Inside or Outside the Struggle?

Exploring Political Possibilities for Freelance Developers

by

Katy Crane

Submitted in partial fulfilment of the requirements
for the degree of Master of Arts

at

Dalhousie University
Halifax, Nova Scotia
May 2020

Table of Contents

List of Tables

Abstract

This thesis examines the everyday work time and work space experiences of freelance web and software developers and programmers. It asks how freelance experiences inform collective efforts to overcome the precariousness entailed in contract tech work. Ten developers participated in this qualitative study, completing two interviews each, while collectively providing 28 reflection entries on their "ideal day" and eight photojournals documenting their daily work spaces. Combining ethnographic approaches with a precarity conceptual framework, this study found that freelancers' everyday behaviours offer insights into how, notwithstanding their apparent independence and autonomy, such workers engage with each other. Finding and using work spaces in Halifax, networking, and achieving an idealized "9-to-5" work day are all strategies that may contain political possibilities for concerted efforts against precarious working conditions.

List of Abbreviations Used

API	Application programming interface
CMS	Content management system
COVID-19	Coronavirus disease 2019
CV	*curriculum vitae*
GNP	Gross national product
HR	Human resources
ICT	Internet and computer technologies
IT	Internet technologies
MA	Master of Arts
MWA	Mobile Workers Assocation
NSCC	Nova Scotia Community College
PhD	Doctor of Philosophy
R&D	Research and development
RAM	Random-access memory
TCPS	Tri-Council Policy Statement

Acknowledgements

I wish to express deep gratitude to my advisor, Dr. Lindsay DuBois. She has all of the qualities of a great advisor: She never hurried or rushed me, never caused me stress, never had me wait long for feedback, and always helped me express my ideas better than I could. Going to her office was always enjoyable, never fraught. She always “got it”.

For the members of my thesis committee, Dr. Martha Radice and Dr. Karen Foster, I would like to express my profound appreciation for their careful and thorough readings of the drafts and for their detailed feedback. They helped me improve the thesis to a great degree, despite abrupt changes to our collective and personal lives due to pandemic.

I would also like to recognize the invaluable assistance and support that the faculty and staff of the department of Sociology and Social Anthropology provided to me throughout the duration of the program, specifically Dr. Fiona Martin, Dr. Brian Noble, Rachelle Fox and Leola Lefebvre. I offer a heartfelt thanks, too, to the staff at the Centre for Learning and Teaching, who not only gave me a part-time job, but loads of friendship and support.

I would like to extend my gratitude to friends and fellow parishioners at St. Paul’s Anglican Church, Halifax, who supported me throughout the past 18 months, specifically: Ann Denny, Adriel and Daniel Driver, Fr. Paul Friesen, Dawn Henwood and Jack Kivik. To my husband, Matt Townsend: I could not have done this without you.

My cohort was a very special group of people, and I thank everyone for their support and offer gratitude for the comraderie of our group.

Lastly, I wish to express my deepest thanks to all of the web and software developers who participated in this study. I am so deeply grateful to have met all of

you, and I'm in awe of how much time you put into this project. I hope to meet you all soon for a coffee.

Chapter 1: Introduction

Freelance, creative technological workers — such as web and software developers — present a conundrum to labour activists and critical scholars of contemporary labour experience and struggle. Depending with whom you are talking, this growing group is either essential to, peripheral to, or outside of the aims of the labour movement. Are they free from the typical constraints and controls placed on workers in the employee-employer relationship? Or does freelance work display an exaggerated experience of those constraints and controls? And further, how might the freelance experience—whether it is felt as freedom, constraint or other—be the basis for political articulation and organization?

Social science scholars wonder what kinds and compositions of labour and social movements are on the rise to meet contemporary challenges stemming from neoliberal policies across North America and Europe. The burdens associated with finding, maintaining—even creating—employment often now fall on the individual. Fewer workers congregate in the same places at the same times to do the same things, in evident contrast to the conditions that led to the widespread worker movements of the late 19th and early 20th centuries. The massing of workers around machines, common intervals of shift work and the social and physical proximities of life in working-class towns were the temporal, material and spatial conditions from which mass worker movements emerged. Their efforts resulted in, for example, the eight-hour work day, occupational safety laws and social safety net policies (Muehlebach, 2017). How does a growing sector of contemporary workers—web workers, gig economy workers, workers who hold several part-time jobs—organize to fight for its own interests when the historical conditions that led to success against exploitative practices in the recent past are disappearing or, for many groups of workers, have already disappeared? My research addresses this question by asking how members of one subset of the new gig workers, freelance web and software developers, understand and organize their working lives both temporally and spatially. We need to understand these processes before we can begin to comprehend the effects of such changes on political beliefs and practices.

Freelance web and software developers use their technical skills to build web sites or software for clients. They make individual choices about working hours, pricing models, sales strategies and work spaces, among a slew of other activities for which there is no boss or manager to tell them what to do. Freelance working life epitomizes labour shaped by the advent of the information, or post-industrial, society (Bell, 1973), network society (Castells, 2000[1996]) or alternatively, post-Fordism (Virno, 2004; Mäkinen, 2016). Scholars and activists agree that the latter transformations have led to: de-populated workplaces and disaggregated workforces due to internet and computer technologies (Castells, 2000[1996]; Kaoosji, 2018; Alberti, Bessa, Hardy, Trappman, & Umney, 2018); wages which are no longer protected by union rights nor indexed to productivity or inflation; and labour market flexibility with respect to work hours and contracts, to name just a few (Kalleberg, 2009). The growth of the creative industries (web and graphic design; software programming; marketing, branding and public relations; and research and development, for example) signals that contemporary capitalism places a premium on the cognitive, linguistic and creative capacities of workers (Berardi, 2005; Virno, 2004; Lazzarato, 1996). Creative work, furthermore, de-emphasizes shared experience and shared identities (Huws, 2014; McRobbie, 2014) leading many to argue that it precludes collective affect, such as solidarity—the animating force behind Ford-era worker movements.

The social science of work, an approach which grew up in age of industrialization, sometimes seems to struggle to adequately adapt to this new reality. It seems often to be constrained by assumptions which come out of the way work used to be organized.

Some scholarship sees an array of objective labour conditions—low wages, dissipating social welfare policies, part- and flex-time work and other exploitations—and charts a trajectory toward class-making from there (see Standing, 2011 as an example). Much social science attempts to discern how and why contemporary workers stray from the course of the ideal Fordist worker efforts—understandably so, as the growing set of objectively poor labour conditions appears, to scholars and activists alike, sufficient to jumpstart a newly energized global labour movement. Instead, recent social movements like the Fight for $15 and Occupy Wall Street shine brightly at times but struggle to achieve goals and longevity.

Discussing these organizing challenges facing freelance creative tech workers took my interview with Solomon, a software developer and a participant in this study, well past our allotted time (he did not mind). Over coffee at a favorite spot amongst Halifax's freelancers, we had just finished talking about the house of cards that is, in his view, the internet and the extreme conditions that can characterize the software industry. He commented on literary journalist Tracy Kidder's now classic *Soul of a New Machine*, a Pulitzer prize-winning nonfiction account of a team of computer engineers under enormous pressure to build a new 32-bit superminicomputer.

"Kidder seemed to enjoy working on the project, but the hours were crazy long; people were being exploited," he said, pausing to take a sip of coffee. I take advantage of the opening into worker exploitation to probe a little further. "This is part of my wondering behind this project," I begin to say. "I think the view of freelance tech is that their work is all fun and games; that they're entrepreneurial, not tied down, and don't care about the plight of the worker—they don't even consider themselves workers—and that they are, at the end of the day, un-organizable. But, as you were just describing, there are deep problems within this kind of work. What do you think is holding back freelance tech in organizing itself?"

He nods. “I’ve definitely thought about these things over the years. Organizing freelance workers is a bit like herding cats,” he laughs. “But, you look at Hollywood and the Writer’s Guild. Have you tried organizing writers?! They’re like cats! And they’re going on strike and stuff, and that’s gig work. It’s even a technology area, where camera technology is changing dramatically; but on the other hand, folks who work for special effects houses, which is very similar to doing videogame work—those tend to be sweatshops. Yeah, some of them get paid well, but once you factor in the hours…So, there’s starting to be unionization in the videogame industry.”

He continues: “Obviously as a freelancer, I have been able to push back, and I haven’t been super exploited or anything. When I had my two years in corporate America, the pay was great but we didn’t have a union, and people just came and went so fast. We couldn’t really get a guild going. Because stuff changes all the time, you couldn’t have a guild of skilled craftspeople. And, now, I’m doing all sorts of stuff: sometimes I’m doing art, sometimes I’m doing engineering, sometimes I’m doing web development...”

I interject: “You are not one thing all of the time.”

He quickly agrees. “I’m not one thing all of the time. I’m not one thing, so it’s harder to lock down…”

His speech trails off, as he tries to put his finger on all of the innumerable and ephemeral aspects of the problem, but, a quick speaker, he picks up again: “People trying to form a guild or a union I don’t even know what to say…but if I was in a workspace regularly, with a fairly structured set of things, I would definitely start thinking about it.”

This exchange indexed a number of realities for me, some only becoming clear at the very end of the research project. First, there is popular agreement about the necessary conditions for particular manifestations of effective worker organizing.

Again, Ford-era workers and their successful efforts dominate the imagination. Large unions running campaigns today, whatever may pass between executive strategists, still run campaigns toward this ideal. My time as a worker organizer revealed to me that institutional unions prioritize strikes, petitions and collective demand letters not only as preferential modes of attacking the other side, but as vehicles for the politicization of workers—or rather, the union assumed that these activities, and the numbers of them, said something real about how committed workers actually were to the vision of the campaign. If a group of workers exhibits certain behaviours, then they must be politicized; they must have class consciousness.

Second, this assumption about what worker action should look like leads to another point: that the persistence of the Ford-era imaginary obscures the ways that today's workers, like Solomon and others within the developer community, are making something out of their everyday, freelance realities in order to achieve some collective ends—the ways they are already engaging in political behaviors. Solomon, as an individual, engaged in political activities, as I learned over the course of the research period; but because his work life lacks a regular workspace, a stable employee roll, fixed occupational activities and other structures, as he said, the ways that freelancers come together and collaborate fail to register as political.

Discovering the unique, on-the-ground, everyday contexts that might inform and shape a different segment of contemporary labour—freelance web and software development—motivated this project and informed its research questions. The questions are founded on the proposition formulated by E.P. Thompson in his work *The Making of the English Working Class* (1963) that labouring classes make themselves out of their everyday, material reality; their everyday conditions and subjectivities are the clay from which class-making activity is formed. Similarly, among scholars attending to contemporary precarious labour and the aftermath of the EuroMayDay demonstrations of 2001, where *precarité* became a political buzzword, there is a recognition that today's workers articulates concerns that encompass much more than mere workplace conditions, wages and security; the *precarité* movement enunciated sites of struggle, rather, that originated in ecology, biology, temporality, or simply, the soul (Precarias a la Deriva, 2010; Papadopoulos, Stephenson, & Tsianos, 2008). Describing the political articulations of those participating in the EuroMayDay demonstrations of 2001, Dimitris Papadopoulos writes: "Precarity was considered simultaneously a new system of exploitation and a practice of liberation from the previous system of exploitation" (2008, p. 137). These political mobilizations and the scholarly work influenced by them imply that, more than mere objective labour conditions, what might point to the political possibilities of precarious labour are experiences of time, space, and everyday life in contemporary capitalism; experiences that may generate, and might become the strength of, new alternative ways of life.

Thus, I first wanted to know about freelance developers' experience of their working lives, and specifically those aspects that fundamentally create the experience of labour—work time and work space—and then how those experiences, and interpretations of them, feed into modes of being freelance, together. If the contemporary work experience looks less and less like the Ford factory with its masses of workers, synchronous time-shift work and typical modes of sociality, how do workers find—and recognize—one another in time, space and experience? If workers are disaggregated, strewn across the geographical landscape and heterogeneous in their working times, schedules and attitudes, do they find ways to overcome those obstacles? What are those ways—and what are the "hows" of those ways?

This thesis is organized in the following manner: the first chapter provides a general outline of the features of work now, providing context for understanding where the freelance web and software developers fit into the overall picture of precarious, alternative work. Next, I introduce the three composite characters who represent the collective narratives of the ten people who participated in this study. As this project did not, in my estimation, reach full ethnographic potential (I did not attend every public meeting or become a web developer myself), the three characters, apart from being the protagonists within a few vignettes scattered throughout the thesis, will act as narrative helpers as I lay out the main theoretical frameworks, findings and analyses of the data, which came from 20 interviews; 28 written, reflection entries; and 30 photos of workspaces (research design and methods are outlined in the appendices).

Chapter three documents and analyzes participants' experiences of work time, beginning with a literature review of relevant works addressing time and labour in anthropology, sociology and political economy. In order to unearth the objective and subjective conditions of freelance managing, tracking and other techniques of making time work for them within their economic relationships, I rely upon social anthropologist Laura Bear's unique contribution of employing Alfred Gell's (1992) time-map concept within a materialist understanding of capital and time from Marx (1992 [1985]). These frameworks allow the researcher to see how today's workers thread together various timelines to get their work done, but also how this threading is, in itself, an act of labour that benefits capital. From a direct application of Gell's time-maps, I discovered that a cultural map of the time-space landscape that developers understand and imagine to be the 9-to-5 office job obscures a political epistemology of temporal labour benefitting capital. Interestingly, while the 9-to-5 time-map obscures understanding of the inequalities embedded in creative work like web and software development, it is also an organizing principle for the social forms of freelance life.

Chapter four focuses on the space-specific experiences of freelance developers' working lives. I aimed, in part, to ask questions about work space that would help to nuance some assumptions about space, place and the placelessness of internet work that are frequently identified as impediments to collective activity. Freelance web and software developers are often painted as “fractured, footloose and free” (Huws, 2014), perhaps trotting the globe as a digital nomad, picking up to a new location whenever fancy strikes, while being paid high incomes. Some studies of technological workers might give the impression that they are as airy and unbound to material life as cloud computing: I wanted to discover the grounding elements of this work. Either by necessity or choice, what things might establish developers as stable fixtures in their local environments? Following that, I discuss developers’ perceptions of working freelance in Halifax, Nova Scotia. Geographical location, it seems, does indeed matter for the experience of internet-based work, especially considering the relationships between places like Halifax and central hubs of money and power, like Toronto and Silicon Valley in California. Freelance developers experience Halifax as a geographical place at a considerable distance from Toronto, but the city and its attitudes to technology, innovation and economic growth are also key to developers’ subjectivity—how they understand their work, their economic positions and their value as workers.

Chapter five explores the implications of these understandings of time and space for the political possibilities of freelance tech workers, examining the “hows” of freelance collaboration and meeting in time and space. While it may mean enjoying freedom away from many of the elements of employee-employer relationships, autonomy does not preclude a recognition of dependence upon others and a desire to be part of a community. Working in autonomy from the typical 9-to-5, together, makes up an aspect of Halifax’s freelance developer community and its collective goals.

As working life in the Global North increasingly takes on elements of freelance time, space, wage and social conditions—45 percent of Canadians generate self-employment income (Peyton, 2018)—the consequences of labour organizers favouring a Fordist vision of contemporary labour are that more and more of the work force is considered difficult to organize; worse, the persistence of this vision occludes from the view of both workers and labour activists the possible promise of quieter actions and everyday habits and patterns that produce collective behavior. These everyday patterns may someday make up a worker praxis—concerted, sustained and organized efforts to mobilize actions and behaviours over time and space (Jessop, 2005)—to meet specific challenges and exploitations head on.

Chapter 2: Developers and Their Work

It seems clear that, when a recent article (Terrell, 2018) in a lifestyle magazine suggests web work for the nearly-unhireable 55+ age group, a serious amount of deskilling has taken place in the world of work based in internet and web technologies. Though web and software development certainly used to be an example of white-collar work with many barriers to access, in 2020, many workers have come from the worlds of retail, low-wage service and manual labor to take up positions as web developers and designers. I have only my own experiences, evidence from those working in the field and some white papers to rely on, as this trend, thus far, has gone unremarked in both scholarly and journalistic accounts of web work. The Brookfield Institute's survey of developers, for example, reports that 60 percent of developers say that they are at least partially self-taught (2017). The internet abounds with variations on the archetypal blog post "Ten Reasons to...": become a web developer, or software developer, learn to code, learn to build websites and so on. My suspicions and anecdotes from friends and acquaintances were validated, when, one night out attending a tech-themed MeetUp, I overheard an owner of a local, boutique web development agency say, "All the oil jobs out West went away. They've come back to be developers." Rather than a career reserved just for the college educated, web development is now accessible to many.

A huge catalyst for this development is the emergence of easy-to-use, content management systems (CMS) such as WordPress, Wix and Weebly, which are built upon, and used in conjunction with, the great body of free-and-open-source software. WordPress is the most well-known, and the most used CMS, powering about 25 percent of the world's websites (Monty, 2016). WordPress is popular and barriers to use are extremely low, mostly due to its template system—with WordPress, one does not have to build a site from scratch. It is entirely possible to spend only a few weeks learning a small amount of mark-up language (code), and mastering its user interface and plugin system to then break out on one's own, building sites for pay.

That was the path of several participants in this study; only a few attended university for the express purpose of becoming a web or software developer, or something else related to the ICT (internet and computer technologies) industry. The nature and composition of technological labor has changed; programming, coding and building up the web is no longer only the purview of the college-educated, or scientists and engineers attached to governmental and corporate institutions. Most participants obtained undergraduate degrees in different fields and came to be freelance developers by economic necessity or circumstance, or in order to achieve autonomy in their work lives.

