists.htm).

Amrute, Sareeta. 2020. “Bored Techies Being Casually Racist: Race as Algorithm.” *Science, Technology, & Human Values* 45(5): 903-933.

Andre, Thomas, Myrna Whigham, Amy Hendrickson, and Sharon Chambers. 1999. “Competency Beliefs, Positive Affect, and Gender Stereotypes of Elementary Students and Their Parents About Science Versus Other School Subjects.” *Journal of Research in Science Teaching* 36:719-747.

Andrews, Chris K., Craig D. Lair, and Bart Landry. 2005. "The Labor Process in Software Startups: Production on a Virtual Assembly Line?" Pp. 40-67 in *Management, Labor Process and Software Development: Reality Bytes* edited by Rowena Barrett. Abingdon, Oxon: Routledge.

Aneesh, A. 2001. "Skill Saturation: Rationalization and Post-Industrial Work." *Theory and Society* 30(3):363-396.

Antikainen, Riikka, and Antti Lönnqvist. 2006. "Knowledge Work Productivity Assessment." *Institute of Industrial Management:*1-20.

Atkinson, Robert D., and R. Court. 1998. "The New Economy Index: Understanding America's Economic Transformation." Retrieved Nov 5, 2020 (www.neweconomyindex.org/index_nei.html).

Baldry, Chris, Dora Scholarios, and Jeff Hyman. 2005. "Organizational Commitment Among Software Developers" Pp. 149 – 172 in *Management, Labor Process and Software Development: Reality Bytes* edited by Rowena Barrett. Abingdon, Oxon: Routledge.

Balch, H. 1994. "Project as Elements of a New Industrial Pattern: A Division of Project Management" Pp. 2 – 11 in *Global Project Management Handbook* edited by David I. Cleland and Roland Gareis. New York: McGraw-Hill International Editions.

Barker, James R. 1993. "Tightening the Iron Cage: Concertive Control in Self-Managing Teams." *Administrative Science Quarterly* 38(3):408-437.

Barker, James R. and Phillip K. Tompkins. 1993. "Organizations, Team, Control and Identification." Unpublished manuscript, Department of Communication, University of Colorado, Boulder.

Barley, Stephen R. and Gideon Kunda. 1992. "Design and Devotion: Surges of Rational and Normative Ideologies of Control in Managerial Discourse" *Administrative Science Quarterly* 37(3):363-399.

Barrett, Nancy J. 1999. "The Struggles of Women Industrial Workers to Improve Work Conditions in the Progressive Era." *Organization of American Historians Magazine of History* 13(3): 43-49

Barrett, Rowena. 2001. "Laboring Under an Illusion? The Labor Process of Software Development in the Australian Information Industry." *New Technology Work and Employment* 16: 18 - 34

Barrett, Rowena. 2004. "Working at Webboys: An Analysis of Control Over the Software Development Labor Process." *Sociology* 38(4): 777 - 794

Barrett, Rowena. 2005. “Managing the Software Development Labor Process: Direct Control, Time and Technical Autonomy. Pp. 68-88 in *Management, Labor Process and Software Development: Reality Bytes* edited by Rowena Barrett. Abingdon, Oxon: Routledge.

Bashaw, R. Edward., and Stephen E. Grant. 1994. “Exploring the Distinctive Nature of Work Commitments: Their Relationships with Personal Characteristics, Job Performance, and Propensity to Leave.” *Journal of Personal Selling and Sales Management* 14:41-56.

Beirne, Martin, H. Ramsay, and A. Panteli. 1998. “Developments in Computing Work: Control and Contradiction in the Software Labor Process” Pp. 142-162 in *Workplaces of the Futured edited* by Paul Thompson and Chris Warhurst. Houndsmill: Macmillan Business.

Belassi, Walid, and Oya Tukel. 1996. “A New Framework for Determining Success/Failure Factors in Projects.” *International Journal of Project Management*. 14(3):141-51.

Bell, Daniel. 1973. *The Coming of Post-industrial Society: A Venture in Social Forecasting.* New York: Basic Books.

Beltrán-Martín, Inmaculada. 2006. “Influence of High Commitment Management on Organisational Performance: Human Resource Flexibility as a mediator Variable.” Phd dissertation, Department of Business Administration and Marketing, Universitat Jaume I.

Benner, Chris, and Amy Dean. 2000. “Labor in the New Economy: Lessons from Labor Organizing in Silicon Valley” in *Nonstandard Work: The Nature and Challenges of Changing Employment Arrangements* edited by Francoise J. Carre, Marianne A. Ferber, Lonnie Golden, and Stephen A. Herzenberg. Cornell University Press.

Bentley, T. 1990. “The Knowledge Workers.” *Management Accounting-London* 68(3): 47.

Blau, Peter Michael. 1955. *The Dynamics of Bureaucracy*. Chicago: University of Chicago Press.

Bolin, Malin, and Annika Härenstam, A. 2008. “An Empirical Study of Bureaucratic and Post-Bureaucratic Characteristics in 90 workplaces” *Economic and Industrial Democracy* 29(4):541–564.

Bonilla-Silva, Eduardo. 2018. *Racism without Racists: Color-Blind Racism and the Persistence of Racial Inequality in America*. Rowman & Littlefield

Braverman, Harry. 1974. [1998 25th anniversary edition] *Labor and Monopoly Capital: The Degradation of Work in the Twentieth Century*. New York: Monthly Review Press.

Briand Louise, and Guy Bellemare. 2006. "A Structurationist Analysis of Post-Bureaucracy in Modernity and Late Modernity." *Journal of Organizational Change Management* 19(1): 65 -79.