Some aspects of the ICT industry are slower to change: there are still fewer women in the field. I sought to achieve equal representation among men and women; in the end, there were eight men and two women. This is, in fact, more in line with the industry ratio, which is ten to one, according to the Brookfield Institute's 2017 report on Canada's developer talent (Lamb and Rubinger, 2017). The table below displays the demographics of the ten participants:

According to the participants, other non-participant informants and my searches on LinkedIn (a social network for professional networking), the differences between two freelancers who both occupationally identify as "developer" may be significant. For example, one may be self-taught and the other formally educated. Both may indicate on their CVs that they are "full stack", that is, able to execute both back- and front-end programming tasks, though their competencies along the "stack" may vary greatly. Further, developers may utilize only some of their competencies during any one project, alternately being more back-end, front-end, or even stepping into the area of graphic design, when needed. While the least skilled developer among the participants may be shy to call herself a developer if she were among extremely skilled

Table 1: Participant demographics

Participant	**Gender**	**Age**	**Higher Education**	**Years as Freelancer**	**Canadian/ Other**
1	Female	30s	Bachelor's	5-10	Canadian
2	Female	50s	Bachelor's (unrelated to ICT)	20+	Canadian
3	Male	50s	Master's (unrelated to ICT)	20+	Canadian
4	Male	30s	Some undergraduate	0-5	Canadian
5	Male	30s	Bachelor's	0-5	Canadian
6	Male	20s	Bachelor's	0-5	Canadian
7	Male	20s	Bachelor's (unrelated to ICT)	0-5	Canadian
8	Male	20s	Bachelor's	5-10	Other
9	Male	30s	PhD	0-5	Other
10	Male	30s	Master's	5-10	Other

programmers, her clients would label her as such (or she would market herself as such). I gathered that the term "developer" mostly belongs to the realm of the marketplace, while "programmer" is the term used in code culture, signifying something far more vocational and related to personal identity.

Thus, it is often difficult to identify how different developers are positioned in the overall production of web and software services and how that affects their pay and opportunities for work. Forsler and Velkova (2018) offer a useful production framework. They outline three types of workers, two of which are relevant here: the *efficient worker* and the *reflective practitioner.* Efficient workers know their tools as a result of repetition and training. WordPress developers are a great example. As mentioned, WordPress' template system allows for quick competency in the tool and could be learned in only a few weeks. A reflective practitioner may be the software developer who helped create the code upon which WordPress runs, and therefore, has a greater mastery over the programming languages required to know WordPress in an expert way. Efficient workers usually work on a higher quantity of short-term and less profitable projects, while reflective practitioners benefit from longer-term contracts on projects of higher value to the client's business model.

In addition to this framework, I offer two other terms to describe the natures of the projects developers work on that will be helpful as findings and analyses are presented in subsequent sections. Important to the working experience of developers are *precedence* and *definition* (the latter is called "scope" by developers). A project that is precedented is one whose tools and processes are straight-forward and in which there are lots of previous examples. A "brochure" website is well-precedented, for example: the client wants a website with only the business information, hours and a contact form. This type of project has been done a million times before. A well-defined project is one for which the end-vision is thoroughly explored and imagined with the client and developer together. A poorly-defined project leads to "scope creep": continuous, unstoppable changes and reversal of decisions to what were already vague objectives and goals, leading to last-minute add-ons and features. Projects may be precedented and well-defined (easy to complete but pay is lower); precedented and poorly-defined (path to completion is straightfoward, but client's end vision is unclear or changing frequently); unprecendented and well-defined (the way forward may be complicated, but tasks and management of project is clear); and lastly, unprecedented and poorly-defined (complicated and unclear). This last is to be avoided at all costs. Developers engage in several sorts of practices to either ensure well-defined projects, or to manage and endure ill-defined projects.

It seems that the bulk of efficient practitioners' work is made up of precedented projects (whether well- or poorly-defined depends on the clients, for the most part), while reflective practitioners work on a greater number of unprecedented projects. This typology is experienced more as a spectrum; there is a mean experience between efficient workers and reflective practitioners. Due to these distinct experiences within the world of development work, I chose to create composite characters, in part, to represent the experiences within this production framework. Next, I outline in greater detail the choice to use composite characters.

Dramatis Personae

As my research design did not reach full ethnographic potential, I felt a thoroughly narrative writing style—heavier on the use of vignettes and other story-telling devices—would not match the experience of the research design, either for myself or for the participants. Instead, I have chosen to rework the stories of my ten participants into three composite characters. Although appearing as protagonists in a small number of vignettes scattered throughout the thesis, they will mostly operate as a small chorus as I present the major theoretical, conceptual and empirical themes in each chapter. In this way, the reader might be better served by more fully imagined, whole people, rather than slivers and shadows, and the distinct developer production experiences outlined above may be preserved. To that end, here I introduce the three characters: Wonda, the WordPress Developer; Webster, the Web Developer; and Solomon, the Software Developer. Aside from these contrived, introductory narratives, all words uttered by these characters throughout the rest of the thesis are direct quotations from the participants. I have preserved colloquial, spoken English, making a few grammar and punctuation alterations for the sake of clarity only.

> **Wonda, the WordPress Developer**: My name is Wonda. I went to Dalhousie University and got my bachelor of arts there. I majored in psychology with a minor in film studies. All through high school and university I wrote and managed my own blog, which was a WordPress blog. I got really good at it, so I ended up creating and managing websites and blogs for a few businesses and agencies around town. I did a co-op semester for a graphic design agency during my third year, and then worked there for a little while after I graduated. They ended up downsizing the company a year after I started, and began passing off a lot of the design work to freelancers. Since I already had a relationship with them, I decided I would just go freelance, and so that studio actually became a client of mine. That was a few years ago. Now, I build and maintain WordPress sites for a fair number of clients, as well as consult on matters of design, marketing and branding. If they ever need me to do something that I can't do, I reach out to more experienced developers I know to make that happen. Since I'm more of a jack-of-all-trades and not

> a super-duper programmer, I position myself to clients as someone who has enough technical background and know-how to coordinate projects and assemble teams. I also work on my art business that's mostly based on Instagram. Art, tech and branding is sort of my niche.
>
> **Webster, the Web Developer**: I'm Webster. I've been developing websites for about five years now and freelancing for two years. I went to Nova Scotia Community College, where I studied computer science and business. I had a full-time job for about a year after I graduated from NSCC, in something unrelated to web dev. It was just a job where I had to sit and do boring things for eight hours a day. During the low times, I taught myself a couple of programming languages and started building up a portfolio of websites to show to potential clients. At this point, I would say I am a "full-stack" developer. The freelancing is still a bit difficult; clients are not always easy to manage, and the income is not stable, but it's way better than a regular job. I can work on other fun projects as a freelancer because I don't necessarily put in forty hours a week; it's usually around 30 hours.
>
> **Solomon, the Software Developer**: My name is Solomon. I went to Dalhousie for a few semesters for computer science, but did not complete my degree. I knew several programming languages going in, so taking the pre-reqs would have been a waste of time, and they don't really teach you the languages and frameworks that real-world development uses, anyway. I've been a freelance software developer for about ten years now. I work on a range of really interesting projects with just one or two major clients at a time. The most recent project that I am really proud of is a web application built to be a companion to a new exhibit at a museum in Europe. It's really interactive and allows for multiple types of engagement with the exhibit. At this point, I don't think I would ever go back to formal employment — I would be a bad employee!

In the following chapter, I review the relevant social scientific literature on labour and capitalist time, and how time has historically figured in political articulations of working classes. I then outline the main theoretical framework underlying the analysis of freelance developers' experiences and interpretations of their working time. Relying on a materialist understanding of the temporal techniques used by workers and capital and how those interplay with representations of time lead to my understanding of how developers' articulate and understand their freedoms and autonomy over work time and how this informs their collective and collaborative activity.

Chapter 3: Labour In and Of Time

Time has long been an object of struggle between labor and capital. The advent of the industrial, capitalistic age brought with it new, temporal disciplinary practices of time — the hourly clock — underneath which communities struggled to preserve their local, social times. E.P. Thompson (1967) observed that, during the transition to industrial capitalism, the internal experience of time went from a *task orientation* to a *time orientation*, whereby the industrial workers' (and their families') lives were re-centered around the factory's timepiece, rather than structured by the necessary tasks of the day (Glennie & Thrift, 1996). The clock became a representation of capital's abstract measuring of the value of labor and resources, rationalizing human workers into formulaic inputs who produced at the same rate, efficiency and quality over the 14 or 16 hours of the work day, which became, through the successful effort of labour movements, an enshrined-into-law, eight hour workday. Post-Fordism imposes other types of temporal discipline. A focus on particular objects of time discipline—or taking for granted the effects of those objects—may obscure how some inequalities and exploitations come about within specific working communities, and also how those communities articulate temporal concerns and act in response to those concerns. Combining ethnographic approaches with a materialist framework allows for a more thorough investigation into experiences of modern capitalist time.

Much inquiry into the experiences of contemporary worktime highlights the antagonistim between a community's social reproduction and hegemonic capitalist time. Relying on Marx's key insight that social time is always in antagonistic relationship with capitalist time, much social scientific research suggests weak worker agency in the face of capital's *homochrony* — homogeneous chronotyping of human labor, sociality and the natural world (Millar, 2015; Harvey, 1989). Even labour movement struggles, though resisting capital's temporal dominance, may adhere to its logic (Weeks, 2009).

Manuel Castells observes that *social arrhythmia*, the “breaking down of the rhythms, either biological or social, associated with the notion of the life-cycle” characterizes informational capitalism (Castells, 2000[1996], p. 476). Whether the *project time* of Silicon Valley (Shih, 2004) running workers ragged and sick through accelerated bursts of energy in the making of the next best piece of software, or the disciplining structure of precarious neoliberalism that keeps workers constantly reaching for a future that they will never have (Jokinen, 2016), this approach may mask how individuals and groups exert temporal agency in and over their lives and economic contexts.

An emphasis on homochrony, moreover, highlights some inequalities while obscuring others. If capital’s policies and disciplining techniques are seen as equally and evenly implemented, the experiences of local people responding to historically-precedented clock-time measurements and other temporal abstractions will appear more easily in analysis. But those workers who do not clock-in and clock-out, nor fill out timesheets at the end of the day, for example, may be overlooked because the visible, discernible and historical homochronizing measures are absent from their experiences. Ethnographic accounts of work in the contemporary world that are open to the possibility of heterochrony illuminate diverse, multiplicitous and synchronistic experiences of time, and serve as a corrective to some Marxist/political economic scholarship that represents the powers of capitalism and neoliberalism as all-pervasive, and experiences of those powers as the same everywhere (Bear, 2014; Millar, 2015; Tsing, 2015; Hodges, 2014). This is why we must not only make room in our analysis for heterochrony, but also attempt to discern the new forms, modes and objects of contemporary capitalism’s temporal abstraction machine.

For this project, I rely heavily on social anthropologist Laura Bear’s effort to offer researchers a way to analyze modern time and disciplining techniques, which, she argues, are experiences of “doubt, conflict, and mediation” (Bear, 2014, 2016). Her theorization of modern time relies on a meeting between Alfred Gell and Marxist philosophy. Alfred Gell, in his *The Anthropology of Time* (1992), offers three ideal types of time: non-human timespace (weather, seasons, tidal patterns, etc.), socially-framed time, and personal experience of time. Gell proposed the concept of *time-maps* to name how people create representations of time which then partially mediate those experiences (Gell, 1992). Gell’s model is similar to semiotician and philosopher Mikhail Bakhtin’s concept of the *chronotope*, a literary principle that describes how time and space are experienced and narrated in the social imagination (Bakhtin, 1981; Noble, 2016).

Bear proposes that modern time is characterized by labouring *in* and *of* time. Not only do workers labour within abstract measures such as hours, days and weeks, but they enact labor in the world to mediate and adjudicate between the various conflictual time-maps of capitalist, nonhuman, social and personal experiences of time. This labour of time stabilizes capital, which is inherently unstable. As Bear writes, “if [a stable capitalism] exists at all, it is due to emergent, piecemeal solutions for the contradictions produced by anti- or afunctional practices of contemporary capitalism. These ‘fixes’...emerge from diverse ethics and experiences of what it is that makes human labour productive” (2014, p. 78).

In addition to an insight from Marx’s second volume of *Capital* (1992 [1985]), where he lays out a diachronic analysis of circulation time, Bear relies—“strategically” and “critically” (2014, p.17)—on contemporary Marxist thinkers such as Michael Hardt, Antonio Negri (Hardt & Negri, 2000) and others from or influenced by the Italian *operaismo* movement, commonly referred to as the Autonomists. A brief overview of this thought is necessary, as it contributes to my analysis of contemporary working time, but also to the question of the political possibilities of precarious workers.

Scholars since the 1970s have sought to describe and theorize the material, temporal and spatial evolutions that have come since the decline of the Fordist epoch. Analytical terms such as post-Fordist, post-industrial, the information, or knowledge, economy, the network society, and semio- (i.e. semiotic) or cognitive, capitalism are used to describe the intensification of the shift of capital's extraction from bodies to minds. These terms explore how capitalism seeks to turn cognitive activities into profit-generating products while also attempting to trace how, through new regimes of accumulation after the industrial period, the disciplinary mechanisms of time and space change, and how workers' respond to and resist them. Here, I favor the term post-Fordism, as it is developed by Autonomist thinkers, as it recognizes the main theses of the information economy (cf. Castells, 2010) while providing the theoretical space to re-think the future politics of contemporary labour. These concepts aid in the task of discovering novel inequalities manifested by the newest forms of capitalist activity while also offering a way to discern the contours of new political bodies and classes.

The key drivers for this new capitalism, according to Autonomists Paolo Virno (2004), Antonio Negri, Michael Hardt, Mauricio Lazzarato (1996) and Franco Berardi (2009), are our cognitive, linguistic and affective faculties. Lazzarato calls this *immaterial labour*, in an attempt to describe both the move towards informationalization of and use of data within the global economy and the new resource inputs of this economy — our affect, sociality and cognitive functions. Importantly, these authors do not argue that there are no more windshield-wiper makers, or that manufacturing does not still have an important role to play in post-Fordist times. Autonomist thinking, as Paolo Virno explains, tries to describe:

> ...a set of characteristics that are related to the entire contemporary work-

> force, including fruit pickers and the poorest of immigrants. Here are some of them: the ability to react in a timely manner to the continual innovations in techniques and organizational models, a remarkable "opportunism" in negotiating among the different possibilities offered by the job market, familiarity with what is possible and unforeseeable, that minimal entrepreneurial attitude that makes it possible to decide what is the "right thing" to do within a nonlinear productive fluctuation, a certain familiarity with the web of communications and information (Joseph, 2005, p. 29).

In effect, what was once peripheral to the production process is now central to it. Virno talks about the "linguistic factory" (Virno, 2004); similarly, others rely on the *social factory* as a spatial metaphor to describe this new production process (Gill & Pratt, 2008). The social factory makes our relationships, affect, desires, logical and linguistic capacities productive — in essence, the entire continuum of life (Papadopoulos et al., 2008). A desire to engage with Virno and other Autonomists who focus on immaterial or post-Fordist labour prompted my decision to work with web and software developers. Their day-to-day is nothing but putting together logical statements and reasoning out semantics. "Programmers must take an idea from the realm of thought and embody it in functional lines of code..."(Rosenberg, 2007, p. 65). They also contribute, in collaboration with other programmers, to the great body of code that the IT industry uses for its own revenue generation. As Laura Bear points out, "[b]y paying attention to time, we can critique and measure inequality in new ways." (2016, p. 489). If post-Fordism describes a production process that encompasses the productive possibility of social life, elements of which are not obviously tied to historical time-discipline techniques, then time in this kind of capital-labor relation requires continued analysis.

Virno says, “Science, information, knowledge in general, cooperation, these present themselves as the key support system of production—these, rather than labor time. Nevertheless, this labor time continues to be valid as a parameter of social development and of social wealth. Thus, the overflow of labor from society establishes a contradictory process...Labor time is the unit of measurement *in use*, but no longer the *true* one unit of measurement.” (2004, p. 101). As Bear suggests, analyzing lived experiences of time, while recognizing that capital relentlessly attempts to rationalize it in creative new ways, can attune us to the inequalities present among labourers who, on the surface, seem to have escaped the time discipline of labour time, per se. In my analysis, I will show that freelance developers manage to distance themselves from some techniques of time discipline, while falling prey to some other ones; and that the nature of creative, cognitive work, along with the rhythms of the internet and code, produces a significant amount of unremunerated and invisible labour. An examination of nonhuman rhythms of the internet and code may be a good place to begin.

Hurricane Dorian and Rivers of Code

On September 8, 2019, Halifax received a visit from a strong, category-one hurricane named Dorian. Thousands of people lost power, and although the damage was less severe than during the 2003 Hurricane Juan, Dorian outdid Juan in the extent and number of the power outages throughout the Halifax Regional Municipality.

Hurricane Dorian was a little serendipitous: it provided me the opportunity to ask the participants to reflect on a very recent, natural event that directly affected their main “productive inputs”: electricity and an internet connection. I wondered: are freelance tech workers more concerned about the probable, near-future changes that the climate crisis will bring? Do they put energy back-ups in place; do they own generators? Do they worry about far-off server farms that hold their code, dependent upon air conditioning to keep the machines cool? Dorian caused a power outage for nearly a week in some parts of the downtown area, and longer in some rural areas. Did this have developers scrambling to meet deadlines and make up for lost work time?

The short answer to all these questions was: No. Many expressed concerns for the climate change in general; outlined what they do, personally, to keep their carbon footprints small; expressed confusion about what we should be doing collectively; but, when it came to their work and businesses, there was hardly any concern at all. All of them pondered the question about the effect on their work with tranquil acceptance. They were not necessarily unconcerned; they simply faced the prospect of limited or spotty energy availability in the same way we face hurricanes: resignedly, as there is nothing we can do to keep the storm from passing. As Webster and I were sitting over coffee in a local deli, I asked him if he was concerned in any way about future variability in energy accessibility. He laughed and said, “I am now!”:

> **Webster**: I mean, if our energy and internet started to get spotty, at that point, there’s bigger fish to fry than “I don’t have power sometimes”. I would probably consider going somewhere else. But, how long until those problems follow you there, right?

Though my assumption that developers would care more than most about energy availability ended up being rather a “null hypothesis”, developers did express frustration and anxiety around another external, uncontrollable force that frequently—on a weekly, and sometimes daily, basis—affects the quality and efficacy of their code: the internet and the collectively-built code that forms it.

The internet (and software) operates on, and is built up with, code literally written by millions of people, past, present and future; new code piled on top of old code. Over time, with the introduction of new programming languages, developer frameworks and new hardware innovation, coding languages and bodies of code start to behave differently and sometimes erratically, as a change in the code closer to the foundational languages affects the behaviour of the code written in more abstract languages. Modern programming languages have a lot of abstraction built in: because it is easier to use code already written for certain everyday functions, one line of code could represent dozens of other lines of code that are essentially pre-packaged. If anything changes in that package, things start to go wonky in the code that uses it."Programmers call these accretions 'layers of abstraction,' because each time a new [layer] is added, something complex and specific is being translated into something simpler and more general...[a]nd every year, the piles grow higher" (Rosenberg, 2007, p. 66). Software, code and the internet are forces unto themselves, and the continued abstraction and sophistication of languages and code creates instability, while also further divorcing it from the control of any one person or group. "The internet is a giant chaotic mess of duct tape and twine mostly held together by volunteer work, but it actually kind of works, which is terrifying", as Solomon laughingly described this state of affairs during an interview.

Developers experience and perceive the dynamics of code and the Internet as an uncontrollable, non-human force. They step into, what Solomon calls, "rivers of code", and grapple with "bugs". Developers might fix a bug, only to find that in fixing it, another one pops up somewhere else. Deploying the code, too, interacts with other bodies of code and software programs in unpredictable ways: "software...engenders direct effects in the world in ways never...expected by their creators and in ways beyond their control or intervention." (Kitchin & Dodge, 2011, p. 39). They would agree with Bruno Latour when he writes "software is an actant in the world" (cited in Kitchin & Dodge, 2011, p. 39).

> **Solomon**: React Native [a way to develop mobile applications] is great when you're working on it, but it's kind of like a river: you never step into the same React Native project twice. Whenever I start a new project,

> something else breaks in a totally undocumented way, and I have to spend hours combing through the internet trying to find stuff. Especially with JavaScript [a programming language], everyone is using it so much, it's in flux all the time, it's changing all the time, and that's part of the headache.

The Internet has enough stability and predictability to put it to economic use, but on the other hand, it has enough inherent chaos to disrupt, stall and degrade the economic uses developers make of it. I want to emphasize this point because I believe the chaotic nature of code and ICT and its effects on the livelihood of programmers is not well understood, nor analyzed in ways that provide understanding of their labour time. Developers' grappling with the unpredictability of the internet and code, sometimes for hours or days at a time, is not always reflected or accounted for in their invoices to clients. A bug could completely derail work plans and project timelines, causing disruption in pay or confusion in clients about hours worked. Chasing bugs can also disrupt the personal rhythms of the developers. Webster found himself up until five a.m. one morning, as he recounted in a reflection entry:

> [**Webster, October 12, 2019**] Today was less ideal after starting to work later at night; accidentally worked until 5 a.m. fixing a long-standing performance bug which I just discovered the source of.

Judging from the content of the interviews and the reflection entries, not a week goes by without the necessity to spend some hours (or days) chasing down a bug or two within dozens or hundreds of lines of code. Chasing bugs can keep a developer up all night, or force him to put off the necessary work — the work the client expects — that progresses the project. There is also the matter of the developer's own bugs which are made consciously; sometimes, a developer must write some poor code in order to meet a deadline or to create a working prototype in order to get the go-ahead from the client (or even a potential client). This usually leads to some back-tracking, or "re-factoring", of code in the later stages of the project. Often, the difference between an ideal day and a not-so-ideal day is the presence of bugs. Here are a selection of statements from the developers' reflection entries:

> [**Webster, November 3, 2019**] Some of the work today was a little frustrating, as a simple update to a site collapsed into endless tweaking of settings and configurations due to a server upgrade.
>
> [**Solomon, November 28, 2019**] I think I was probably productive today, but I don't feel like I moved things forward very far. During a meeting with a teammate, I uncovered a rather large bug. I fought the bug for a few hours, and was able to solve it, but I was really out of energy. I just feel I could have done a lot more if I didn't have to spend so much time fixing a single bug.
>
> ...
>
> [**Solomon, December 21, 2019**] What wasn't great about today was that, instead of working on the new features I'm trying to include, I needed to go back and refactor a bunch of old code. The last crunch [in the project phases] required me to cut a lot of corners to get to the deadline, so now I'm cleaning up a bunch of that mess so I can build the new features on top of it.

Further, resolving a bug or creatively solving a problem may benefit more than just the developer—developers everywhere, including those working for Microsoft, Google and other tech giants, rely on the same packages of code that are free-and-open-source or accessible by fee on GitHub, for example, which is a software development platform owned by Microsoft. Developers may post their code there publicly, or refer to others' code for their own projects.

The instability and ever-evolving nature of programming languages and bodies of code has work time implications for the developer over months and even years, not only throughout the brief weeks of a project. Those layers of code not only continually accrete, but their foundational languages and means of organizing it will change, degrade and depreciate over time. Clients' projects may re-surface after months or years if updates and changes cause the project's code to behave differently: after these varying intervals of time, developers' pricing strategies may no longer apply, or their work ethic dictates that they bring the code current for free (this is discussed in greater detail in the following section). To make an analogy, the constant change in code would be like a factory worker walking up to her position at the conveyor belt every morning to find that her tools have been slightly altered. The supervisor has left no memo but she must reach his quota anyway, having no indication of what this new tool means for that day's production of the product. Often these changes and upgrades to code result in trivial work for the developer, but a big change might mean lots of testing and tweaking of sites and programs, sometimes across multiple operating systems. In a very real way, developers are responsible for all of the code and projects that they have ever created and must attend to this body of code if they want to preserve their "brand" in the marketplace, as Webster describes:

> **Webster**: Sometimes an old way is deprecated or depreciated, so you have to fix that because it throws an error. And I find that's a pain. Also, I use a back-end language now that I just want to throw away, because it's built for how things used to be instead of how things are now. I don't want to build stuff the old way because that's not good for building my own brand.

Their "brands" are not only built on how they present themselves to clients and the way they build up their image in the marketplace; for developers, brand image is also partially built upon the continuation of quality, over time, of websites and programming that are in constant flux. Unlike the factory worker in the analogy given above, the developer does not simply part with his product, like the factory worker parts with her completed auto part, for example. That developers and their product remain attached to one another over time is a reality conveyed by one of the industry's clichés: "a project is never complete."

A Project Is Never Complete

Sitting at my desk one day, I heard the *ping* of an incoming email. I pulled the browser tab up quickly; it was an interview day, and I was more alert to my messages in case of last-minute re-scheduling. Sure enough, Wonda had sent me a LinkedIn message:

> "Hey Kate. I have to cancel our meeting AGAIN...Some stuff I did for a client a few months ago has reared its ugly head, so I have to rearrange my whole afternoon. Meet next week?"

No interview question received as much sardonic laughter as the one that asked: "When is a project complete and how do you know?" It is an industry hazard that plagues all who work in software (Rosenberg, 2007). The nature of a developer's code and how it lives in the world means that the developer is often connected with the code and the project long after the conclusion of the contract. Many developers do small tasks and rounds of maintenance on sites or programs months or years after the client signs off. Often, the developer will do this for free, either because it is trivial, or because he wants to maintain the relationship with the client, or he is watching out for the reputation of his "brand", or all three at once.

> **Solomon**: The nature of these things is that, you've sent them off to live in the world, and until somebody explicitly erases them, they're there. So if there's an edit or change, or if it goes down, or if it stops working correctly or somebody has an idea for what they'd like to add; all of those things could make my telephone ring.

Having a very faint idea of this reality before I began conducting interviews, I envisioned the developer—a man, given the gender balance in the industry— sitting down at his desk each morning and opening his time management web app. Toggl, Harvest, Clockify, Clockk, Todoist, Pomodoro Timer, Noisli, Trello—enter "time tracker" or "time management" into the Google App Store, and you will see that the search results are many. Some newer time tracking applications are programmed to watch what you do on your computer—which programs you open, how long you spend in them, when you close them—all in order to record every second of productive work. I assumed that, since creative work such a programming has these slippery temporal characteristics, developers would be keen to track every moment at work.

Instead, what I discovered was a mostly flippant regard for tracking one's time like this. Although many developers do bill by the hour and time-tracked their sessions at the computer for some of their clients and projects, this method time tracking is not preferred, and most developers rely on other systems to steer their invoicing and billing systems away from the logic of pay-by-hour work. What I found over and over again was the often simultaneous use of three billing techniques: contracts that spell out the scope of a project, or part of a project; "packages", which are bundles of services sold to the client with associated lump-sum fee; and lastly, hourly billing. Developers moved between the use of these three means, and sometimes all three over the course of the project, depending on client and project specifics.

These billing tools, of course, are a way to protect the developer from the unpredictable vagaries of programming, ill-defined and unprecedented projects, and the bug grappling that will inevitably happen:

> **Wonda**: I work with packages. Once in a while there will be things where you just can't know how long it's going to take or what it will entail, so it's impossible to create a package. Then you would have to try doing it hourly for a short while to come up with a package, but I avoid doing that as much as possible.

Webster usually draws up a contract, stating exactly what he will do in what time-frame, and then will bill by the hour for those more trivial, precedented tasks that usually occur towards the end of a project or a few weeks after the conclusion of a project: a change in the color of a button, or a re-visit to fix the appearance of the site on a mobile phone.

For all of them, the time it takes to complete a task—even the most precedented one—is subject to the developer's elastic notion of "time spent":

> **Webster**: For clients, my minimum is half-an-hour. So, if something takes five minutes, that's great for me. After that, I bill in 15 minute increments. If something is particularly frustrating or really buggy, I'll round up to the nearest 15-minute increment, even if my timer only says "7 minutes". I ask myself: Was it a real pain in the butt? Yeah—that'll be an extra 15 minutes.

When they can get away with it, developers rely on the "black box", as Webster called it, to keep their wages as high as possible. Webster uses the term black box to describe how many clients view the developer and his work—like a wizard-type figure waving hands and whispering incantations over a technical problem behind a curtain. If the client himself is an expert programmer, or has programmers among the decision-makers, there is usually less opportunity to draw up contracts or use packages to insert padding to protect from code variability and client whims. In this case, hourly billing is the norm.

Further, though they depend on the "black box" and the practice of time padding contracts to protect themselves from inevitable accidents, calculating by the hour remains a key heuristic in determining what to charge. After assessing the project scope, they make a stab at how many hours it might take and then assign a per-hour charge. This per-hour charge is determined by industry averages and what they think the client will tolerate, but the effect of "gig economy" find-work platforms such as TaskRabbit, where developers may charge as low as $15 per hour, drags down potential earnings for freelancers. They may guess required hours more-or-less rightly, if the project is precedented and well-defined, but all bets are off if this is not the case:

> **Webster**: Sometimes I can guess, sometimes I've been very wrong. I once had a project that looked like a very simple site, but turned into a two-month long nightmare. But because I had quoted everything upfront, I was stuck. That's what I got paid.

Software development often represents uncharted territory. Each new client may present a vision that the software developer builds from scratch. Though this usually does not mean that the developer learns entirely new programming languages or frameworks, for every new client; most developers rely on the same few languages and frameworks. However, the software developer often learns on the job, adding some new skill or approach—or becoming current with changes in the language—with each new project, as he attempts to bring to life the client's vision of the product or piece of software. This requires many meetings, an iterative approach between the vision and the code, and most likely, at least one tear-down-and-start-again moment. For such unprecedented work, the software developer relies almost exclusively on the contract—even a series of contracts—as a means to manage a client's expectations, protect himself from a client's whims, and to put in some padding for when things go awry.

> **Solomon**: Especially with software, clients never really know what they want until you've built something and see that it's not really what they want. It's not like building a building where you have the blueprint. Everything takes longer than you think it might. You try to write up a really good contract, but going over budget, having to bite off hours on the invoice—this happens basically all the time.

Using three ways of charging the client does not always protect the developer entirely, or thoroughly, from the "scope creep" of unprecedented projects, client whims or juggling all of the variables required for web work. In fact, it seems that the longer the client and developer are in a contractual relationship with one another, the more work goes unremunerated, even if it is not technically free. Solomon and some of the non-participant informants who have the most freelance experiences reported that longer-term relationships might mean that the developer is called to work on smaller and smaller tasks as time goes on, but might put in the same amount of client relations work; or, because that client remains in the roster, it prohibits taking on new clients.

Though the use of contracts, packages and hourly billing may, at the end of the day, be a substandard tool in capturing all potential earnings, it allows for an extremely important practice to take place within the working life of the developer: the contract allows the time and relational space (distance from micromanagement from the client, for example) necessary for the developer to lean into his or her own personal and productive rhythms and whims.[1]

> **Wonda**: I love working with packages. For me it's less brain clutter, I can just do the work.

> **Webster**: This work is definitely more about tasks than it is about time. So, if I get a bit tired after five hours, I can take a nap. Sometimes I need only two hours. If I don't get something done in a certain amount of time, I will just move it to another day.

> **Solomon**: When you hit that "start" button, you have to be "on". But, take for instance this project I'm working on now: we've divided it into four phases. Because that's a few weeks long, I was able to take a weekend after our most recent call to rest for a little while; and I was able to just throw my hands up in the air and give in to the power outage when Dorian came through. I read a lot of books that weekend.

Freelancers use a handful of means and strategies to capture as much of the work they do within their invoices to their clients. Some of this work is impossible to capture, including elements of work necessary to freelance successfully that is outside of the set of tasks directly attached to a project, like hustling for new contracts or

[1] A possibly helpful concept here might be Andre Gorz's *heteronomous work*, work that is determined externally by organizations and abstract logics (1982). Workers, Gorz argued, may only find freedom outside of this kind of work, in the realm of autonomous work (Furåker, 1984).

writing proposals. In the next section, I will describe how the term "freelance" is not merely a label or a tax category. Rather, the term should describe a whole set of labouring practices which may go unrecognized, unpaid or inadequately remunerated.

The Work of Freelancing

The first WordPress MeetUp I attended was held on a balmy September evening. I arrived about fifteen minutes early to the meeting place, a bookable conference room in Volta, Halifax's most successful (and most subsidized) co-working space (there are at least two others in the downtown Halifax area). A bit nervous about trying to ask intelligent questions to people with enormous technological skill, I braced myself for a room that I imagined to be full of nerdy men, bespectacled and skinny, but with the belly paunches typical of coding enthusiasts. In my mind's eye, they stare intently at their computer screens, throwing their heads over their shoulders to talk to their neighbors in unintelligible techno-speak. I entered, instead, a different room.

Mostly soft grey, littered about with vibrant green desk-chair hybrids, the room was as quiet as a library. A dozen or so people, of all ages and backgrounds, were waiting patiently for the meeting to begin. A trio clustered against the far wall were chatting almost silently; others had their hands folded over their notepads or laptops. It could have been the last few moments before the start of a night class in community college. A blond woman in her forties clicked her pen; a young, possibly Middle Eastern man energetically tapped his foot on the floor. Around 7:05 p.m., the moderator came bouncing into the room, dropping a bagful of colorful candy on the table next to the door. "From our sponsors!" she sang out, enthusiastically. We all chuckled.

The meeting began with its usual agenda items: the goal of the MeetUp, a message from a sponsor, an introduction of the moderators. Both moderators were freelancers who volunteer organized the MeetUp. With an encouraging and relaxed demeanor, they explained: “We’re not experts, just people who love and work with WordPress a lot. We want to help and grow the WordPress community”. They asked us to quickly introduce our selves: What is your name, why are you here and what is the url to your WordPress site?

The blond woman explained that she is struggling with her cooking and travel blog—how might she effectively drive more traffic to her website? Her click revenue had been dragging the past few weeks. The young man with the foot tap explained that his clients will be expecting certain WordPress features and functionality that he does not have a handle on yet. He hopes someone there might help him during the portion of the meeting called the “Happiness Bar”, a time for peer-to-peer troubleshooting. The trio sitting behind me came to recruit; they had web development positions in their agency to fill. When it came to my turn, I told everyone why I was there, and though I had no url to share, I called out my email address, which the moderator added to the Word document projected on the white screen at the front of the room. At the end of the meeting, I chatted with the trio, while others got the help they needed at the Happiness Bar.

Throughout the meeting, I was struck by the diversity in the room. Not only the demographic diversity, but the work diversity—many were freelance, self-employed or part of the growing group of people who try to earn extra cash through their blogs or social media accounts, while also formally employed full- or part-time in other industries. The classroom-style atmosphere brought some of my reading about the contemporary workforce to the forefront of my mind: freelance tech workers must learn and re-skill more or less constantly. How much of their work lives are occupied with learning and re-skilling, and what other work demands must they meet that may not result in direct earnings?

Though freelance developers may expect to earn between $20 and $50 per hour (according to the participants), the hours they spend on client projects do not make up the total hours of work they put in every day, especially for those freelancers who have decided to make this a way of life—that is, to put in the work to make this reliable into the far future. Wonda, for example, only expects to achieve between three and four hours of "billable work" each day—the rest of her time is spent contacting potential clients, writing proposals, organizing and recording her expenses, and attending workshops, panels, and other events to network and keep an eye on trends. She may present at such events or sit on a panel, sometimes for a modest honorarium, but frequently not. Though I did not attempt to rigorously record how many participants did or did not include things like email correspondence, proposal writing and phone calls into their invoices for clients, I gathered that much of this work goes uncaptured.

Talking with the participants, it became clear that freelancers might be considered to contain, all in one person, the various departments of a regular corporation. They do the work of the marketing and branding department; they are in charge of payroll and taxes; they do the "C-suite" work of creating all business strategy, setting goals and objectives. They do R&D, learning new skills and developing new approaches to work, earning and product development. They are the sales team. An appointment with "HR" might be sitting down with a journal to record successes and frustrations, or calling out for commiseration on Twitter—Wonda has made this a habit in her own business practices.

Additionally, the resources and structures in place within a corporate environment that protect the corporation from all kinds of liability, unpaid invoices, and other potential harms to business, freelancers must create on their own (cf. Pitts, 2016). Some retain the services of a lawyer, in case of theft of service. They are often responsible for knowing government policy and regulation concerning web security and the careful handling of customers' information and data (this was the topic of the second WordPress MeetUp I attended). They are in charge of their own taxes, or they hire an accountant. They have to handle all of the communication between the organizations who host their websites and the people who run and manage the servers where the data is stored. An entire morning may be lost in the act of contacting the website host, locating the correct passwords for all sorts of accounts, or ensuring that clients' data is secure and intact, should something irregular happen.