Briand, Louise, and Damien Hodgson. 2015. "The Re-Regulation of Control in the Context of Project-based Work." *International Journal of Work Innovation* 1(3): 287 - 304.

Britton, Dana M. 2000. "The Epistemology of the Gendered Organization." *Gender & Society*. 14(3): 418 – 434

Bronfenbrenner, Kate, Sheldon Friedman, Richard W. Hurd, Rudolph A. Oswald, Ronald L., and Seeber. 1998. *Organizing to Win: New Research on Union Strategies*. Cornell University Press

Brooks, Frederick. 1975. *Mythical Man Month: Essays on Software Engineering*. Massachusetts: Addison Wesley.

Brophy, Enda. 2006. "System Error: Labour Precarity and Collective Organizing at Microsoft." *Canadian Journal of Communication* 31(3): 619 – 638.

Brophy, Enda. 2008. *The Organizations of Immaterial Labour: Knowledge Worker Resistance in Post-Fordism*. Queen's University, Kingston, Ontario, Canada.

Brophy, Enda, and Greig de Peuter. 2007. "Immaterial Labor, Precarity, and Decomposition" *in Knowledge Workers in the Information Society* edited by Catherine McKercher and Vincent Mosco. Plymouth, United Kingdom: Lexington Books

Brophy, Jere. 1987. "Linking Knowledge Workers and Information Technology." *Office* 106(3): 88–92.

Browne, Irene, and Joya Misra. 2003. "The Intersection of Gender and Race in the Labor Market." *Annual Review of Sociology* 29:487 – 513

Burawoy, Michael. 1979. *Manufacturing Consent: Changes in the Labor Process under Monopoly Capitalism*. Chicago, IL: University of Chicago Press

Bureau of Labor Statistics. 2020 "Occupational Outlook Handbook: Software Developers." Accessed Dec 15, 2020. https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm

Bureau of Labor Statistics. 2020. “Labor Force Statistics from the Current Population Survey.” https://www.bls.gov/cps/cpsaat11.htm

Bureau of Labor Statistics. 2020. “Labor Force Statistics from the Current Population Survey.” https://www.bls.gov/cps/cpsaat18.htm

Bureau of Labor Statistics. 2021. “Union Members-2020.” https://www.bls.gov/news.release/pdf/union2.pdf

Cabrera, Alberto F., and Amaury Nora. 1994. “College Students’ Perceptions of Prejudice and Discrimination and Their Feelings of Alienation: A Construct Validation Approach1.” *Review of Education, Pedagogy, and Cultural Studies* 16(3-4)1994: 387–409. doi:10.1080/1071441940160310.

Cachia M, Millward L., 2011. “The Telephone Medium and Semi-Structured Interviews: A Complementary Fit. *Qualitative Research in Organizations and Management: An International Journal* 6(3):265–277.

Callaghan, George, and Paul Thompson. 2001. “Edwards Revisited: Technical Control and Call Centres.” *Economic and Industrial Democracy* 22(1): 13-37

Carr, Eloise C.J., and Allison Worth. 2001. “The Use of the Telephone Interview for Research. *Nursing Times Research* 6:511–524.

Caruso, Loris. 2016. “The ‘Knowledge-Based Economy’ and the Relationship Between the Economy and Society in Contemporary Capitalism.” *European Journal of Social Theory* 19(3): 409-430

Case, Peter and Erik Piñeiro. 2009. “Stop Whining, Start Doing! Identity Conflict in Project Managed Software Production.” *Ephemera* (2): 93-112.

Castells, Manuel. 1996. *The Rise of the Network Society: The Information Age: Economy, Society, and Culture, Volume I.* Oxford: Blackwell Publishers.

Cicmil, Svetlana, Monica Lindgren, and Johann Packendorff. 2016. “The Project (Management) Discourse and its Consequences: On Vulnerability and Sustainability in Project-based Work.” *New Technology, Work and Employment* 31(1): 58-76

Chakravartty, Paula. 2006. “Symbolic Analysts or Indentured Servants? Indian High-Tech Migrants in America’s Information Economy.” *Knowledge, Technology & Policy* 19:27-43

Choi, Thomas Y. and Varney, Glenn H. 1995. “Rethinking the Knowledge Workers: Where Have all the Workers Gone?” *Organizational Development Journal* 13:41–50.

Chun, Jennifer Jihye. 2016. "Building Political Agency and Movement Leadership: The Grassroots Organizing Model of Asian Immigrant Women Advocates." *Citizenship Studies* 20(3-4):379-395

Clawson, Dan. 1980. *Bureaucracy and the Labour Process: The Transformation of U.S. Industry, 1860-1920*. New York: Monthly Review Press.

Clegg, Stewart, and Courpasson David. 2004. "Political hybrids: Tocquevillean Views on Project Organizations." *Journal of Management Studies* 41(4): 0022 - 2380

Cobbs, Charles G. 2011. *Making Sense of Agile Project Management: Balancing Control and Agility*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Cockburn, Highsmith J. 2001. "Agile Software Development: The People Factor." *Computer* 34(11): 131 - 133

Collinson, David L. 1992. *Managing the Shopfloor: Subjectivity, Masculinity and Workplace Culture*. Berlin: Walter de Gruyter

Cohen, Philip N., and Matt L. Huffman. 2003. "Occupational Segregation and the Devaluation of Women's Work across U.S. Labor Markets." *Social Forces* 81(3): 881-908

Conn, H.P. 1984. "Improving Use of Discrepancy Time Raises Productivity of Knowledge Workers in Offices." *Industrial Engineering* 16(7):70-77.