They often act as tech oracles, always keeping their eyes on developments in IT, either at their client's behest, or from the need to prove one's competency and competitive edge. They will act as consultants frequently, and only those who include such a service in their contracts or packages will receive remuneration for it. And, unless they are diligently starting and stopping a timer, the back-and-forth between clients and project team members through email, Skype, or team-based communication platforms like Slack, those conversations—and the mental effort required to transition from team to team, problem to problem—goes unaccounted for. More experienced freelancers take much greater care in charging for these things; the less experienced are more concerned with collecting and retaining work and will often let such services slide out of invoices.

Sometimes, a developer will suffer from "project inertia", as Solomon calls it, where a project stalls because the client is late in getting data or crucial information to the developer. The behaviour of the client has a great deal to do with the developer's working life; an unresponsive, incompetent organization will cost the developers time and money, and may force them to lose out on other opportunities.

Clearly, the labour of freelancing enacted by developers exemplifies the neoliberal move to shed as many tax, infrastructure and employment burdens onto individuals as possible (Alberti et al., 2018). However, most developers take all of this in their stride, preferring to celebrate their freedoms and autonomy, chalking up most of this unrecognized work as simply the normal way of things if you are freelance. The more experienced one becomes at the labour of freelancing, the more automatic it is. It takes several years to get good at it, and those I spoke with, even Solomon who has spent much of his working life as a freelancer, reflect on how difficult some of these things remain.

For freelance developers in Halifax, it is the time-map of the corporate, formal 9-to-5 job that dominates their imaginations of what the world of work looks like, including its unfairness and injustices, and it is in response to this time-map that freelancers come to understand their choices as freelancers, and recognize the effects of their working decisions and actions in their own lives.

The Time-map of the 9-to-5

I was amazed at how frequently freelancers invoked the stereotypical characteristics of the corporate job, bound between the hours of 9 a.m. and 5 p.m. Even for those freelancers who had spent very little time in full-time, corporate employment, the 9-to-5 was a prominent presence in their discussion of how they arranged their space and time. It seemed the most salient representation of the regular world of work—an idealized notion, possibily, considering that many formally employed people who work in offices work more than eight hours a day, or have alternative hourly arrangements with bosses.[2] The strict 9-to-5 seems to belong now to a bygone era. However, the participants, as they recounted to me during the interviews, frequently responded and referred to this idealized notion by way of their temporal and spatial practices and attitudes, indicating that freelancing in itself is a form of resistance against certain forms of labour discipline specifcally attached to the corporate office

[2] A report by The Conference Board of Canada (2018) found that 93 percent of Canadian organizations offer flexible work hour arrangements to their employees.

job.

The 9-to-5 representation includes several salient characteristics. First, leadership and management display numerous deficiencies and are, on the whole, mediocre. They are slow to act and make absurd requests. They have insufficient knowledge and experience of the technical skills of rank-and-file developers, and therefore make strange and arbitrary decisions concerning corporate-level technical protocol. They hold back innovation due to difficulty in organizing synchronicity among various departments, or they are simply disinterested in pursuing innovative projects.

> **Wonda**: When you work in an office, you have to work within the confines of everyone else. And I don't like being told "No you can't do this because I don't understand how it works."

> **Webster**: This one job...I had to learn so much, and they felt I should be moving faster, but the application they had was a mess. I felt like it needed to be redone from scratch; it was like, duct-taped together from ten to fifteen years of different programmers coding in it.

> **Solomon**: I left my last job partly because I was having issues with the corporate bureaucracy. New management ended up issuing a whole bunch of rules, like what to install on our machines, which ended up interfering with the work. Ironically, they introduced the rules to capture the innovative work we were doing, but they ended up just killing it.

Second, corporate temporal logic insists upon the model of a worker who outputs a consistent and constant rate of production and efficiency throughout the eight-hour workday; and what is more, this logic assumes that the 9-to-5 captures the most productive hours of a human's daily energy output. Further, the 9-to-5 logic demands that workers remain present and engaged throughout these eight hours, even when there is nothing to do, or all the day's tasks are complete. These insights gleaned from the participants confirms an important point in *Time, Labour and Social Domination* by Postone (1993) which argues that the abstract measures of time deployed by capitalism take on a teleological necessity of homogeneous, ready-to-be-filled, blank time. This blank time thus becomes an objective norm, squeezing out the reality of concrete, qualitatively-variable, lived and social time. This objective norm, with all of its attendant temporal disciplining mechanisms, is singularly abhorrent to all the participants:

> **Wonda**: When you're working 9-to-5, you're just there because you're supposed to be there from the hours of 9 to 5. And it's like, you're sitting there playing solitaire because there was literally nothing to do. I don't like that.

> **Webster**: One of the things I found working at a 9-to-5 job is that, sometimes, I just didn't have it in me. I'm sure everybody feels the same way. But sometimes the work isn't there. Whatever mojo needs to be there for me to get shit done is just missing. And sometimes I just need to call it a day.

> **Solomon**: Full-time weeks are harder on people than they realize. Especially in a field like programming, the stress that gets put on your brain. You're kind of dazed after a long day of working, and five days a week, eight hours a day—that puts a big mental toll on you. I'd say I have two to three hours to be fully engaged in work, tops. And, on having to take your lunch at a specific time: what if I have to take the dog to the vet, or something?

Like Gell's time-maps and Bakhtin's *chronotope*, the corporate 9-to-5 is both a time and a space, a reference point on a narrative map toward which freelancers orient their choices (Noble, 2016; Bakhtin, 1981; Gell, 1992). For the purposes of thematic organization, space is considered in Part II, but the experience of time and the experience of space are so connected that the experience of one can be said to be an experience of the other (Davies, 2003; Lefebvre, 1974, p. 12), so a discussion of space relating to the office job is warranted here. Developers characterized the spatial characteristics of the 9-to-5 as confining and inimical to productivity. The spatial layout materialized the offensive and absurd characteristics of corporate cultural norms, bosses and management:

> **Wonda**: I didn't feel comfortable moving about or taking breaks in the corporate environment. It was like, taking 45 minutes to do yoga at the gym next door was frowned upon. I would start finding an excuse for frequent bathroom breaks, just to get up. And, the developers and designers at my old job were split up, on different floors. We were kind of siloed. The boss didn't understand that that's not how we, as developers, work.

> **Webster**: I had my own office at my old job, and I always felt like I was rattling around in a cage. Not necessarily because I hated my office, but when you work in an office, you have to work within the confines of everyone else. Working for myself, I'm not ever bound by the limitations of others—I get to learn all the time and my work is interesting.

> **Solomon**: At my old job, most people were working in an open-concept workspace, which I am not in love with. Everyone was just in this giant space. Way too noisy for development work. It was one of the reasons I left. I just can't take that.

Developers—engaged in work that is more akin to writing than to, say, bridge building—have a different notion of what productivity looks like for them, and it has very little to do with hours spent. When developers talk about being productive or feeling productive, they often mean reaching the state of "being in the zone"; this phrase appeared early on in the interviews, and so I was able to ask each participant about this state. When I asked, everyone knew exactly what I was talking about; according to participants and non-participant developers, this is a shared experience among people who code. Summarizing their descriptions of it, "being in the zone" means reaching a state of deep work, where they are working—effortlessly—on the internal logic of the project. Two hours of being in the zone is more satisfying than, say, five or six hours of more shallow, trivial work. The shallow work is, paradoxically, often more visible to the client, such as a change in the color of a button on the screen. Deep work, or, "being in the zone" is not so easy to achieve every day, nor is it a constant experience, and neither can it be easily stopped (or begun), simply because the clock has struck a particular hour. Solomon describes it this way:

> **Solomon**: Like, take a stenographer. Suddenly that words per minute limit is lifted. Time is shifting, it becomes more malleable. It doesn't feel like you're rooted in time anymore. You feel like you're more a master of it, even though, rationally, it progresses at its linear pace. It's like I'm working at light speed; thoughts are progressing more smoothly; I have a clear picture in my head of what I am trying to do; I can open more [desktop] windows and work through them faster; and my connection to the keyboard has become stronger; and I feel that I've become one, not just with my computer, but with the project. You are communicating what you need to exactly as fast as you need to. Time stops affecting how quickly I can work.

Developers report that they most often achieve the zone state when the project problem is clearly defined; when they have few other stressors, including those of family and personal life; and when they have large blocks of time that remain uninterrupted by meetings and requests sent via email and text. In fact, knowing that their mornings will be completely uninterrupted is a highly important factor in reaching the zone state. A non-participant developer told me that he more or less structures his day in order to create the best conditions for reaching the zone state. The external impositions experienced at the 9-to-5, like being burdened with useless meetings and constant interruptions, the corporate work place is often where "being in the zone" is hard to achieve.

Highly popular films like *Office Space* and television shows like *The Office* are useful examples in imagining how developers, and many of us in other industries, regard the corporate environment. Absurd, incompetent bosses; useless paper work; arbitrary dictates about where, when and why work must happen. We use phrases like "rat race", "gerbil wheel" and "cubicle rat" to describe the banal inhumanities office workers suffer. It is far from coal mining, but it is "hell", as Webster says:

> **Webster**: Right now, I do a lot of work for one client, and that's rather putting all of my eggs in one basket, but, the freedom of working from home, not doing a strict 9-to-5...I want to make sure it stays that way, 'cause I don't want to be in a cubicle doing 9-to-5. I said "never again"

> after my last job. To me it's...hell. It's just terrible. I can't function. Personally, on a mental health level, everything just deteriorates when I'm in that environment.

The 9-to-5 time-map plays a powerful role in mediating developer's experiences of their freelance work time. It mediates how they come to know things about their own labor and agency. In their quotidian responses to this picture, freelancers can come to understand how, in their freelance work, they prioritize, and take the lead from, their own personal and productive rhythms, and actively promote an ethos of care for self and family. They can work transparently and truthfully, not having to fake how many hours they have logged; they can drop all pretense around meeting norms of productivity and work-place engagement:

> **Wonda**: I think it's that employee-averse person in me that really hates tracking my time. It's a distraction for me, and it's just that extra thing you have to do, writing it down. I'd just rather get the work done. And, tracking hours does not give a complete picture of how my business is going; for example, today, I didn't accomplish much for billable hours, but I did meet with a colleague to spitball options to collaborate on speaking engagements. So, not much in the money department, but I felt I was moving the needle forward. It's a great reminder that I have flexibility in my business that a 9-to-5 wouldn't offer.

> **Webster**: Today I was getting over a cold, too sick to really go out but not sick enough to be forced to stay in bed. The perfect kind of sick. This is another place freelancers have an advantage I think. Really you have as many sick days as you want, and I feel like the comfort you get from being at home next to a box of tissues and a bottle of cold relief medicine turns a potentially nothing day into whatever you want it to be.

> **Solomon**: What's different about time-tracking as a freelancer is that I don't have to record all the points where I was being non-productive [like a lunch hour] so it means that throughout the day the number of hours I record is significantly less, but they are the hours I'm actually working, not the hours I'm pretending to work. I think about all those people on an 8-hour-a-day job, and how much of that day is actually work. A lot of it's...coffee *[laugh]*. And useless meetings. One of the things I like about my lifestyle right now is that I have the flexibility to go to the vet and

> grocery store at 10 a.m. without having to get permission from a boss. I mean, this here, interviewing for this project—I could be working right now! But I don't have to.

When freelancers use the term "freedom" and evoke images of it, it was, nearly universally, a freedom *from* the constraints of the 9-to-5. Freelance developers want to pursue interesting projects. They want to make decisions about how much time to spend on a task, or not to spend time on certain tasks. They want the freedom to follow their own productive and physical rhythms, to capture the "mojo" when it comes around. They want to decide when they need absolute quiet, and when they need the friendly hum of the company of others. They know that some programming will go faster or slower under specific conditions, and they make work plans accordingly. They realize that productivity, like creativity, ebbs and flows; there is no such thing as a constant rate. To my great surprise, then, I found that they rely on the 9-to-5 time-map to realize this *freedom from*, replicating its elements in their daily lives.

The Ideal Day

The ultimate question in the first interview guide, the one everyone is asked, is: "What is your ideal day as a freelancer?"

I expected a somewhat high degree of diversity in these answers. I expected some repeating statements about earning as much money, or working as few hours, as possible, but other than that, I readied myself to hear all sorts of different things. However, a bewildering degree of similarity emerged through the answers to the interview question and among the responses to the related reflection entry prompt.

The first indication toward similarity over difference emerged from a research instrument I did not realize I had, at first: the consent forms themselves. I began to notice that, when the forms asked which days would *not* be convenient ones to receive requests to take a photo or make a reflection, eight out of ten circled “Saturday” and “Sunday”. I expected more of them to work over the weekends, or have a lax definition of “weekend”, or to simply say “no particular day” would be inconvenient.[3] Similarly, when asked which time-frame they would most like to receive reminders to take photos for their photo journals, most participants indicated that they began work around 9 or 10 a.m., and ceased work around 4 or 5 p.m.

A clear daily structure emerged as preferred and ideal among most participants. It generally goes like this:

> **Wonda**: I wake up, make tea, and I’m at my desk at 9. I close a client by 12. I have a good lunch. I’ll do some volunteer work, then take a walk, and then come back to my desk for another couple of hours. No interruptions is ideal. By 5 o’clock —I’m gone. I do not answer phone calls after five, and I’m not available on weekends.

> **Webster**: My ideal day starts early. I’ll do my 4-hour block in the morning. I might take two hours for lunch. Then I’ll come back for another 4-hour block. Then the evening is whatever. Cook a good meal. Then do it all over again the next day. It would be a regular work day for someone, like 8 to 4. If I have that day, that’s an ideal day.

> **Solomon**: Get up, catch up on emails, then head out around 9 a.m. Get in a couple of hours of solid work, then have lunch; wander around and do an errand or something. Then another couple of hours in the afternoon, making progress on encouraging projects. Then finish up around 4 p.m., then head out to eat or grab a drink and work on my side projects, or meet up with friends.

Essentially, the pattern that repeated itself was: wake, commit to a timeblock of work, eat, do another timeblock of work, then head home (or leave the home office). Of course, everyone added descriptions that de-clawed the typical workday at

[3] Of course, developers may take weekends off because family and friends have weekends off, making it the socially convenient thing to do. However, it may not be the primary reason. Many formally (full- and part-time) employed people do work weekends, and most developers in this study have freelance or self-employed partners, or are part of social circles made up of freelance and self-employed people.

the office: it must be calm with few distractions, with adequate time for breaks and taking care of an errand or two. In effect, the ideal 9-to-5 workday—the office job as it should be. Freelancers borrow the 9-to-5 time-map in order to subvert, reject, or remold, for their own uses, the offensive temporal and spatial logics and ethos that are attached to it within the spaces of formal, full-time employment.

Conclusion

A more careful study of how developers labour in and around various conflictual time-maps, rhythms and technologies of time might show how they produce the stop-gaps necessary for the circulation and accumulation of capital. This study has no pretensions to that, as it was limited in time and scope. However, using Bear's approach that attempts to discern nonhuman, social and personal experiences of time and how they interact with post-Fordist, contemporary time was useful to lay bare the fact that freelance developers are not "escapees" from an existing labour-capital relation, but they, too, are at the mercy of unpredictable rhythms, unrecognized and unremunerated labour, and the shirking of responsibilities once held by the state and the corporation.

The nonhuman rhythms of changing and depreciating bodies of code are difficult to capture, and justify, in invoices to clients, as the basis for remuneration is the end result of the programming activity outlined in the contract or package; developers do not charge for every false start, every revision or every foray into hundreds of lines of code to locate bugs. When they do press"start" on work timers, the wage-per-hour is affected by the rock-bottom pricing of gig work, or the need to make oneself more competitive in the marketplace. Further, freelancers engage in other types of work that they are not directly remunerated for, such as finding new clients and writing proposals. And, if they want to be a developer in the long term, the upkeep of personal code and websites throughout time, in addition to the continuous learning of new languages, frameworks and approaches will also fall into the un- or under-remunerated category.

However, freelancers consider this an acceptable price to pay to escape the confines of the typical 9-to-5 office job. The ability to set their own hours, goals, priorities and make decisions about how many meetings to attend or phone calls to make in a day takes the sting out of the experiences of not getting paid as well as one might have or the experience of programming a poorly-defined project for an indecisive client. The *freedom from* the formal, full-time 9-to-5 allows freelancers the *freedom to* recreate the 9-to-5 time-map—as one of the more salient and tenacious cultural representations of what work is and what it looks like, freelancers mirror this representation in order to achieve structured, daily rhythms on their own terms.

Freelance developers in this study have gone freelance to honour their own rhythms to work without pretense, or being subject to the farcical behaviour of bosses and companies; and to materialize, through their temporal and spatial practices, an attitude that some might call *post-wage* or even *post-work*. Being free from clocking in and out means that developers try to capture "being in the zone" more freely without fear of interruption, completing more deep work on projects that they have chosen for themselves, even *if* some projects are nightmarish. The phenomenological experience of creating code leads developers to reject the notion that one hour is equivalent to some stable quantity of work; like artists, writers or other craftspeople, productivity and creativity are often difficult to catch or rationalize. Thus, wages earned and hours put in are weakly associated with developers' sense of their productivity and the value of the end product.

Many of the developers in this study chose to go freelance to have more control over their work time as well as greater autonomy in work situations that directly affect their personal rhythms, desires and sense of work ethic. While autonomy, freedom and flexibility are value-laden terms important to the propagation of neoliberal labour regimes, making the systems of just-in-time production and flexible scheduling acceptable to many, the autonomy and flexibility of developers, as shown in this study, should neither brand them as apolitical puppets for neoliberal regimes nor mark the desire for autonomy and freedom as antithetical to political affect that promotes solidarity and collective action. Some critical scholars and labour activists, understandably, eschew such characteristics and values as unhelpful to the overall aims of progressive labour movements. Hopefully, the approaches taken in this project clear a path toward an understanding how the experiences of and desire for freedom from particular work experiences politicizes freelance subjects, creating subjectivities that resist some capitalist logics, such as Postone's blank-slate work hour and the surveillance embedded in the office space.

While mimicking the 9-to-5 structure in their daily work lives allows them to experience freedom and autonomy, it also, paradoxically, stabilizes the 9-to-5 such that it allows this time-map to continue mediating daily work life, sustaining it as a common experience. I have discovered temporal overlap among the participants, rather than ten unique temporal experiences, contrary to some suggestions by the literature and popularly-held ideas about our 24/7 world (Crary, 2013) that has "unhinged" shared social time (Virno, 2004, p. 103). This temporal overlap provides developers common points in the day during which that they might meet for coffee, a brainstorming session, a networking event or an evening pint after the day is done.