Constantine, L.L. 1993. "Work Organization: Paradigms for Project Management and Organization." *Communication of the ACM* 36(10):30-43.

Corrington, Abby. 2020. "Perceptions of Team Contributions for Men and Women." Phd dissertation, Department of Psychology Sciences, Rice University.

Cressey, Peter, and John MacInnes. 1980. "Voting for Ford: Industrial Democracy and the Control of Labour." *Capital & Class* 4(2): 5-33.

Crowley, Martha, Julianne C. Payne, and Earl Kennedy. 2013. "Working Better Together? Empowerment, Panopticon and Conflict Approaches to Teamwork." *Economic and Industrial Democracy* 35(3): 483-506.

Cusumano, Michael A., and Stanley Smith. 1995. "Beyond the Waterfall: Software Development at Microsoft." *Working paper* (Sloan School of Management, Massachusetts Institute of Technology): 3844-95.

Davenport, Thomas H., Jarvenpaa, S.L., and Beers, M.C. 1996. "Improving Knowledge Work Processes." *Sloan Management Review* 37: 53–65.

Davenport, Thomas H., Robert J. Thomas, and Susan Cantrell. 2002. "The Mysterious Art and Science of Knowledge-Worker Performance." *MIT Sloan: Management Review* 44(1): 23-30.

Damarin, Amanda Kidd. 2006. "Rethinking Occupational Structure: The Case of Web Site Production Work." *Work and Occupations* 22:429-463.

Damarin, Amanda Kidd. 2013. "The Network-Organized Labor Process: Control and Autonomy in Web Production Work. *Research in the Sociology of Work* 24: 177 – 205.

Danford, Andy. 1998. "Work Organization Inside Japanese Firms in South Wales: A Break from Taylorism?" Pp. 142 – 162 in *Workplaces of the Future* edited by Paul Thompson and Chris Warhurst.

Daniels, Jessie. 2015. "'My Brain Database Doesn't See Skin Color': Color-Blind Racism in the Technology Industry and in Theorizing the Web." *American Behavioral Scientist* 59(11): 1377-1393.

Davies, Rhys, Robert McNabb, and Keith Whitfield. 2015. "Do High-Performance Work Practices Exacerbate or Mitigate the Gender Pay Gap?" *Cambridge Journal of Economics* 39(2): 537-564

Delbridge, Rick. 1995. "Surviving JIT: Control and Resistance in a Japanese Transplant." *Journal of Management Studies* 32(6): 803-817.

Deetz, Stanley. 1998. "Discursive Formations, Strategised, Subordination and Self-Surveillance." Pp. 151-172 in *Foucault, Management and Organisation Theory* edited by Alan McKinlay and Ken Starkey. London: Sage Publications.

Dickens, Linda. 1998. "What HRM Means for Gender Equality." *Human Resource Management Journal*. 8(1): 23-40.

Di Leonardo, Micaela. 1985. "Women's Work, Work Culture, and Consciousness." *Feminist Studies* 11(3): 491-495.

Dove, Rick. 1998. "The Knowledge Worker." Automotive Manufacturing and Production 110: 26–28

Drucker, Peter F. 1969. *The Age of Discontinuity: Guidelines to our Changing Society.* London: William Heinemann Ltd.

Drucker, Peter F. 1979. "Managing the Knowledge Worker." *Modern Office Procedures* 24:12–16.

Drucker, Peter F. 1999. "Knowledge-Worker Productivity: The Biggest Challenge. *California Management Review* 41:79–94.

Eagly, Alice H., & Johnson, Blair T. 1990. "Gender and Leadership Style: A Meta-Analysis. *Psychological Bulletin* 108(2): 233–256. https://doi.org/10.1037/0033-2909.108.2.233

Eagly, Alice H. 2007. "Female Leadership Advantage and Disadvantage: Resolving the Contradictions." *Psychology of Women Quarterly*31(1)1–12.doi:10.1111/j.1471-6402.2007.00326.x.

Edmonds, E.A. 1974. "A Process for the Development of Software for Non-Technical Users as an Adaptive System" *General Systems* 19: 215 – 218.

Edwards, Richard. 1979. *Contested Terrain*. New York: Basic Books.

Edwards, P.K. 1990. "Understanding Conflict in the Labor Process: The Logic and Autonomy of Struggle" Pp. 125-152 in *Labour Process Theory: Studies in the Labour Process* edited by David Knights and Hugh Willmott. London: Macmillan Press.

Ekstedt, Eskil, Rolf A. Lundin, Anders Söderholm, and Hans Wirdenius. 1999. *Neo-Industrial Organising: Renewal by Action and Knowledge Formation in a Project-Intensive Economy*. London: Routledge.

Ellway, Benjamin P.W. 2013. "Making it Personal in a Call Centre: Electronic Peer Surveillance." *New Technology, Work and Employmen*t 28(1) 37-50.

Esterberg, Kristin G. 2002. *Qualitative Methods in Social Research*. McGraw-Hill

Ewing-Nelson, Claire. 2021. "All of the Jobs Lost in December Were Women's Jobs." National Women's Law Center.

Ezzamel, Mahmoud, and Hugh Willmott. 1998. "Accounting for Teamwork: A Critical Study of Group-Based Systems of Organisational Control." *Administrative Science Quarterly* 43(2):358–396.

Ezzamel, Mahmoud, Hugh Willmott, and Frank Worthington. 2001. "Power, Control and Resistance in 'The Factory that Time Forgot.'" *Journal of Management Studies* 38(8): 1053-1079

Fields, Diane, and Tom Bramble. 1992. "Post-Fordism: Historical Break or Utopian Fantasy?" *Journal of Industrial Relations* 34(4): 562-579.

Fine, Janice. 2015. "Alternative Labor Protection Movements in the United State: Reshaping Industrial Relations?" *International Labour Review* 154(1): 15-26.

Fitzgerald, Mark. 1999. "Guild targets Microsoft." *Editor and Publisher* 132(5): 31 – 34.

Fitzgerald, Brian, Kieran Conboy, Ken Power, Ricardo Valerdi, Lorraine Morgan, and Klass-Jan Stol. 2013. *Lean Enterprise Software and Systems*. 4th International Conference, LESS

Fleming, Peter and Andrew Sturdy. 2009. "'Just Be Yourself!' Towards Neo-normative Control in Organizations?" *Employee Relations 3*1(6): 1-24.

Fleming, Peter, and Andrew Sturdy. 2010. "'Being Yourself' in the Electronic Sweatshop: New Forms of Normative Control." *Human Relations* 64(2): 177–200. doi:10.1177/0018726710375481.

Florida, Richard. 2002. *The Rise of the Creative Class: And How It's Transforming Work, Leisure, Community and Everyday Life*. New York: Basic Books

Fox, Marilyn L. 1990. "Arming Knowledge Workers with Computer Systems." *Production and Inventory Management Review and* APICS News 10:50–54.

Frenkel, Steve, Marek Korczynski, Leigh Donoghue, and Karen Shire. 1995. "Re-constituting Work: Trends Towards Knowledge Work and Info-Normative Control." *Work, Employment and Society* 9(4): 773-796.

Friedman, Andrew L. 1977. *Industry and Labor: Class Struggle at Work and Monopoly Capitalism*. London: The Macmillan Press Ltd.

Friedman, Andrew L. 1984. Management Strategies: Market Conditions and the Labor Process. Pp. 176-215 in *Firms Organisation and Labor* Frank H. Stephen.

Friedman, Andrew L. 1985. "Managerial Strategies, Activities, Techniques and Technology: Towards a Complex Theory of the Labour Process." Paper presented to the Interdisciplinary Perspectives on Accounting Conference, University of Manchester.

Fries-Britt, Sharon, and Bridget Turner. 2002. "Uneven Stories: Successful Black Collegians at a Black and a White Campus." *The Review of Higher Education* 25(3):315–330. doi:10.1353/rhe.2002.0012.

Fujimoto, Takahiro. 1999. *The Evolution of a Manufacturing System at Toyota*. Oxford: Oxford University Press.

Gephart, Robert P. Jr. 2002. "Introduction to the Brave New Workplace: Organizational Behavior in the Electronic Age." *Journal of Organizational Behavior* 23: 327-344.

Glass, Robert. L., Iris Vessey, and Sue A. Conger. 1992. “Software Tasks.” *Information and Management* 23(4):183-91.

Gleadle, Pauline., Damian. Hodgson and John Storey. 2012. “The Ground Beneath My Feet: Projects, Project Management and the Intensified Control of R&D.” *New Technology, Work and Employment* 27 (3): 163 – 177.

Gottfried, Heidi. 1991. “Mechanisms of Control in the Temporary Help Service ‘ Industry.” *Sociological Forum* 6(4): 699-713.

Greenbaum, Joan. 1999. “On Twenty-Five Years with Braverman’s Labor and Monopoly Capital (or, How Did Control and Coordination of Labor Get into the Software So Quickly?).” *Monthly Review* 50(8): 28 – 32.

Hackett, Rick D., Bycio, Peter, & Hausdorf, Peter A. 1994. “Further Assessment of Meyer and Allen’s (1991) Three-component Model of Organizational Commitment.” *Journal of Applied Psychology* 79:15-23.

.

Hamilton, Angus. 2001. *Managing Projects for Success: A Trilogy*. Thomas Telford Publishing: Quay: London.

Han, Byung-Chul. 2017. *Psychopolitics: Neoliberalism and the New Technologies of Power*. London: Verso.

Harvey, David. 1990. *The Condition of Postmodernity: An Inquiry into the Origins of Cultural Change*. Cambridge: Blackwell.

Harvey, David. 2005. *A Brief History of Neoliberalism*. Oxford University Press: Oxford.

Head, Simon. 2003. *The New Ruthless Economy: Work and Power in the Digital Age.* Oxford: Century Foundation/Oxford University Press.

Heckscher, Charles. 1994. “Defining the Post-Bureaucratic Type. Pp. 14 – 62 *The Post-Bureaucratic Organization: New Perspectives and Organizational Change* edited by Charles Heckscher and Anne Donnellon. Thousand Oaks, CA: Sage.

Helton, B Ray. 1988. “The ‘Best Work’ Method of Knowledge Worker Assessment.” Industrial Management 29:26–29.

Hewlett Packard Enterprise. 2015. “Agile is the New Normal: Adopting Agile Project Management.” Available at: https://softwaretestinggenius.com/docs/4aa5-7619.pdf

Hobday, Mike. 1998. “Product Complexity, Innovation and Industrial Organisation.” *Research Policy* 26: 689 – 710.

Hobday, Mike. 2000. “The Project-Based Organization: An Ideal form for Managing Complex Products and Systems?” *Research Policy* 29(7-8): 871-893.

Hodgson, Damian. 2002. “Disciplining the Professional: The Case of Project Management” *Journal of Management Studies* 39(6): 803 – 821.

Hodgson, Damian. 2004. “Project Work the Legacy of Bureaucratic Control in the Post-Bureaucratic Organization.” *Organization* 11(1): 81 – 100.

Hodgson, Damian, and Svetlana Cicmil (eds). 2006. “Making projects critical: an introduction” in *Making Projects Critical*. New York: Palgrave.