Chapter 4: Work Space and Place

This chapter details various aspects of the spatial experience of freelance development work. I wrote the interview guide for the second interview to inquire as comprehensively as I could about space and place: Does everyone have a home office? Why or why not? If so, do they struggle with separating work from home life? Do they work in cafes and which cafes do they consider the best to work in and why? If freelancers really can work from anywhere, why do they live and work in Halifax versus anywhere else? What are the advantages and disadvantages? What makes a good work space—what are the necessary elements? And, do they ever make work space decisions based on whether or not they will meet others there, either intentionally or unintentionally?

Contrary to an aspect of a study done by Liegl on freelance creative tech workers such as graphic designers (2014), developers, in general, are reluctant to work in public or semi-public spaces. Productive coding work requires quiet and little interruption. This need for quiet, though, seems to make it even more imperative to find ways to be social and meet with colleagues; participants talked about this imperative as meeting a basic human need. Web and software development too, I found, is "heavier" than some make it out to be; again, the image of the digital nomad carries with it assumptions that both the tech and its use are as airy and insubstantial as cloud computing. The nature of programming and development also affects how developers experience work in the home and how they feel about their work/life balances: the computing requirements necessary to write code means that developers do not generally program in bed on an iPad, for example. A constant presence of work was not something participants complained about.

I also found that Halifax contributes to the experience of both work space and work time, and interestingly, developers' subjectivity, in several respects. First, its relative distance from the major hubs of money and power has a tempering effect on the fast-paced project cycles that seem typical of those places (Shih, 2004). Halifax, as an urban place, has Goldilocks appeal: not too big and not too small. There are just enough cafes, just enough busyness and just enough people to get momentum around projects or ideas, according to participants. Lastly, Halifax contributes to local developer's subject formation. Like other mid-size cities competing for talent and money, local leadership tends to prioritize economic development that includes ICT innovation and rewards entrepreneurship in this field. Also, many participants described Halifax as having an identity of Maritime scrappiness through economic struggle over the years, which colours some leaders' focus on tech innovation and also some participants' motivation behind their own efforts at building collective projects. However, freelancers are often not considered for the seed money and resource support given by governmental agencies and venture capital. The freelance developers in this study expressed a complicated relationship with this state of affairs, affecting how they relate to the inner circle made up of tech entrepreneurs and venture capital, and how they position and articulate their own projects organized within the wider, freelance creative tech community.

Marking Boundaries and Crossing Thresholds

Solomon and I were leaning over his iPad at a table in Pavia Café on the fifth floor of Halifax's beautiful Central Library. The café, being on the uppermost floor, is all window, affording customers a panoramic view of the quieter streets behind Spring Garden Road. Gingerly avoiding some wet spots on our table, Solomon slid his tablet over for my viewing;

"You know, these are all pretty boring. It's basically the same thing all the time."

I looked through the collection of photos. For three days in the past week, he had worked from home. I saw three, nearly identical pictures of his home office: desk, two computer monitors, ergonomic mouse, keyboard and a small, potted cactus. His drawn window blinds occupy the photo's margins. We talked about preferences for desk knick-knacks and the ebb and flow of messiness and tidiness. In Tuesday's photo, his cat walked across the keyboard. Laughing over the cat's antics and the joy of pets, we moved on to the photos he took while working outside his home. He double tapped the screen to enlarge the first one.

"I took the photo from where I was sitting that day, right before I got to work."

An image of a laptop's screen dominates the photo. I peer into the photo's margins, attempting to discern which café he might be in. All I see beyond the laptop is a blurry assemblage of chairs, tables and a cold food display.

"It's just your screen!", I exclaim in surprise.

He laughs, "Well, that's my work!" Becoming pensive, he adds: "It's funny seeing these photos. They don't really capture everything that goes on in programming. It's hard, through a photo, to convey what it's actually like."

Some scholars of space and labor find the uber-mobility of freelance and telecommuter workers disconcerting. Ursula Huws points to the "fractured and footloose" nature of contemporary work, especially highlighting the work experiences of freelancers, telecommuters and independent contractors as a key problem in building strong, identity-based worker movements (Huws, 2014). The concept of time-space compression paints a picture of capitalist accumulation across the globe annihilating space through the speed of transaction, uprooting workers and communities into the immaterial flows and circuits of global capital (Harvey, 1989; Castells, 2000[1996]). Commonly, the freelance worker is drawn as a digital nomad, globe-trotting with his or her light-weight laptop in hand, turning work into an unending vacation (Liegl, 2014). Castells also argues that the *network society* "is a result of the disaggregation of labor." (2010, p. 302). Fractured, footloose, disaggregated — do workers meet in time and space any longer, and, if yes, what results from those proximities?

In recent decades, there has been much sociology and anthropology dedicated to exploring how and in what formations work has escaped from the corporate office into public and semi-public spaces, how this affects the experience of work and sociality at work, and how new spatial constellations of freelance creative work contribute to subject formation. For example, de Reuter, Cohen and Saraco argue that the new practice of co-working "softens effects of flexploitation, albeit in a manner that tends to deepen neoliberal subjectification" (2017, p. 701). Michael Liegl observes freelance creatives take advantage of their mobility as a resource for getting the job done; freelancers operate within a "spatio-technical" *dispositif*: an assemblage of places, ICT technologies and self-employment that gives way to an aesthetic-affective attachment to place that allows for self-reflection and self-interrogation. He says: "Packing up and changing to a new location is not only a way to rhythmically separate tasks but also to find inspiration" (2014, p. 179). These authors also point out a freelancer's need to fashion external discipline and monitoring by putting themselves in the presence of others. My findings largely echoed these authors' results, with some differences due to the specific nature of programming and what it means, materially and spatially, to deal in and with code.

As described in Chapter two, Wonda and Webster's mostly precedented, well-defined work allows them a higher degree of mobility, but the precedented work also means lower pay; Wonda and Webster have greater *local* mobility, but weekend trips to Paris are out of the question—they simply do not earn the incomes. The unprecedented and often ill-defined nature of Solomon's projects means that he is much more dependent on the stability and availability of spaces and work tools, like a whiteboard in addition to his computer. For Solomon, especially, being rich in "screen real estate", as he calls it, is essential to getting the job done.

Screen real estate in measured in inches. Most developers have a two-screen environment, either in their homes, or in a secondary work place location (or both). Generally speaking, one monitor displays the text editors in which developers write code and view other open files of code and reference material. The other screen displays the test site, which is what the end user will ultimately see. Usually these monitors are between 17 and 24 inches wide; some developers have one big monitor that stretches for thirty inches. For well-defined and precedented projects, the developer is much more able to comfortably rely on the use of only one screen. The developers with the most well-defined and precedented problems are, therefore, the most mobile, as they feel nearly as productive on one screen as on two and only need their laptop to work on the project.

> **Webster**: One of my projects takes up to six gigabytes of RAM, just to run the containers. My sweet spot is 15 inches of screen and 16 gigabytes of RAM.

> **Solomon**: If I'm working on a really well-defined problem, I can just go hide with my 12 inch screen somewhere, working purely with the code with no having to cross-reference everything. When I work on a really complex project that's a lot of stuff and a lot of deep thinking, I need an office, a monitor, a white board, and I need to be able to pace back and forth and figure things out. For that kind of work, I like to know I can be there for a long while.

Developers rely on their screen real estate, white boards, and steno pads to wrangle with assigning logic and sense to a client's often inchoate vision. The developers need these tools to help them see the solution, to track the logical statements that will ultimately produce buttons, plug-ins, drop-down menus, and so on. As participants described their work set-ups and their material needs, and as I compared those needs with how the developer fit into the production framework, it seemed that the relationship between the necessary spatio-material conditions and unprecedented projects is inverse: the less defined the project, the more physical space and material is required to tackle it. Some developers have even rented office space outside of the home to wrangle big, complicated projects. Emails, posting to social media, taking notes and phone calls can happen anywhere and require little "screen real estate". The inverse relationship between the nature of the project and the materials required for it suggest a greater gravity in programming and development work—developers are more anchored in space and material than we might think. It suggests place, material and space are highly relevant to their work.

Solomon has a very stringent attitude toward his work space. His programming work must be done at his desk that lives in the living room. In fact, his living room is partitioned, by the arrangement of furniture, into a work space and a leisure space. He moves across the room to take breaks in the leisure space. Neither leisure activity nor food is allowed to cross into the work space.

> **Solomon**: It's just so important to me. If this is just a place where I could start eating and watching YouTube it just wouldn't work for me.

However, Solomon's boundary rules allows some leakage of work into other nonwork realms in the home, with something like an iPad or mobile phone providing drips. His iPad allows work to happen at the kitchen counter and sporadically in the bedroom, but the hardware and software capacities of such a small device only allow for tasks such as correspondence or calendar checking. As Solomon's work requires access to his screens and files of code, it is difficult for it to follow him around the house, so leakage is generally controlled. Wonda might work in the bedroom, living room and kitchen all in one day, because her work includes a great deal of correspondence, writing and social media posting. Her work requires less screen real estate, so she works in more spaces in her home. Her coding, though, happens at her desk with the big monitors. Webster's experience is somewhere in the middle, as he can get away with use of a laptop for programming more frequently than can Solomon.

The use of smaller devices, such as laptops, iPads and mobile phones echoes what Henrikson and Tjora found in their studies of nomadic workers working in cafés. Pieces of technology become domesticated by the user, integrated into cultural spaces like the home or a café, where they are put to use to maintain control over the structure of their lives (2018, p. 354). Extrapolating from this analysis of their findings, I believe that the use of the smaller devices provides controlled passage of work into other nonwork areas of the home; developers expressed no difficulty in maintaining work-life separation. This might be aspirational rather than an actual experience. However, by marking off work space in the home for big tasks and deep work, and relegating certain shallower tasks to smaller, portable devices, developers are still able to feel that they can "leave" work. Possibly, in many analyses of work done in the home, scholars conflate the ubiquity of screens with the ubiquity of work. Certainly, many freelance and telecommute workers experience a general, oppressive ubiquity of work, but the nature of programming, marking boundaries and using pieces of technologies to control passage of work throughout the home, Wonda, Webster and Solomon are able to limit and control the "where" of work.

Wonda, Webster and Solomon all have fixed work stations but also exhibit nomadic practices; Solomon and Webster generally lean toward fixed work stations they have in their homes, though Wonda is so established in her business now that she almost exclusively works from home to have access to her files, home printer and a computer-desk-chair configuration that alleviates her lower back issues. In general, though, they all have a handful of locations in which they pursue work, and the day's specific tasks often dictate decisions to be in any one of them. I found a disciplined practice of marking off work and non-workspace, in addition to the use of movement and portable pieces of technology to create thresholds to cross, mostly in the pursuit of managing productivity, motivation and emotion.

Office Space

Often, the decision to stay home or work in a café has a lot to do with emotional states. Transitioning from one space to another allows the developer to capture, or recapture, productivity or motivation that either the location or the movement itself (and the time it takes) provides. Liegl captured this in his study of how freelance designers organized their spatial working lives. He argues that mobility is a resource for creativity, rather than a hurdle to cross, and that the decision from day-to-day (and hour-to-hour) on work space allows, too, for practices of self-interrogation. Freelance workers utilize movement as a resource to access creativity, while also engaging in self-observation that includes place and themselves as creatives inside of it, as central to answering the question "Am I productive here?" (2014, p. 169).

Certainly, like Liegl, I, too, discovered a "care of place" element among the developers. For developers, though, it is most of all productivity and motivation that they are after, rather than creativity. However, for Wonda, Webster and Solomon, making transitions across space, moving from home to cafe, was a way to subdue and manage emotions around work. Mobility, at least sometimes, prevents self-interrogation rather than produces it. More specifically, mobility often obscured interrogation of the self as a worker, doing hard things and struggling to reconcile conflicting needs and concerns. Solomon put it this way: "Because I *can* work anywhere, it makes it hard to figure out where I work best". Movement, mobility (even within the small space of the home) and marking boundaries and crossing thresholds was a way to manage and control emotions around work, especially those of worry, anguish, and anxiety.

Wonda: Usually I get a critical mass of things piling up on my desk,

> and then I have a breaking point, and today was one of those breaking points, and I clean it all up. There's a lot going on in my business, I'm out of control with some things, I lost some clients; a retainer told me that they were going to bring it in-house, so I'm going to lose that retainer. Cleaning and organizing usually comes on the heel of something...of a change. I think it comes down to a sense of control. I can't get all my clients today, or I can't get this code to work, but by golly, I make up my bed or file my invoices. It lets your brain just relax a bit.
>
> **Webster**: I tend to get distracted at home, mostly with video games, especially If I am working on something that I am struggling with. Then I'll leave the house 'cause that removes the video game thing. One avoidance mechanism.
>
> **Solomon**: Sometimes it's really tricky to get started. I've found it really difficult to get the right head-space sometimes. That's why working in bed in the mornings with the cat seems to work quite well. It's one of the things that's kept me from burning out too much — starting my day on my own terms.

Further, Liegl also notes that, for freelance creatives, being out-and-about not only generates creativity, but allows the worker to be seen working by others; freelance digital creatives suffer from an invisible production process and working publicly puts their production process on display—they can prove to the world that they are working. This particular behaviour and its motive was less apparent among freelancer developers, probably because programming and coding produce programs that do work in the "real world," and code, though invisible to some clients, is highly visible among the developer community. Code is a "social object" (Kitchin & Dodge, 2011, ch. 1) and developers have their own communities that recognize, not only the hard-work that goes into creating it, but also its objective qualities ("clean" and "elegant" code is the aspiration of all programmers). When Wonda, Webster and Solomon seek public places in which to work, it generally is not to be seen or to chance upon interaction: it is to be present within the sheer ambience of human life.

> **Wonda**: Working by yourself can do a number on your head; I'm an extrovert, it's really important for me to have people. So that's why the

> groups I belong to are really important: I get that human interaction there.

> **Webster**: When I go to a café or the library I don't really interact with other people, other than ordering tea from the barista. But I guess the possibility is there. It's more just the fact that there are people around. I don't like feeling secluded. It's weird: a lot of people would say I'm a bit of a loner, and I like alone time, but not 24/7. I like being around people. Working in a café...it's interaction, but without interaction. It's just presence.

> **Solomon**: It's nice to get out and see people. I go to the library a lot, just to see other faces, and people having other lives, because when you're alone all day, you get square-eyed—you start to forget what interacting with humans is like!

Because programming necessitates, for the most part, several hours of quiet at the home office, everyone emphasized how important it is to go out to "get that human interaction". Developers, for all of their dislike of the 9-to-5 office space, often expressed missing out on the fun of chatting with co-workers and the effervescence of working with others in the same field on the same problems. Though they dislike the spatial realities of the office—often noisy, full of interruptions, with annoying surveillance and norms of movement—missing out on some of the "water cooler" elements of the office job encourages developers to sacrifice some of their productivity by packing up their laptops and going to the library or a cafe. All of them stated that they have a practice of attending a group outing or a MeetUp at least once a week. Though the conflicting, various rhythms and exigencies of programming might mean that work plans need to be scrapped (remember Webster's accidental all-nighter chasing a bug), developers' commitment to the ideal day—replicating the 9-to-5 template—means that they are successful at finding opportunities to go out and mingle with their colleagues.

Halifax, Global Tech Hub

Volta, one of Halifax's co-working space and an "innovation lab" for local start-up companies, occupies two floors of a large, downtown building. Until my first interview with Webster, I had no idea where it was, despite being a frequent downtown pedestrian. There is no outdoor signage. The internet, of course, guided me there, and one day early in the afternoon I climbed a fair number of stairs to reach the entrance; the doors are set higher and slightly back from the sidewalk. Like walking up into a cathedral, I thought, as I wrenched open the doors.

Volta is a short walk across the lobby. The architecture of the building gives it an inscrutable, geometric shape, with the floor widening out as one moves toward the back of the room. The space is open, airy and bright. The tables, chairs and co-working booths support the airiness by their minimalism and straight lines. All of the chairs have wheels; the better, it seems, to capture the magic of spontaneous collaboration. The modern aesthete's favorite design element is present too: one wall is fully red brick. The young women at reception, relaxed in comfortable "Volta" t-shirts, look at you as if to say: "Good afternoon. What is *your* big idea?"

Webster and I agree to meet in the front, where a collection of leather chairs is assembled. I settle in there to wait, conscious that I have no "big idea" to share with anyone who may spontaneously approach me. A compact, pop-up café, occupying a corner just inside the doors, plays some garbled music out of a cheaply-made, blue-tooth speaker, which seems incongruous with the Silicon Valley-esque surroundings. This low-tech feature clashes with the general atmosphere that conveys the ambience of what I believe to characterize the workplace of the founder of Facebook: young tech workers, dressed in their friendly and self-consciously approachable jeans and hoodies, inhabited by a deep, slightly frightening drive to innovate, innovate, innovate.

Webster approaches me with a smile, wearing jeans and carrying his laptop, which is covered with stickers representing tech companies and programming frameworks, his only "marketing budget," he later tells me. He leads me to a small conference rooms he has booked for the next hour. It is brightly lit, with tall ceilings, but no larger than a big closet. We commence with the interview, and arrive at the topic of his current projects and clients. I ask him how he defines time management and task management, and how he tracks them.

"Well, time management I don't really have to worry about. I don't have deadlines."

I look up from my notes and ask, a little incredulous, "You don't have deadlines? Why not?"

"I guess I prefer clients that aren't super time-oriented. I mean, I had a client who wanted something done within a month, but it didn't feel like much of a deadline, as it was pretty easy to do. A lot of my clients don't really have the ability to see what's possible. What developers do is sort of a black box for people, so often clients have low expectations. Also, Halifax is sort of behind everyone else; there isn't a lot of front-end here. There's a lot more fun stuff going on in Vancouver or Toronto. If a company sets itself up in a certain way, like, if it doesn't have an API [Application Progamming Interface] you can't really build more complicated stuff. Some developers here are still working like it's ten years ago."