Hodgson, Damian, and Louise Briand. 2013. “Controlling the Uncontrollable: ‘Agile’ Teams and Illusions of Autonomy in Creative Work.” *Work, Employment and Society* 27(2): 308 – 325.

Hodgson, Damien E., Monica Lindgren, Johann Packendorff, and Svetlana Cicmil. 2016. “The Politics of Projects in Technology-Intensive Work.” *New Technology, Work and Employment* 31(3): 1 – 3.

Hodson, Randy. 1995. “Worker Resistance: An Underdeveloped Concept in the Sociology of Work.” *Economic and Industrial Democracy* 16(1): 79 – 110.

Hoeller, Kieth. 2014. *Equality for Contingent Faculty: Overcoming the Two-Tier System*. Nashville, Tennessee: Vanderbilt University Press.

Holt, Amanda. 2010. “Using the Telephone for Narrative Interviewing: A Research Note.” *Qualitative Research* 10(1):113–121. doi:10.1177/1468794109348686.

Holweg, Matthias. 2007. “The Genealogy of Lean Production.” J*ournal of Operations Management* 25:420 – 437.

Huffman, Matt L., and Philip N. Cohen. 2004. “Racial Wage Inequality: Job Segregation and Devaluation across U.S. Labor Markets.” *American Journal of Sociology* 109(4):902-936.

Huws, Ursula. 1999. “Material World: The Myth of the Weightless Economy.*” Socialist Register* 35:29-55.

Huws, Ursula. 2003. *The Making of a Cybertariat: Virtual Work in a Real World*. Monthly Review Press.

Hyman, Richard. 1987. “Strategy or Structure: Capital, Labour and Control.” *Work, Employment and Society* 1(1):25–55.

Ilavarasan, P. Vigneswara. 2007. "Is Software Work in India Task Fragmented? A Study of Software Workers in Bangalore." *Sociological Bulletin* 56(3):127-144.

Irvine Annie, Paul Drew, and Roy Sainsbury. 2013. "'Am I Not Answering Your Questions Properly?' Clarification, Adequacy and Responsiveness in Semi-Structured Telephone and Face-to-Face Interviews." *Qualitative Research* 13(1):87–106.

Janz, Brian D., Jason A. Colquitt, and Roymond A. Noe. 1997. "Knowledge Worker Team Effectiveness: The Role of Autonomy, Interdependence, Team Development and Contextual Support Variables. *Personnel Psychology* 50:877–904.

Johnson, Phil, Geoffrey Wood, Chris Brewster, and Michael Brookes. 2009. "The Rise of Post-Bureaucracy. Theorists' Fancy or Organizational Praxis." *International Sociology* 24(1):37–61.

Jost, Micah, and Prieb Stoltzfus. 2011. "Independent Contractors, Employees, and Entrepreneurialism under the National Labor Relations Act: A Worker-by-Worker Approach" *The Washington and Lee Law Alumni Association Student Notes Colloquium.* Pp 311 – 352.

Kahle, Jane Butler and Marsha K. Lakes. 1983. "The Myth of Equality in Science Classrooms." *Journal of Research in Science Teaching* 20:131–140.

Kakar, Adarsh Kumar Satindarlal. 2017. "Engendering Cohesive Software Development Teams: Should we Focus on Interdependence or Autonomy?" *International Journal of Human-Computer Studies* 111:1-11.

Kalev, Alexandra. 2009. "Cracking the Glass Cages? Restructuring and Ascriptive Inequality at Work." *American Journal of Sociology* 114(6):1591-1643.

Kalleberg, Arne L. 2009. "Precarious Work, Insecure Workers: Employment Relations in Transition." *American Sociological Review* 74:1 – 22.

Kalleberg, Arne L. 2013. *Good Jobs, Bad Jobs: the Rise of Polarized and Precarious Employment Systems in the United States, 1970s to 2000s*. Russell Sage Foundation

Kelloway, E. Kelvin, and Julian Barling. 2000. "Knowledge Work as Organizational Behavior." *International Journal of Management Reviews* 2(3): 287-304.

King, Martha W. 2014. "Protecting and Representing Workers in the New Gig Economy: The Case of the Freelancers Union" in *New Labor in New York: Precarious Workers and the Future of the Labor Movement* edited by Ruth Milkman and Ed Ott. Cornell University Press.

Kinnie, Nick, Sue Hutchinson, and John Purcell. 2000. '"Fun and Surveillance": The Paradox of High Commitment Management in Call Centres." *International Journal of Human Resource Management* 11(5):967-985.

Klein, Janice A. 1989. "The Human Costs of Manufacturing Reform." *Harvard Business Review*: 60-66.

Kraft, Phillip. 1977. *Programmers and Managers: The Routinization of Computer Programming in the United States.* New York: Springer-Verlag.

Kraft, Phillip. 1979. "The Industrialization of Computer Programming: From Programming to 'Software Production.'" Pp. 1 – 17 in *Case studies on the Labor Process* edited by Andrew Zimbalist *Case studies on the Labor Process*. London: Monthly Review Press.

Kunda, Gideon. 1992. *Engineering Culture: Control and Commitment in a High-Tech Corporation.* Philadelphia, PA: Temple University Press.

Knights, David, and Hugh Willmott. 1990. *Labour Process Theory: Studies in the Labour Process*. London: Macmillan Press.

Lamphere, Louise. 1985. "Bringing the Family to Work: Women's Culture on the Shop Floor." *Feminist Studies* 11(3): 519-540.

Lechuga, Vincente M. 2012. "Exploring Culture from a Distance: The Utility of Telephone Interviews in Qualitative Research. *International Journal of Qualitative Studies in Education (QSE)* 25(3):251–268.

Liker, Jeffrey K. 2004. *The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer*. New York: McGraw-Hill.

Lindahl, Monica. 2006. "Engineering Improvisation: The Case of Wärtsilä." In *Aesthetic Leadership: Managing Fields of Flow in Art and Business* by Pierre Guillet de Monthoux, Claes Gustafsson, and Sven-Erik Sjöstrand. Palgrave Macmillan

Lindgren, Monica, and Johann Packendorff. 2006."What's New in New Forms of Organizing? On the Construction of Gender in Project-Based Work." *Journal of Management Studies* 43(4):841–866., doi:10.1111/j.1467-6486.2006.00613.x.

Lloyd, Tom., and Karl Erik Sveiby. 1989. *Managing Knowhow*. London: Bloomsbury.

Locke, Richard, Thomas Kochan, and Michael Piore (eds.). 1995. *Employment Relations in a Changing World Economy*. Cambridge (USA): MIT-Press.

Lopez, Steven H. 2004. *Reorganizing the Rustbelt: An Inside Study of the American Labor Movement*. Berkeley: University of California Press.

Lövstål, Eva, and Anne-Marie Jontoft. 2017. “Tensions at the Intersection of Management Control and Innovation: A Literature Review.” *Journal of Management Control*. 28(1):41–79., doi:10.1007/s00187-016-0244-3.

Maravelias, Christian. 2003. “Post-Bureaucracy: Control through Professional Freedom.” Journal of *Organizational Change Management* 16(5):547-566.

Marks Abigail, and Cliff Lockyer. 2005. “Professional Identity in Software Work: Evidence from Scotland” Pp. 129-148 in *Management, Labor Process and Software Development: Reality Bytes* edited by Rowena Barrett. Abingdon, Oxon: Routledge.

Marks, Abigail, and Dora Scholarios. 2007. “Revisiting Technical Workers: Professional and Organisational Identities in the Software Industry.” *New Technology Work and Employment* 22(2): 98-117.

Martin, James E., and Melanie M. Peterson. 2017. “Two-Tier Wage Structures: Implications for Equity Theory.” *Academy of Management Journal* 30(2). https://doi.org/10.5465/256275

Marx, Karl. 1990 [1976]. *Capital: A Critique of Political Economy*, vol. 1, translated by Ben Fowkes. New York: Penguin Books

Massey, Doreen. 1995. “Masculinity, Dualisms and High Technology.” *Transactions of the Institute of British Geographers* 20:487-499.

McDowell, Linda. 1998. Elites in the City of London: Some Methodological Considerations. *Environment and Planning A* 30: 2133-46.

McKercher, Catherine, and Vincent Mosco. 2008. *Knowledge Workers in the Information Society.* Plymouth, United Kingdom: Lexington Books

Merton, Robert K. 1949. *Social Theory and Social Structure*. Illinois: Free Press.

Metcalfe, Beverly. 1997. “Project Management System Design: A Social and Organisational Analysis.” *International Journal of Production Economics* 52(3): 305–316. doi:10.1016/s0925-5273(97)00091-1.

Michel, Zoe Ziliak, and Liz Ben-Ishai. 2016. “Good Jobs for All: Race Inequities in Job Quality.” CLASP:1 - 14.

Milkman, Ruth, and Ed Ott. 2014. *New Labor in New York: Precarious Workers and the Future of the Labor movement*. Cornell University Press.

Milton, Laurie P. 2003. “An Identity Perspective on the Propensity of High-Tech Talent to Unionize.” *Journal of Labor Research* 24(1): 31 – 53.

Morien Roy. 2005. Agile Management and the Toyota Way for Software Project Management. *2005 3rd IEEE International Conference on Industrial Informatics* (INDIN).

Moody, Kim. 1988. *An Injury to All: The Decline of American Unionism*. New York: Verso

Moody, Kim. 1997. *Workers in a Lean World*. New York: Verso..

Morris, Peter W.G. and Jeffrey K. Pinto (eds). 2004. *The Wiley Guide to Managing Projects*. New York: Wiley.

Nishikawa, Makiko. 2010. "(Re)Defining Care Workers as Knowledge Workers." *Gender, Work & Organization* 18(1):113–136. doi:10.1111/j.1468-0432.2010.00537.

Nkomo, Stella M. 1992. "The Emperor Has No Clothes: Rewriting Race in Organizations." *The Academy of Management* 17(3): 487 – 513.

Nomikos, George E. 1989. "Managing Knowledge Workers for Productivity." *National Productivity Review* 8(2): 165+

Novick, Gina. 2008. "Is There a Bias against Telephone Interviews in Qualitative Research?" *Research in Nursing & Health* 31(4):391–398. doi:10.1002/nur.20259.

Ohno, Taiichi. 1988. *Toyota Production System: Beyond Large Scale Production*. New York: Productivity Press.

Orlikowski, Wanda J. 1988. "The Data Processing Occupation: Professionalization Proletarianization?" *Research in the Sociology of Work* 4:95–124.

Packendorff, Johann. 1995. "Inquiring into the Temporary Organization: New Directions for Project Management Research." *Scandinavian Journal of Management* 11(4): 319–33.

Padavic, Irene. 1991. "The Re-Creation of Gender in a Male Workplace." *Symbolic Interaction* 14(3): 279-294.

Pellow, David Naguib, and Lisa Sun-hee Park. 2002. *The Silicon Valley of Dreams: Environmental Injustice, Immigrant Workers, and the High-Tech Global Economy.* New York: New York University Press.