When I asked Wonda, Webster and Solomon about the advantages and disadvantages are of working as a developer in Halifax, they offered several reflections about how the city helped or hindered their progress, but they also offered their perspectives on the relationships between Silicon Valley, Toronto, Halifax and far-flung India, for example, and how it affected their local positions as developers. Their suggestions seem to align with Manuel Castells' theses about his *network society* which creates a *space of flows* as opposed to *the space of places.* The space of flows is the spatial logic of informational capitalism, while the space of places is the common experience of space, locally (Castells, 2000[1996]). Castells argues that ICT gives elite capital-holders the power to scatter this spatial logic throughout the globe, but a study of this logic proves that, instead, its spatial behaviour is characterized by both dispersion *and* concentration in a node-hub type pattern.

These elite networks, due to a combination of historical, geographical and social factors, create nodes and hubs of concentrated activity — the node contains the upper echelon of talent, skill and money, while hubs contain the lower rungs of the hierarchy, supporting and feeding the activity of the node through the arterial circulation of the network. Castell says,

> [o]n the one hand, advanced services [such as advertising, design, and R&D] have substantially increased their share in GNP in most countries, and they display the highest growth in employment and...investment rates in the leading metropolitan areas of the world. They are pervasive...On the other hand, there has been a spatial concentration of the upper tier of such activities in a few nodal centers of a few countries. (2010, p. 410).

Freelance tech workers, of course, are caught up within the flows of the web and software network; Silicon Valley is its terminal, or *technopole* (Castells, 2000[1996], p. 421). While developers in Halifax may not have all the programming fun (nor enjoy the salaries) that the Silicon Valley developers do, building all the new products for Apple, Microsoft, Google, or tech companies like them, they are protected from the excesses and exploitation of that place—protected from the "project time" and burnout cycles of Silicon Valley (Shih, 2004)—on this peninsula that juts out in the middle of the Atlantic. In Halifax, geographically distant from nodal cities, developers' work is not, as Wonda said during her first interview, as urgent or important as "open-heart surgery"—by which comment I deduced that the projects they work on for clients are usually peripheral to that business or do not directly address the pressing problems of the world (or what the *technopoles* deem the pressing problems of the world). Wonda and Solomon confirm Webster's observations about being a developer in Halifax, and they also note how the distance from node pressures influence pricing and pace:

> **Wonda**: Everything is slower here. Clients will take a week to get back to you about something. Sometimes I appreciate the quicker pace of my Toronto clients, 'cause clients back here would be like, "Well, let me think about it."

> **Solomon**: The gigs I've had have often been fairly price sensitive. People would be like, "This is really expensive", and it's like, well, I've been doing this for ten years, and this is a fraction of what I would cost in Toronto or if you were hiring somebody from an agency, or someone working in Silicon Valley.

Though Halifax and its tech workers are not shielded entirely from the excesses of the node—they still work for nodal companies while living in Halifax—local developers operate at a clear remove and it seems to diminish the excessive, urgent project time that characterizes the *technopole*. An urban place's position within the network as node or hub has a lot of influence on that locations' knowledge worker talent—their work time and client relationships. In his analysis of the *network society*, Castells argues that time is "timeless", and the *space of flows* contributes to the breaking down of common, social rhythms. Halifax may be hub to Toronto or Silicon Valley's node, but the developers' experience of time in Halifax in relation to these other places (in addition to their reference to the 9-to-5 time-map) may be better characterized as that of the outer ripples of a whirlpool. The water flows the fastest in the center, while the outer rings circle at a slower pace.

Hub Politics

As in many economically-middling cities in North America that have seen their traditional industries shrink, during the past several decades, Halifax's provincial leadership has emphasized internet and computer technologies as an integral part of strategy for local economic growth (Ivany, d'Entremont, Christmas, Fuller, & Bragg, 2014) and employment and as a way for the province to seem more attractive to both outside investors, and transient college students. Millions of dollars have been awarded to technological companies, or given as subsidies to organizations that support, or "seed", new tech startups. Volta, Halifax's "tech innovation hub" has been awarded $2.25 milllion over three years (Financial Post, 2017). According to the 2019 report by Halifax Partnership (an economic development organization), over half of Halifax's business startups use software or IT as a major part of their business models. "With more than 60 startup companies founded in the last five years, Halifax is an increasingly attractive destination for entrepreneurs looking to start and scale their companies." (Halifax Partnership, 2019, p. 43). In 2010, Halifax welcomed tech giant IBM as a heavy-weight employer— its "Client Innovation Centre" employs around 500 people (Financial Post, 2017).

Pro-ICT leadership, as elsewhere, is infatuated with startup culture that has produced the likes of Facebook and Uber. Halifax's current mayor, Mike Savage, recently opened up the first annual Atlantic Technology Summit on November 20th, 2019 with words of hope and encouragement for the city's ICT community, highlighting how important the city deems its entrepreneurial and creative tech workers.

However, freelance developers, at least those who participated in this study, are not eligible for that seed funding. It seems that many freelancers are uninterested, unwilling, or unable to meet the eligibility requirements which are influenced by venture capital's worship of the newest "disruptive" idea. Neither Wonda, Webster nor Solomon were interested in producing a billion-dollar idea, taking on employees, or jumping through bureaucratic hoops or meeting expectations of venture capital. Volta seems to mimic the behaviour and agenda-setting of tech elites in nodal cities.

Freelance developers operate in spaces like Volta in interesting ways: they may often benefit from this leadership bias and obsession toward new tech (with "lunch-and-learns" and access to free advice) but, as they are overlooked by venture capital, they can poke fun, be disappointed in and point out its flaws from the outer ring:

> **Wonda**: In my experience, it's those of us who are not in the startup culture who are like, "Thank you for your time, but we're actually keeping it [local]", or "We want to do all of our testing here" in Nova Scotia, or we want to ensure that we are giving back. But it feels like, in the eyes of the city, the tech is only good if you end up being a golden goose, and then what does a golden goose do? It goes to Toronto or Montreal.

> **Solomon**: The province is throwing money at tech, but they want a huge return on investment which you can only achieve through the start-up machine. The people at Volta are like: "When are you going to hire employees?", "When are you going to make your first $3 mil?" It's like: Do I have to? The startup founder is kind of like the labour for venture capital management and so they want them to burn out—the idea is that you have ten startups fail, but one of them go really large. They want to make sure it's all growth at any cost. But it hurts that company too, it distorts the whole economy. Startups are always such a weird thing.

Halifax and its leadership affect the way that Wonda, Webster and Solomon see their work and their own roles in the tech community. This indicates that the city plays an important role in forming worker subjectivity, though urban spaces are often seen as only the "where"—the geographical place—of political action. Studies that explore the experience of the urban citizen in neoliberalized cities often emphasize power relations (such as Castells' study of cities in the network society), or analyze how already politicized groups respond and react to, for example, neoliberal wealth and welfare reform policies or gentrifying urban planning practices (Hoffman, 2014; Guano, 2016).

Lisa Hoffman suggests that urban places are not simply the geographic location of politics but an integral part of how urban subjects are formed, and thence, how political communities articulate and move on their concerns. Influenced by Foucault, she argues that "...we must first understand the constitution of subjects before we may talk about how actors converge in a 'community of action' (Foucault, 1984b) that is made up of 'political actors'" (Hoffman, 2014, p. 1577). She describes how, during China's move toward a socialist market economy during the '80s and '90s, self-care practices that produced "dutiful socialist cadres" became problems in the face of the demands of the world capitalist market that required entrepreneurial, individualistic human capital. The social and the spatial, thus, promoted certain rationalities that re-shaped socialist subjectivities into entrepreneurial ones (p. 1581-1582).

According to discussion with participants and what I saw and heard throughout the research period, Halifax seems to promote similar rationalities in trying to position itself in certain ways to global markets and trends, and looks to its talented pool of tech workers and developers to help build a high-performing, economically-revived Halifax. But it presents a clear bias toward particular ways of materializing this image. Wonda, Webster and Solomon are wary of the embrace of Silicon Valley-esque startup culture, knowing that new, "disruptive" tech companies, in all likelihood, will disrupt themselves straight out of the province and into the welcoming arms of Toronto's nodal elite. In a small city like Halifax, the effects of that are tangible—a lot of money goes to support these companies in their infancy and then a lot of money leaves with them as they mature into larger markets.

Without Hoffman's arguments, it is too easy to see freelance developers as part of Richard Florida's *creative class* (2004), apolitical drivers of economic growth and gentrification or as disengaged from civic and party-defined politics (for example, Liberal, Conservative; environmental activism, etc.). Because Halifax's leadership wants to encourage technological entrepreneurialism, and because this desire is closely associated with Maritime identity of scrappiness and dreams of a Maritime economic resurgence, freelance developers' choices in the size and scale of their businesses and projects are a political response to the social and spatial rationalities that attempt to re-mold Halifax into a Maritime Silicon Valley.

I cannot offer precise detail of some of the developer's activities and projects that veer away from the rationalities and ethics surrounding the circle of venture capital and tech entrepreneurs in Halifax—doing so would endanger the anonymity of the participants. Generally, though, several participants are active members of groups whose purpose is to help local individuals deepen their technical and artistic skills, alongside self-marketing, branding and networking skills, to ease the path to self-employment and/or artistic fulfillment, while also aiming to form a long-lasting community out of these endeavors. Paraphrasing one local group's understanding of the importance of such groups, meeting up, in person, is key to *becoming* more creative and more productive. There is an understanding that community members contribute to the making of one another that was communicated to me during interviews. Other groups and local MeetUps also reflect variations on this theme. MeetUps especially geared toward freelancers and the self-employed stress the importance of being in a community to support one another's individual careers. Several other freelancers in this project chose the freelance life in large part to contribute to the health and vibrancy of such local groups, utilizing their own skills and technical know-how as volunteers, or using their freelance incomes to support pet projects that reflect the interests of their communities, rather than the interests of the start-up machine. No one in this group of participants expressed an interest in becoming "the labour for venture capital", as Solomon expressed it.

Because post-Fordism is characterized by capitalist accumulation across the social continuum of life and especially worker subjectivities, Autonomists and other theorists of immaterial labour argue that the epoch of post-Fordism contains within it both oppression and emancipation (Mäkinen, 2016); post-Fordist workers, similarly, may be equally ambivalent, displaying characteristics that could either augment or inhibit capitalist exploitation (Virno, 2004). Though Wonda, Webster and Solomon engage in practices and discourse that may not look all that different from those that occur within the Volta sphere of influence—they are entrepreneurial, work on their "brands", and treat every new contact or meeting as containing potential for future earnings or opportunities—the developers in this study expressed reasons for these activities that bend away from the goals and expectations of the Volta sphere: quick growth of profit-generating start-up companies who plan to scale up in production and clients with an eye toward "making it big" and moving Toronto.

The next chapter will review the relevant precarity literature and its influences to discuss what implications post-Fordism has for the political possibilities of contemporary, creative workers.

Chapter 5: Identifying the Political

Precarity scholar and anthropologist Kathleen Millar, in her ethnography of recyclables collectors who live on the garbage dumps of Rio de Janeiro, pushes back on scholars who deem the extremely poor as "apolitical", or only episodically political, when workplace gains come under threat (2018, p. 154). She argues that, by looking only for overt, antagonistic, and formally organized types of collective action, we miss out on the political nature of everyday, ordinary social behaviours. She says, in response to scholarly under-estimation of the ordinary:

> ...it would be difficult to see the 'ordinary' life of the dump—that stretch of time between instances of contentious collective action—as a world of fragmented, atomized, individual actions. Social projects like [partnered collecting] … were part of the fabric of everyday life on the dump, and when needed, they helped produce the relationships and collectivities that sustained more overt struggles. (p. 175)

Similarly, some scholars, in reaction to propositions arising from social movement theory, are keen to point out that the political arises from the habits and grooves of banal, everyday behavior. Political activity is not, in other words, a radical interruption or departure from everyday activities and behaviours. In their study of a group of immigrant women who engage in urban protest for immigration justice, Boudreau, Boucher and Liguori found that "[m]any of these women are politically engaged through the 'intensification of the everyday layout of life', pushed by what they live everyday...more than by a systematic reflection on domination (antagonism)." (2009, p. 340). The authors found that the women's engagement in the protest in a public square was buoyed by their previous experiences sharing frustrations on the bus ride to work that took them past the public square, daily. The conversational frustrations and familiarity with the square as a place intensified into political action—their presence at the immigration justice rally. Relatedly, Bureau and Corsani, during the course of their time amongst entrepreneurial workers that belong to one of France's business and employment cooperatives, found that "[a]utonomy is no longer an individual situation but a collective project..." (2018, p. 67). Though freelance developers work in very different contexts from those of Rio de Janeiro's garbage collectors, I take inspiration from Millar and others in trying to identify the political in the overlooked everyday, especially as the struggles of precarious labour may manifest in unanticipated ways.

Much like the analytical blind spot Millar points out among scholars about the political lives of the extremely poor, I wonder if there is a similar blind spot when scholars consider the political lives of freelance labour. They are presumed to be more likely to nourish a subjectivity and behavioural ethos that seeks synergy with capitalistic enterprise (Graziano, 2018; Gill & Pratt, 2008). Some scholars find that freelance work and self-employment in "creative capitalism" augment an individualistic attitude to work and nurture a sense of one's own creative uniqueness, making cross-occupational solidarity and organization difficult to imagine (Gill & Pratt, 2008). Many note how freelance material and physical working conditions (mobility, lack of office space, etc.) draw workers up and away from shared space and experience with other workers, compounding their relative isolation from labour struggle.

The rise of informalized work in the Global North, combined with ecological, political, and economic instability, produces, say some scholars, an ontological experience of precariousness. Precarité, or "precarity," a term that connotes generalized insecurity stemming from neoliberal labour policy, has, since the EuroMayDay 2001 demonstrations, served also as a political buzzword. Shortly thereafter, it appeared as an analytical term in the social sciences (Neilson & Rossiter, 2008; Butler, 2004; Lewchuk, 2017; Millar, 2017; Muehlebach, 2013; Alberti et al., 2018; Papadopoulos, 2018). As mentioned previously, freelance developers are thought of as belonging to this objective, sociological category of labour, and in qualitative research are found to share subjective experiences of precarious labour, such as worry, anxiety, frustration, and hyperactivity (Papadopoulos et al., 2008; Jokinen, 2016).

Scholars like Millar, involved in theoretically and empirically exploring what precarity means, have offered suggestions on how to think about work, labour struggle, class and political action; or rather, they have offered suggestions on how *not* to think about them.

As a tool to analyze the experience of contemporary labour and collective action, precarity is not as good at identifying class struggle—especially Fordist class struggle. For example, do garment factory workers and Milanese low-wage Fashion Week workers really form a class (Gill & Pratt, 2008)? Such a claim requires "collapsing the variations of precarity into some stable, undivided subject position" (Neilson & Rossiter, 2008, p. 65). Further, calling everyone "precarious" makes it difficult to analyze what is currently happening in the world of work and life (Millar, 2017). The least flattering critique of precarity as a way to analyze and predict near-future, antagonistic class struggle (cf. Standing, 2011) is that it represents the collective whining in the Global North over the contemporary failure of state-sponsored social welfare programs, coupled with an unexamined desire to uphold normative ideas of what work should look like, thereby ignoring how and for whom precarity has always been a reality (Millar, 2017; Munck, 2013).

However, as a way to examine where "precarious life and labour" (Millar, 2018, p.) intersect, it may illuminate not only how people are exploited and through what means, but also how people refuse, resist and imagine. Millar writes:

> …when approached as an open question about the relationship between forms of labor and fragile conditions of life, precarity retains both its

> analytical and political value. For some, precarity certainly describes an experience of loss. But for others, it might constitute a refusal of waged work, an alternative political subjectivity, or a mode of life that does not conform to liberal ideals. (p. 7)

Precarity scholarship is also focused on the breakdown of national sovereignty across the globe as multinational and surpranational corporations re-form into "vertical aggregates" of power (Papadopoulos et al., 2008); relatedly, the materialities and stabilities of citizenship and discrete political groupings are dissolving, too. These breakdowns form an axiom of one of Paolo Virno's propositions about post-Fordism (2004), that he uses to chart "cartographies of virtualities made possible by post-Fordism" of the multitude. The multitude is a political-theoretical label to describe "a mode of being of the many" without country, union or any particular sovereignty; the dynamics of the multitude contain "the debris which can sometimes jam the big machine" (23). Dependent upon certain arguments that make Autonomist Marxism different from classical Marxism about the nature of capitalism and the nature of labor and class, the multitude is ambivalent, amphibious and has no telos— the multitude are as likely to engage in struggle that contribute to capitalism's radical fracturing or in struggle that attempts to heal it over. This is sobering but, in fact, Autonomist belief that "[w]orkers are a class for themselves before being a class against capital" is hopeful. For, it is the working class that creates the forms and modes of life that capitalism then adopts for itself (11).

Ultimately, precarity as a conceptual frame problematizes work as the main, or only, (symbolic or actual) site of struggle, and the most visible types of antagonistic organizing (strikes, pickets, demonstrations) as the preeminent catalyst for balancing unequal labour relations. In short, our attempts to see new political classes and struggles clearly should not accompany the desire to contrive its "mode of being from some revolutionary essence." (Lotringer, 2004, p. 16) What if we hope for too much from mass worker action, while we overlook the power of quieter forms of resistance?

Though labour scholars and labour activists lament many of the new characteristics of contemporary labour, it is precisely because they do not align with Fordist practices and ethos that we must pay attention to them. Labourers like freelance tech workers cause furrowed brows because they exhibit individualism, entrepreneurialism, "footloose"-ness, and a lack of Ford-era political sensibilities. These are results of the processes of post-Fordism. Today's production processes are grounded in labor's human faculties—linguistic capacity, logic, reason, the ability to play, in addition to pure body power. In this way, the freelance tech worker seems apolitical, because its way of life is generally devoid of typical concepts, labels and categories that gained serious cache during the Ford era. As Paolo Virno states: "The intellectuality of the masses (another name for the multitude) is at the center of the post-Ford economy precisely because its mode of being completely avoids the concepts of political economy." (Virno, 2004, p. 108).

Political Possibilities of Freelance Tech

At a WordPress MeetUp at the beginning of data collection, I fell back at the close of the meeting to chat with Wonda. I was giving her the run down of my project, and sharing my musings about the contemporary freelance experience among developers. We got on the topic of the internet, activism and computer nerd culture. I shared with her my particular concern that online engagement with political issues does not translate to the real world in the way that we might believe that it does; and, importantly, I wanted to know how much the lure of tech and existence of digital social worlds brought people, especially those closest to it, out into the material world, the only place where collective capacity building can truly occur.