Perlow, Leslie. 1997. *Finding Time: How Corporations, Individuals, and Families Can Benefit from New Work Practices*. Ithaca, NY: Cornell University Press.

Piore, Michael, and Sabel Charles. 1984. *The Second Industrial Divide*. New York: Basic Books.

Plumwood, Val. 1993. *Feminism and the Mastery of Nature*. New York: Routledge.

Plunkert, Lois M. 1998. "The 1980's: a Decade of Job Growth and Industry Shifts." *Monthly Labor Review*: 1-16.

Prusak, Laurence. 1997. *Knowledge in Organizations*. Butterworth-Heinemann.

Poppendeick Mary and Poppendieck Tom. 2003. *Lean Software Development: An Agile Toolkit.* Addison-Wesley.

Poppendieck Mary, and Cusumano Michael A. 2012. "Lean software Development: A Tutorial." *IEEE Software,* 29(5): 26 – 32.

Powell, Walter W., and Kaisa Snellman. 2004. "The Knowledge Economy." *Annual Review of Sociology* 30: 199-220.

Raelin, Joseph A. 2011. "The End of Managerial Control?" *Group and Organization Management* 36(2): 135 – 160.

Rasmussen Bente, and Johansen Birgitte. 2005. "Trick or Treat? Autonomy as Control in Knowledge Work." Pp. 89-108 in *Management, Labor Process and Software Development: Reality Bytes* edited by Rowena Barrett. Abingdon, Oxon: Routledge.

Reed, Michael I. 1996. "Expert Power and Control in Late Modernity: An Empirical Review and Theoretical Synthesis." *Organization Studies* 17(4):573–597.

Reich, Robert. 1991. *The Work of Nations: Preparing Ourselves for 21st-Century Capitalism*. London: Simon and Schuster.

Reitman, Meredith. 2006. "Uncovering the White Place: Whitewashing at Work." *Social and Cultural Geography.* 7(2):267 – 282.

Reus, Taco H., and Yongmei Liu. 2004. "Rhyme and Reason: Emotional Capability and the Performance of Knowledge-Intensive Work Groups." *Human Performance* 7(2): 245-66.

Robinson, Cedric J. 1983. *Black Marxism: The Making of the Black Radical Tradition*. North Carolina: University of North Carolina Press.

Rodino-Colocino, Michelle. 2007. “High-tech Workers of the World, Unionize! A Case Study of WashTech’s “New Model of Unionism” *in Knowledge workers in the information society* edited by Catherine McKercher and Vincent Mosco. Plymouth, United Kingdom: Lexington Books.

Rosenfeld, David. 2006. “Worker Centers: Emerging Labor Organizations—Until They Confront the National Labor Relations Act.” *Berkeley Journal of Employment and Labor Law* 27(2):469-513.

Román-González, Marcos, Juan-Carlos Pèrez-González, and Carmen Jimènez-Fernádez. 2017. “Which Cognitive Abilities Underlie Computational Thinking? Criterion Validity of the Computational Thinking Test.” *Computers in Human Behavior* (72): 678-691.

Rubin, Herbert J., and Irene S. Rubin. 1995. *Qualitative Interviewing: The Art of Hearing Data*. New York: Sage.

Silver, Beverly. 2003. *Forces of Labor*. New York, NY: Cambridge University Press

Quah, Danny T. 1997. “Increasingly Weightless Economy.” *Bank of England Quarterly Bulletin*, 49.

Quinney, Loretto, Trudy Dwyer, and Ysanne Chapman. 2016. “Who, Where, and How of Interviewing Peers.” *SAGE Open* 6(3). doi:10.1177/2158244016659688.

Samset, Knut, and Gro Holst Volden. 2016. “Front-End Definition of Projects: Ten Paradoxes and Some Reflections Regarding Project Management and Project Governance.” *International Journal of Project Management* 34(2): 297-313.

Scarborough, Harry. 1999. “Knowledge as Work: Conflicts in the Management of Knowledge Workers.” *Technology Analysis and Strategic Management* 11(1): 5-16.

Schiller, Herbert. 1973. *The Mind Managers.* Boston: Beacon Press.

Sewell, Graham, and Barry Wilkinson. 1992. ‘“Someone to Watch Over Me”: Surveillance, Discipline and the Just-in-Time Labour Process.” *Sociology* 26:271-289.

Smith, Vicki. 1995. “Braverman’s Legacy: The Labor Process Tradition at 20.” *Work and Occupations: An International Sociological Journal* 21(4):403–21.

Smith, Vicki. 1997. “New Forms of Work Organization.” *Annual Review of Sociology* 23: 315 – 339.

Srivastava, Prachi, and Nick Hopwood. 2009. "A Practical Iterative Framework for Qualitative Data Analysis." *International Journal of Qualitative Methods* 8(1):76–84. doi:10.1177/160940690900800107.

Stark, David. 1980. "Class Struggle and the Transformation of the Labor Process." *Theory and Society* 9(1): 89-130.

Stephens, Neil. 2007. Collecting Data from Elites and Ultra Elites: Telephone and Face-to-Face Interviews with Macroeconomists. *Qualitative Research* 7(2):203–216.

Strauss, Anselm, and Juliet Corbin. 1998. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage Publications.

Storey, John. 1985. "The Means of Management Control." *Sociology* 19(2):193–211.

Sturdy, Andrew, Peter Fleming, and Rick Delbridge. 2010. "Normative Control and Beyond in Contemporary Capitalism" in *Working Life: Renewing Labour Process Analysis* edited by Paul Thompson and Chris Smith Palgrave Macmillan.

Tampoe, Mahen. 1993. "Motivating Knowledge Workers – the Challenge for the 1990s." *Long Range Planning* 26(3): 49–55.

Tancred-Sheriff, Peta. 1989. "Gender, Sexuality, and the Labor Process" Pp. 45 – 55 in *The Sexuality of Organization* edited by Jeff Hearn, Deborah Sheppard, Peta Tancred-Sheriff, & Gibson. Burrell. London: Sage.

Taylor, Fredrick W. 1911. *The Principles of Scientific Management.* New York: Harper and Brothers Publishers

Thomas, Janice. 2006. "Problematising Project Management" in *Making Projects Critical* edited by Damien Hodgson and Svetlana Cicmil. New York: Palgrave

Thompson, Paul. 1998. "Crawling from the Wreckage: The Labour Process and the Politics of Freedom" Pp. 95-124 in *Labour Process Theory: Studies in the Labour Process* edited by David Knights and Hugh Willmott. London: Macmillan Press.

Thompson, Paul. 2003. "Disconnected Capitalism: Or Why Employers Can't Keep Their Side of the Bargain." *Work, Employment and Society* 17(2): 359 – 378.

Tschang, Ted F. 2007. "Balancing the Tensions Between Rationalization and Creativity in the Video Games Industry." *Organization Science* 18(6): 885 – 1027.

Toffler, Alvin. 1970. *Future Shock.* New York: Random House

Townsend, Keith. 2005. "Electronic Surveillance and Cohesive Teams: Room for Resistance in an Australian Call Centre*?" New Technology, Work and Employment* 20(1) 47 – 59.

Upadhya, Carol. 2009. "Controlling Offshore Knowledge Workers: Power and Agency in India's Software Outsourcing Industry." *New Technology, Work and Employment* 24(1):2-18.

Vallas, Steven P. 2003. "Why Teamwork Fails: Obstacles to Workplace Change in Four Manufacturing Plants." *American Sociological Review* 68(2):223-50.

Van der Vegt, Gerben S., Ben Emans, and Evert Van de Vliert. 1998. "Motivating Effects of Task and Outcome Interdependence in Work Teams." *Group Organization Management* 23:12-155.

Van Jaarsveld, Danielle D. 2004. "Collective Representation Among High-Tech Workers at Microsoft and Beyond: Lessons from WashTech/CWA. *Industrial Relations* 43(2): 364 – 385.

Vidal, Matt. 2006. "Manufacturing Empowerment? 'Employee involvement' in the Labor Process after Fordism." *Socio-Economic Review* 5(2):197-232.

Vidal, Matt. 2017. "Lean Enough: Institutional Logics of Best Practice and Managerial Satisficing in American Manufacturing." *Socius* 3:1-17.

Voss, Kim, and Rachel Sherman. 2000. "Breaking the Iron Law of Oligarchy: Union Revitalization in the American Labor Movement." *American Journal of Sociology* 106(2):303 – 349.

Wajcman, Judy. 1998. *Managing like a Man: Women and Men in Corporate Management*. Cambridge: Polity Press.

Walby, Sylvia. 2010. "Is the Knowledge Society Gendered?" *Gender, Work & Organization* 18(1):1-29. doi:10.1111/j.1468-0432.2010.00532.x.

Walton, Richard. 1987. "From Control to Commitment in the Workplace." *Harvard Business Review* 63:76-84.

Williams, Christine L., Chandra Muller, and Kristine Kilanski. 2012. "Gendered Organizations in the New Economy." *Gender & Society* 26(4): 549-573.

Williams, Terry. 1995. "A Classified Bibliography of Research Relating to Project Risk Management." *European Journal of Operational Research* 85:180-8.

Willmott, Hugh. 1993. "Strength is Ignorance; Slavery is Freedom: Managing Culture in Modern Organizations." *Journal of Management Studies* 30(4):0022-2380.

Wingfield, Adia Harvey. 2019. *Flatlining: Race, Work, and Health Care in the New Economy*. Oakland, CA: University of California Press.

Whitley, Richard. 2006. "Project-Based Firms: New Organizational Form or Variations on a Theme?" *Industrial and Corporate Chang*e 15(1): 77-99.

Womack James P, Jones Daniel T, and Roos Daniel. 1990. *The Machine That Changed the World: The Story of Lean Production*. New York, NY: Free Press.

Wood, Stephen. 1982. *The Degradation of Work?: Skill, Deskilling, and the Labour Process*. London: Hutchinson.

Wood, Stephen, and Lilian De Menezes. 1998. "High Commitment Management in the UK: Evidence from the Workplace Industrial Relations Survey and Employers: Manpower and Skill Practices Survey." *Human Relations* 51: 485-515.

Worstall, Tim. 2012. "The Story of Henry Ford's $5 a Day Wages: It's Not What You Think." Forbes. Accessed April 23, 2021: https://www.forbes.com/sites/timworstall/2012/03/04/the-story-of-henry-fords-5-a-day-wages-its-not-what-you-think/?sh=240517fc766d

Wuthnow, Robert, and Wesley Shrum. 1983. "Knowledge Workers as a 'New Class': Structure and Ideological Convergence among Professional-Technical Workers and Managers." *Work and Occupations* 10(4): 471 – 487.

Youndt, Mark A., Scott A. Snell, James W. Dean, and David P. Lepak. 1996. "Human Resource Management, Manufacturing Strategy and Firm Performance." *Academy of Management Journal* 39(4): 836-66.

Zika-Viktorsson, Annika, Per Sundstrom, and Mats Engwall. 2006. "Project Overload: An Exploratory Study of Work and Management in Multi-project Settings." *International Journal of Project Management* 24(5): 385-394.