She nodded her head vigorously and shared an anecdote from her recent experience at a developer conference. She expressed surprise over the fact that the conference rooms where many of the designers (stereotypically more extroverted and socially successful) gathered were far more subdued, while the rooms with the developers were buzzing with introductions, chatter and gregarious interaction. She said, "I think it's because we're all open-source people. Working with and contributing to open-source code and software, we're used to sharing and peering over one another's shoulder into their screen to see what that person is up to. It's like, *here*." As she said "here", she made a gesture around her torso, mimicking a movement as if to pluck out one of her own organs and pass it along to someone else.

A week later, Wonda and I stumbled past one another at the Atlantic Technology Summit and decided to attend a panel together on an up-and-coming technology called blockchain, the last panel of the one-day summit. Leaving the auditorium, we made a plan to attend Third Wednesdays, a MeetUp for tech, media and marketing. We hop on the bus while chatting about the day's panels and, a short ride later, arrive at the Foggy Goggle, a bar in one of Halifax's trendier neighborhoods.

We climbed the stairs, threw our things behind a shabby couch in the corner and started to relax in the pub's cozy ambience. Wonda introduced me to the organizer and he asked if I would like to make an announcement about my project. I happily accepted. The other announcers included someone looking to recruit a developer for a project, another requesting user testers for a brand new app, and the last announces that she is seeking work and lists her skillsets. The organizer introduced us all with a respectful and buoyant attitude, enjoying making the crowd laugh during his intervals of speech. After announcements, pints in hand, attendees milled about, forming groups and chatting happily. It was obvious that many know each other well, and the atmosphere was one more of respected colleagues than stiff networking event; business cards fly around like bestowals of camaraderie and invitations to join the open-secret plots of tech and media nerds. I found myself standing around a high-top table talking with a few others about data security, Google's collection of personal data and how that affects everything: social life, economics and our internet personas that are fast becoming intertwined with official, governmental identities. The conversation starter described how her work is trying to solve some of these privacy issues. Someone later approached me to inquire if he is eligible for my study.

In any given week in Halifax, there is at least one MeetUp related to technology. Tech enthusiasts may attend meetings for WordPress and even for specific frameworks such as ReactJS or Angular; there is another MeetUp for those who identify as mobile workers and another for digital and visual commercial artists. New groups start up frequently and different groups cross-pollinate and recombine over time. As many say, it is easy to find yourself knowing the entire tech community in a small city like Halifax.

Freelance developers often call upon one another to work on a single project in a team environment that may last several days or several months. If one developer lacks certain skills, or is uninterested in developing one aspect of the project, he or she will contact his or her freelancer colleagues to fill the gap. Freelancers with too many contracts on their plates or irons in the fire willingly pass on projects and clients to those who need and want them. They may become ephemeral employers, hiring a developer for several hours to fix a bug, or become ephemeral employees, working under the employ of another developer.

When asked the question: "Are relationships important to your work?", every participant in this study said "Yes". A few talked about how, as a freelancer, one must be open and honest with loved ones about the nature of the work, and also, how important the support of loved ones is to manage the related stress. However, most of them responded in a way that intimated a sure knowledge about a fact of both precarious life and labour: that the freelance developer depends upon the creativity, contributions and opportunities opened up by others. All of them recognized this dependency as a condition of successful life and labour.

> **Wonda**: Relationships are incredibly important. Community, too. We all get stuck in our own ass; I just like to have peers to be like, "Am I crazy?", or "Could I get better at this?", or "How did you handle that?"
>
> **Webster**: Subcontractors are important too. I want to deliver what my clients want, but with someone I trust, who has the same ethics sense that I do; not right or wrong, just similar, so you get the job done. Being with people who are trying to build something and they want me to be involved and I want to be involved. Those are important relationships to maintain. I don't mean for selfish reasons, just, I hope that the relationships are for everyone's mutual benefit.
>
> **Solomon**: I'm still getting work from people related to one of my first projects going way back. I still keep in touch with people from the various places I've lived and various projects I've worked on. For my friends and acquaintances, I'll offer what I know and consult free of charge. Connections are so important.

Halifax's Mobile Workers Association (MWA), and its mobile co-working community project, seek to remake the typical networking event, from the inorganic and contrived exchange of business cards to on-going, relaxed hangouts, where people may get to know one another slowly over time. The MWA uses an online platform to organize pop-up meeting places on a near daily basis, and then the MeetUp structure to organize its networking hangouts. MWA's tagline is "Working together is always better". (Mobile Workers Association, 2020) MeetUps, co-working groups, organizers of workshops and panels use the spaces that the city provides: free-to-book meeting rooms in public libraries, cafes or bars with large spaces, and the conference rooms at Volta.

The freelance developer, designer, and visual artist communities put together panels on how best to charge clients, how best to market your skills, and how to protect yourself and your clients when navigating internet security, to name just a few topics. Several developers in this project commit some of their free time to helping others learn code or find development jobs and mentoring newly-minted freelancers. Most tech MeetUps start with announcements from individuals seeking a job, a candidate for a job, or information or advice. One could understand this activity and its content as a result of successful subjectivization by neoliberal regimes; however, if post-Fordism is a time where all the usual categories are emptied of their meanings, these activities could be equally ambivalent or transformative. Freelance life forms subjects that are "...mobile and detached, adaptable, curious, opportunistic and cynical, also toward institutions; they are inventive and share knowledge through communication and language; they are mostly de-politicized, also disobedient." (Lotringer, 2004) All of these practices and behaviours might be mobilized toward the end of organized political activity, either critical, radical or conservative.

Autonomy, or freedom, does not have to be a bad word when it comes to labour politics and worker ethos, I would argue. It is a discipline and mandate of post-Fordism, yes, but then takes the shape of skill, a way of being, and a way of being with others. The short-term partnerships and transient teamwork exhibited by freelancers seems similar to what Millar observed on the dumps of Rio de Janeiro in the ways collectors helped one another that she calls *relational autonomy*, the "relative degree of control over work and time [that] enables [collectors] to sustain relationships, fulfill social obligations, and pursue life projects in an uncertain everyday" (Millar, 2018, p. 71). Rather than viewing autonomy as a symptom of neoliberal governmentality, where the self is conceived of "existing prior to relationship" (Gershon, 2011 in Millar, 2018, p. 89), freelance developers display a disposition toward valorizing their autonomy for the ways that it allows them to follow their own physical and mental rhythms, take care of family and pets, participate in social groups, pursue hobbies and organize events and workshops for the community.

Of course, this benefits capital. Navigating freelance life using the 9-to-5 time-map offers freedom from that particular grind and set of constraints while also obscuring and distracting from other ones outside the bounds of that narrated, social time-space. However, the practices required to do freelance labor are also practices that help freelance labor manifest alternative ways of working and living; if any are nameable, one might say they set the "post-wage" example for others. If the freelancer's labour of time is the labour of making freelance work, then it is certain that at least some portion of Halifax's freelance developer community is in an extended collaborative project to help each other do that labour in the best way that each person can. Wonda, Webster and Solomon's spatial and temporal decisions in responding to their perceptions of the general demands of the world of work were just as important as an end as they were as a means to future opportunities for work and money. They place themselves in strategic positions so that they may benefit from the elitism of the technopole and the provincial government's desire for a tech utopia, while also harboring a fair amount of cynicism, and derision for what Silicon Valley represents.

The temporal and spatial practices I saw through this study give a greater idea of the needs of developers and how they try to meet those needs in and through time and space. If the political emerges from the organic, banal everyday, these practices may inform future praxis. The varied work place landscape (home, cafes and co-working sites); the experience of coding and the spatial boundaries it produces; and replicating the 9-to-5 time-map indicate, if not a degree of, but a disposition toward, sociospatial-temporal alignments and overlaps that do actually allow for concerted efforts, in real space-time, of collective practices for the benefits and care of the freelance tech community. Freelance possession of the 9-to-5 time-map prevents a reality that many fear: "unhinged social time" (Virno, 2004, p. 102) and "social arrhythmia" (Castells, 2000[1996], ch. 7).

> **Solomon**: Does my freelance life affect time for family and friends? Sometimes it does. This last project has been so crazy. But, other than that, it's been great. There aren't that many hours that I have to rearrange to make personal life stuff happen. Most of my friends are in the same sort of circles as I am, doing the same kind of stuff. We're all kind of flexible together.

Participants led me to realize that their freelance lives lead to more mutual support and cooperation rather than more isolation and individualism, as the literature leads us to believe. Some participants were quite aware of Silicon Valley logics that local political and business leaders attempt to recreate in Halifax and it is evident that many of their choices in their freelance lives are political ones, aimed at creating a different kind of tech entrepreneur, one that values community, small and slow business growth, and fulfillment of artistic and creative drives. Freelance developers can help us imagine what new forms of collective activity and solidarity that could emerge from this kind of labour organization.

Chapter 6: Conclusion

There is growing concern about the precarious nature of all types of work. Precarious conditions of labour seem to foreclose traditional forms of worker struggle and resistance. In addition, the perceived nature of creative work—individualistic and conforming to neoliberal ideals—prevents those concerned with sustaining labour struggle in the present and future from seeing how freelance creative labour resists neoliberal and capitalist norms, expectations and logics. My analysis of the temporal and spatial experiences of freelance developers—who seem to have escaped the typical confines of formal employment and who generally make a decent living—it shows that they neither experience total labour precariousness, but nor are they totally free from external constraints and controls. Though developers have autonomy from some typical frustrations of 9-to-5 work and the labour-capital relation that it represents, they enact labour—in addition to coding and programming—to mediate between the unpredictable nature of code and the variability of the internet; to resolve conflicts between a client's vision and the end result of the programming through stopgap measures and billing practices that do not always capture income and sometimes lose it. They also take on forms of labour that are necessary to make freelancing work, and for which they are not directly remunerated: networking, proposal writing and re-skilling, to name a few. Developers perceive these types of labour *of* time (Bear, 2016) as the price to pay for their autonomy.

The experience of contemporary work like creative tech is characterized overall by the disaggregation of and de-population of workers from shared work places; the onerous demands of neoliberal regimes to be self-sufficient in everything (at any and all times of the day); and the political ambivalence of the post-Fordist worker "multitude". However, the ways freelance developers in Halifax experience and perceive their work time and space and how they respond to those experiences indicate that these workers find ways to overcome the conditions of contemporary work to meet one another in time and space and often for the purposes of acting collectively to ameliorate some of the issues that come about from their labouring *of* time. Freelancing has set them free from the insufferable time and place that is the 9-to-5 corporate office job. They replicate this time-map to get their work done under their own terms, understanding their own agency in the process, which also allows for simultaneity with other freelancers.

Sometimes these temporal labouring practices require freelancers to rent out office space on a short-term basis to grapple with an unprecedented and poorly-defined project. Intangible though their work is, they are still tied to material: monitors, keyboards, whiteboards, cords, cables and mice. They are also tied to spaces and places, the movement among them necessary to manage emotions around work, both negative and positive. Developers often miss the watercooler elements of the office job and, after working for several hours alone in the home office, they make particular effort to socialize or network at least once a week, experiencing these times as essential to their well-being. The participants overwhelmingly talked about this in terms of a basic human requirement; it seems an imperative. While the MeetUps, group lunches and other meetings often have to do with freelancing or technology or both, they also are the grounds for a community that seems to exist to help one another make a success out of the freelance life and make successes out of projects that are deemed too small or not profitable by the venture capital start-up circle. The freelance community whose members embrace their autonomy also recognizes that autonomy needs company. For some freelancers, to be freelance in Halifax, Nova Scotia is to take part in the region's scrappiness and fighting spirit, while not "selling out" to a vision that belongs to Silicon Valley. The everyday lives of freelance developers seem to suggest possibilities for a concerted politics—in fact, in many ways they already are political.

Though I am confident in some of the patterns discovered in the analysis of the research data, there are limits to what can be definitely said about the temporal and spatial experiences of only ten freelance developers. If the participant pool were made up of more people who lived in the suburbs surrounding Halifax, reports on the topics of isolation, community and frequency of networking, for example, would have certainly been different. The suburbs are also where many immigrants settle with their families, and while those immigrants who participated in this study shared many aspects of the developer experience with the Canadian citizens, they have their own timelines and anxieties about their work and their futures that are tied to the processes of immigration. Additionally, a greater balance between men and women would have provided a different tone to the analyses, for too many reasons to list here. Not only do women earn less and have differing struggles around work, they seem to fulfill specific social roles within tech communities. This is only an inkling that I have from discussion with a couple of participants (and not a deep discussion at that), but women might carry more burdens when it comes to making the organized groups "go"; they might perform more of the hospitality, outreach and community-building duties than their male counterparts. Or it may be that women-led groups are more characterized by hospitality and community-building. This is a topic for further research.

Lastly, if a greater portion of my research time were spent in the MeetUps and other organized groups, I could have done a more thorough analysis of various parts of the political aspects: in what other ways developers' subjectivities are co-consituted by Halifax as a place within the space of flows governed by the rationalities of Silicon Valley and start-up culture; how MeetUps contrast with other organized groups in goals, attitudes and organization; and, because MeetUps have their own particular forms, what are their limitations as a vehicle for political activity—how do MeetUps, where freelancers can meet in place and time, compare with meeting in place and time at a co-working space? Finally, more time among the groups and the community would have afforded more detail of the modes and means of the everyday kinds of political activity among developers. I would have liked to have tried my hand at learning to code to achieve a much deeper insider perspective on it all.

Whatever seismic social, economic or political event may happen in the near future that could, in the words of E.P. Thompson, help freelancer workers "make themselves" into a class working for its own interests, their spatial and temporal praxes may not look like the Fordist ones. If politics emerges from the patterns of everyday life, a politicized freelance developer group might be doing all of the things it already is, except the practices will have taken on new meaning and purpose. As I write this conclusion, countries around the world are battling the social and economic effects of a new coronavirus and its disease, COVID-19; many millions in the United States and Canada have lost their jobs, and many other millions have taken their work home with them, as North Americans obey stay-at-home orders. Millions of us have become freelancers overnight. Judging by the opinion pieces and analysis done by recent journalism, it seems clear that a consensus is forming that the COVID-19 crisis has intensified the existing class, racialized, and other inequalities and along ways in which power has separated people from one another. I have wondered how this crisis has changed the everyday for the participants who have already developed work-at-home practices and how they might be experiencing some of their frustrations differently. I also wonder how the new work-from-homers are experiencing and perceiving their new state of affairs; what are they are noticing about their employment that they did not notice before?

For labour organizers, scholars and activists, I also hope this project helps to break open some of the reigning assumptions about the transformative possibilities of certain kinds of labour, and the kinds of temporal, spatial and affective conditions necessary for capacity building. Though freelance developers lead unique working lives, their experiences do have some overlap with other laboring groups, especially as workers everywhere turn more freelance and flexible.

References

Alberti, G., Bessa, I., Hardy, K., Trappman, V., & Umney, C. (2018). In, against and beyond precarity: Work in insecure times. *Work, Employment and Society*, *32*(3), 447–457.

Bakhtin, M. (1981). *The dialogical imagination: Four essays* (M. Holquist & C. Emerson, Eds.). Austin: University of Texas Press.

Bear, L. (2014). For labour: Ajeet's accident and the ethics of technological fixes in time. *Journal of the Royal Anthropological Institute*, *Supp*(S1), 71–88.

Bear, L. (2016). Time as technique. *Annual Review of Anthropology*, *45*, 487–502.

Bell, D. (1973). *The coming of post-industrial society: A venture in social forecasting.* New York, NY: Basic Books.

Berardi, F. (2005). What does cognitariat mean? Work, desire and depression. *Cultural Studies Review*, *11*(2), 57–63.

Berardi, F. (2009). *The soul at work: From alienation to autonomy.* Los Angelos, CA: Semiotext(e).

Financial Post. (2017, December 5). *Halifax tech incubator triples in size, touts track record.* Retrieved from https://business.financialpost.com/pmn/business-pmn/halifax-tech-incubator-volta-labs-triples-in-size-touts-track-record

Mobile Workers Association. (2020, March). *Homepage.* Retrieved from https://themwa.ca

Precarias a la Deriva. (2010, December). *Precarious lexicon.* Retrieved August 25, 2019, from https://caringlabor.wordpress.com/2010/12/14/precarias-a-la-deriva-precarious-lexicon/

Boudreau, J., Boucher, N., & Liguori, M. (2009). Taking the bus daily and demonstrating on Sunday: Reflections on the formation of political subjectivity in an urban world. *City*, *13*, 336–346.

Bureau, M., & Corsani, A. (2018). The French Business and Employment Cooperative: An autonomy factory? In A. Bove, A. Murgia, & E. Armano (Eds.), *Mapping precariousness, labour insecurity and uncertain livelihoods: Subjectivities and resistance* (pp. 60–69). London, England: Routledge.

Butler, J. (2004). *Precarious life: The powers of mourning and violence.* London, England: Verso.

Castells, M. (2000[1996]). *The rise of the network society.* Oxford, England: Blackwell.

Crary, J. (2013). *24/7: Late capitalism and the ends of sleep.* London, England: Verso.

Davies, K. (2003). Responsibility and daily life: Reflections over timespace. In J. May & N. Thrift (Eds.), *Timespace: Geographies of temporality* (pp. 133–148). London, England: Routledge.

De Peuter, G., Cohen, N., & Saraco, F. (2017). The ambivalence of coworking: On the politics of an emerging work practice. *European Journal of Cultural Studies*, *20*, 687–706.

Florida, R. (2004). *The rise of the creative class: And how it's transforming work, leisure, community and everyday life.* New York, NY: Basic Books.

Forsler, I., & Velkova, J. (2018). Efficient worker or reflective practitioner? Competing technical rationalities of media software tools. In P. Bilić, J. Primorac, & B. Valtysson (Eds.), *Technologies of labour and the politics of contradiction* (pp. 99–120). Cham, Switzerland: Palgrave Macmillan.

Furåker, B. (1984). Gorz's "Farewell to the working class". *Acta Sociologica*, *27*, 367–370.

Gell, A. (1992). *The anthropology of time: Cultural constructions of temporal maps and images.* Providence, RI: Berg.

Gershon, I. (2011). Neoliberal agency. *Current Anthropology*, *52*, 537–555.

Gill, R., & Pratt, A. (2008). In the social factory? Immaterial labour, precariousness and cultural work. *Theory, Culture & Society*, *25*, 1–30.

Glennie, P., & Thrift, N. (1996). Reworking E. P. Thompson's 'Time, work-discipline and industrial capitalism'. *Time & Society*, *5*, 275–299.

Gorz, A. (1982). *Farewell to the working class: An essay on post-industrial socialism.* London, England: Pluto Press.

Graziano, V. (2018). Fake it until you make it: Prefigurative practices and the extrospection of precarity. In E. Armano, A. Bove, & A. Murgia (Eds.), *Mapping precariousness, labour insecurity and uncertain livelihoods: Subjectivities and resistance* (pp. 157–169). London, England: Routledge.

Guano, E. (2016). *Creative urbanity: An Italian middle class in the shade of revitalization.* Philadelphia: University of Pennsylvania Press.

Halifax Partnership. (2019). *Halifax index 2019* (Tech. Rep.). Retrieved from https://halifaxpartnership.com/sites/default/uploads/pages/downloads/Halifax_Index_jun14-2019_INTERACTIVE.pdf

Hardt, M., & Negri, A. (2000). *Empire.* Cambridge, MA: Harvard University Press.

Harvey, D. (1989). *The condition of postmodernity.* Oxford, England: Blackwell.

Henrikson, I., & Tjora, A. (2018). Situational domestication and the origin of the cafe worker species. *Sociology*, *52*(2), 351–366.

Hodges, M. (2014). Immanent anthropology: A comparative study of 'process' in contemporary france. *Journal of the Royal Anthropological Institute*, *20*, 33–57.

Hoffman, L. (2014). The urban, politics and subject formation. *International Journal of Urban and Regional Research*, *38.5*, 1576–1588.

Huws, U. (2014). *Labor in the global digital economy: The cybertariat comes of age.* New York, NY: Monthly Review Press.

Ivany, R., d'Entremont, I., Christmas, D., Fuller, S., & Bragg, J. (2014, February). *Now or never: An urgent call to action for nova scotians* (Tech Rep). Halifax, NS: OneNS. Retrieved from https://onens.ca/img/now-or-never.pdf

Jessop, B. (2005). Gramsci as a spatial theorist. *Critical Review of International Social and Political Philosophy*, *8*(4), 421–437.

Jokinen, E. (2016). Precarious everyday agency. *European Journal of Cultural Studies*, *19*(1), 85–99.

Joseph, B. (2005). Interview with Paolo Virno. *Grey Room*, *21*, 26–37.

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

Kaoosji, S. (2018). Lessons learned from eight years of experimental organizing in Southern California's logistics sector. In J. Alimahomed-Wilson & I. Ness (Eds.), *Choke points: Logistics workers disrupting the global supply chain* (pp. 214–229). London, England: Pluto Press.

Kitchin, R., & Dodge, M. (2011). *Code/space: Software and everyday life.* Cambridge, MA: The MIT Press.

Lamb, C., & Rubinger, D. (2017, December). *Stacking up: A report on Canada's developer talent* (Tech Rep). Toronto, ON: Brookfield Institute. Retrieved from http://brookfieldinstitute.ca/wp-content/uploads/BrookfieldInstitute_StackingUp.pdf

Lazzarato, M. (1996). Immaterial labour. In M. Hardt & P. Virno (Eds.), *Radical thought in Italy: A potential politics* (pp. 133–147). Minneapolis: University of Minnesota Press.

Lefebvre, H. (1974). *The production of space* (D. Nicholson-Smith, Trans.). Cambridge, MA: Blackwell.

Lewchuk, W. (2017). Precarious jobs: Where are they, and how do they affect well-being? *The Economic and Labour Relations Review*, *28*, 402–419.

Liegl, M. (2014). Nomadicity and the care of place—on the aesthetic and affective organization of space in freelance creative work. *Computer Supported Cooperative Work (CSCW)*, *23*, 163–183.

Lotringer, S. (2004). We, the multitude [Foreword]. In P. Virno, *A grammar of the multitude: For an analysis of contemporary forms of life* (pp. 7–19). Los Angelos, CA: Semiotext(e).

Maclean, K. (2018). *Flexible work arrangements: Transforming the way Canadians work* (Tech. Rep.). Ottawa: The Conference Board of Canada.

Marx, K. (1992 [1985]). *Capital* (Vol. 2). England: Penguin.

McRobbie, A. (2014). *Be creative: Making a living in the new culture industries.* Cambridge, UK: Polity Press.

Millar, K. (2015). Reading twenty-first-century capitalism through the lens of E.P. Thompson. *Focaal—Journal of Global and Historical Anthropology*, *73*, 3–11.

Millar, K. (2017). Toward a critical politics of precarity. *Sociology Compass*, *11*(e12483), 1–11.

Millar, K. (2018). *Reclaiming the discarded: Life and labor on Rio's garbage dump.* Durham, NC: Duke University Press.

Muehlebach, A. (2013). On precariousness and the ethical imagination: The year 2012 in sociocultural anthropology. *American Anthropologist*, *115*(2), 297–311.

Muehlebach, A. (2017). The body of solidarity: Heritage, memory, and materiality in post-industrial Italy. *Comparative Studies in Society and History*, *59*(1), 96–126.

Munck, R. (2013). The precariat: A view from the south. *Third World Quarterly*, *34*, 747–762.

Munford, M. (2016, December 22). *How WordPress ate the internet in 2016... and the world in 2017.* Retrieved February 29, 2020, from https://www.forbes.com/sites/montymunford/2016/12/22/how-wordpress-ate-the-internet-in-2016-and-the-world-in-2017/#edb0ae9199d9

Mäkinen, K. (2016). Valuable selves: Potentiality and temporality in work-related coaching. *European Journal of Cultural Studies*, *19*, 69–84.

Neilson, B., & Rossiter, N. (2008). Precarity as a political concept, or, Fordism as exception. *Theory, Culture & Society*, *25*(7-8), 51–72.

Noble, B. (2016). *Articulating dinosaurs: A political anthropology.* Toronto, ON: University of Toronto Press.

Papadopoulos, D. (2018). The two endings of the precarious movement. In E. Armano, A. Bove, & A. Murgia (Eds.), *Mapping precariousness, labour insecurity and uncertain livelihoods: Subjectivities and resistance* (pp. 137–148). London, England: Routledge.

Papadopoulos, D., Stephenson, N., & Tsianos, V. (2008). *Escape routes: Control and subversion in the twenty-first century.* London, England: Pluto Press.

Peyton, G. (2018, July 4). Work out: A guide to coworking in Atlantic Canada. *Atlantic Business Magazine.* Retrieved March 10, 2020, from https://www.atlanticbusinessmagazine.net/article/work-out-a-guide-to-coworking-in-atlantic-canada/

Pitts, F. (2016). Rhythms of creativity and power in freelance creative work. In J. Webster & K. Randle (Eds.), *Virtual workers and the global labour market* (pp. 139–159). London: Palgrave Macmillan.

Postone, M. (1993). *Time, labour and social domination: A reinterpretation of Marx's critical theory.* England: Cambridge University Press.

Rosenberg, S. (2007). *Dreaming in code: Two dozen programmers, three years, 4,732 bugs, and one quest for transcendent software.* New York, NY: Crown Publishers.

Shih, J. (2004). Project time in Silicon Valley. *Qualitative Sociology*, *27*, 223–245.

Terrell, K. (2018, March 30). *Code your way to career growth.* Retrieved February 29, 2020, from https://www.aarp.org/work/career-change/info-2018/coding-boot-camp-fd.html

Thompson, E. P. (1963). *The making of the English working class.* New York, NY: Vintage Books.

Thompson, E. P. (1967). Time, work discipline and industrial capitalism. *Past & Present*, 56–97.

Tsing, A. (2015). *The mushroom at the end of the world: On the possibility of life in capitalist ruins.* New Jersey: Princeton University Press.

Virno, P. (2004). *A grammar of the multitude: For an analysis of contemporary forms of life* (I. Bertoletti, J. Cascaito, & A. Casson, Trans.). Los Angelos, CA: Semiotext(e).

Weeks, K. (2009). "Hours for what we will": Work, family, and the movement for shorter hours. *Feminist Studies*, *35*, 101–127.

Appendices

A: Research Design, Methods and Ethics

The available time for this project lasting only a short while, I could not achieve full ethnographic potential—I did not attend every MeetUp or group outing, nor did I learn how to code and strike out as a freelance web developer myself. However, I designed the project to achieve as much contact with the ten participants as possible through a variety of research methods. I sought to conduct a deep, qualitative study with the means and time available to me.

Recruitment was accomplished by three means: cold-call, online messaging through the career and social networking site LinkedIn; through snowball sampling through the first few initial contacts; and through participation and attendance at open-to-the-public tech events, mostly MeetUps, that occurred during the data collection period. All participants, ultimately, were recruited via online messaging and snowballing. Pursuant to my research question that aims to discover what collaborations and collectivities exist and how they are produced, I tried to gather new participants through snowballing as much as possible, in order to better see the contours of any groups or collectives that exist within the Haligonian tech community. This did indeed place me within a circuit of developers who know one another through the various channels of tech work and tech hobbyism, or by virtue of extended residence in Halifax. I was surprised to discover that four out of the ten were recruited via LinkedIn; however, upon reflection, I realized that this is perhaps not surprising at all, and that I was able to take advantage of a particular attitude of freelance workers: that one never knows what might come out of a coffee date, and that to say "no" to any networking opportunity is to foreclose future work or collaboration opportunities.

Two semi-structured interviews (60-90 minutes each) took place with each participant, one on the theme of time, the other on space. The two interviews served to give me more time with each participant, while also augmenting trust and rapport over the period of data collection during which I repeatedly asked participants to complete something for the study—I sent invites via email (five times to each participant) to respond to the reflection entry prompt (described below) and sent daily reminders for nearly a week before the second interview for the photo journal activity.

The time in between interviews allowed the participants to engage in the photo journal activity: for five consecutive days before the scheduled second interview, participants received text reminders to take photos of their work spaces and work places. The reminder texts read: “Please take a photo(s) of the places where you work today (at home or otherwise) and the spaces that you work in.” Most participants prepared a photo journal for the second interview, and those were used as a catalyst for conversation about work spaces, choice of places, environments, and working as a freelancer in Halifax, Nova Scotia. This interview took place around 10-14 days after the first for most participants. Only two were unable to complete the journal; most took at least two pictures, and three took five or more pictures. For those who did not complete the journal, they were asked to reflect retrospectively on work space and place.

In addition to the photo journal, participants were asked to submit five “reflection entries” on the topic of their ideal day as a freelancer. I told participants that they would randomly receive five invites to respond to this prompt: “Please reflect, in any style you choose, on how today compares with the ideal day that you described in our first interview together. You may write as much as you like, but please write a minimum of 150 words.” I kept a physical calendar at my work space to record which days and times each participant received an invite. I made sure to send invites randomly on different days of the week and according to the participants' timing preferences they indicated on the consent forms (between eight and ten a.m., for example). If a participant ignored the invites several times in a row, I checked in with the participant to ask if they would like to stop receiving invites. These activities added depth and texture to the responses elicited during the interviews, as they provided the participant with different tools to consider, critically reflect upon, and share their working experiences.

The reflections were collected via the *Opinio* survey software hosted by Dalhousie. Invitations to reflect were sent randomly to participants' email inboxes throughout the entire data collection period, which spanned September 2019 to January 2020. For those participants recruited early on (about four), this meant that I was able to chart their responses across many weeks. This meant that I could get a sense of how the ideal is or is not mostly achieved on average. Even for those who came late in the data collection period, though, the entries conveyed tension with the participant's first-blush description of the ideal day in the first interview. The entries met my expectations for their usefulness: a general description of an ideal day might tend to mask the ways freelance days are often not ideal; by asking the participant to reflect on the past 24 hours, those tensions and deviations from ideal surfaced more easily. I received 28 responses in total, with four participants submitting all five requested entries. Some participants ignored the request for 150 words and submitted smaller, "Tweet"-sized blurbs. Others used the tool to journal quite descriptively and with detail.

I also held several interviews with other informants—local people in the industry, organizers of MeetUps (groups organized through the online platform MeetUp.com), and developers who are formally employed—to glean as much context as possible. I attended the first annual Atlantic Technology Summit, a one-day event held on

November 20, 2019, to highlight and celebrate the advance of ICT and its applications in Nova Scotia. These informants provided anecdotal information on time and project management in web and software development, the peculiarities of the industry, and the province's view on the place of ICT in Maritime strategic and economic planning.

Whenever a participant mentioned a group she or he belonged to, I attended at least once. While present at these meetings, I made all attendees aware of who I was and what my aims were. Interestingly, the open-to-the-public groups and meetings are comprised of people who know each other quite well so the meetings felt somewhat intimate; though they were used to outsiders coming in, such as recruiters for development companies, so my presence was not perceived as strange. Though I did not record quotes verbatim from any person attending these groups, I kept some field notes recording general observations and topics of conversation.

Further, I initiated each new interview with an explanation of the research process and how I was going to keep their identities anonymous in the write up of the results. I explained my intention to create "composite characters" that would represent the experiences of several people, so that readers would not easily be able to identify participants. Also knowing that some participants and I would be in social contexts together, I asked the participants how they would like their relationship to my research to be identified to others, should the topic arise in conversation. When this did occur, participants were unbothered by naming themselves as study participants and describing their participation in the project. I was also careful, as the community in Halifax is small, not to make the participants known to one another unless they divulged that to one another by their own volition and then told me that they had done so.

Also, I sent each completed interview transcript to the participants to give them the opportunity to review what they shared and check for accuracy. Some expressed gratitude for the project, saying that it gave them opportunities to reflect on their work and other topics that came up during the interviews.

B: First Interview Guide and Schedule

This will be a semi-structured interview, thus, not all of these questions will be asked; they serve as a guide only, to allow for emergent themes to arise. The questions may not be asked in this exact order or in their entirety: conversation will be allowed to follow the interviewee's lead. This is in keeping with Article 10.5 of the TCPS 2 (2014) regarding the emergent nature of qualitative fieldwork. Questions that will be asked of every participant are highlighted in bold.

Introduction

1. Establish rapport – small talk, make the interviewee comfortable.

2. Brief statement of introduction by the interviewer – reiterate name, that I'm an MA student in Social Anthropology at Dalhousie, and that I'm conducting a study on the working conditions of freelance developers/designers. Include detailed information about all research methods and what full participation looks like. In addition to the two interviews, the research project would be enriched from the participant's engagement in reflection journal entries and a photo journal. Explain what those are, and indicate that I will request participation in those activities at the end of the interview, and the participant may decide to opt in at that time.

3. Explanation of the purpose of the interview – to understand the participant's experience of time in their work and personal lives.

4. Explain their rights as a participant – review consent form, guarantee confidentiality, request permission to audio record.

5. Have consent form signed before continuing. This is the point at which audio recording will commence if consented to.

1. Personal work history

a. **Would you please briefly describe your work history?**

b. **Would you please describe, in as much detail as you like, your current work life, including, but not limited to: how many current clients or projects you have; what kinds of skills and knowledge you need to execute your work; etc.?**

c. **Would you return to non-freelance work if granted an opportunity? Or, if you have always held a freelance position, would you ever consider working for a company or agency?**

2. Time management

a. **Do you use a time-management app or tool? Can you lead me through how it works?**

b. **What does it not account for?**

c. How do you assign time for the various tasks involved in your work?

d. How much of your work that goes into a project is accounted for in clients' invoices, or in your time-management apps?

e. **What are the differences between time-management, task-management, and project-management? Do you track all three?**

f. **When is a project complete? How do you know?**

3. Tech world

a. Explain your professional expertise in the knowledges required for your job. How much time did/do you spend on each?

b. **How would you describe the life cycles of IT/hardware/software innovation/change? How does that affect your work?**

c. Do you keep up with news inside the "tech world"? Why or why not? Can you describe the trends that you notice? What does the future look like? What does the future of freelancing, as a web developer, look like?

d. **Are relationships important to your work? How?**

4. Billing a. How do you decide which worktimes to apply to an invoice and which work time not to include?

b. What should your work time be used for, when you're working on a project for clients?

c. What should your time be used for, if you are working on other projects? Why?

d. What are all the activities that you include within the category of "work"?

e. **How do you allocate time to non-work areas of life—family, social, hobbies, leisure, etc?**

f. **Do you have workday rituals?**

g. How often do you revisit your resume, CV or online profiles, such as LinkedIn, to update or make changes?

h. **Have you ever sought collaboration or help from other people? Can you describe?**

5. **Would you describe, in as much detail as possible, your ideal work day as a freelancer?**

Conclusion

At the end of this interview, ask the participant if s/he would like to engage in the journal activities. Remind them that not participating will not exclude them from the study, and that they may opt out of the journaling activities at any time. Answer any questions from the participant. Present consent documents for each journal activity if participant wishes to engage.

Offer gratitude for their time and remind them of researcher's contact information if they have any questions or concerns.

C: Second Interview Guide and Schedule

Before this interview begins, participants will have taken photos of all the work spaces they inhabited for the past 5 workdays. The photos will serve as catalysts for discussion.

This will be a semi-structured interview, thus, not all of these questions will be asked; they serve as a guide only, to allow for emergent themes to arise. The questions may not be asked in this exact order or in their entirety: conversation will be allowed to follow the interviewee's lead. This is in keeping with Article 10.5 of the TCPS 2 (2014) regarding the emergent nature of qualitative fieldwork. Questions that will be asked of every participant are highlighted in bold.

Introduction

1. Establish rapport – small talk, make the interviewee comfortable.

2. Explanation of the purpose of this second interview – to understand the participant's experience of space in their work and personal lives.

3. Explain their rights as a participant – review consent form, guarantee confidentiality, request permission to audio record.

4. Have consent form signed before continuing. This is the point at which audio recording will commence if consented to.

Questions:

For each photo representing a workplace:

a. **Tell me about this place. Why did you choose to work here?**

b. **How long did you work here?**

c. Are there particular elements of this photo that were important for you to capture?

d. (If photographs are of a home-based worksite): How often do you work from home? Is there a difference between working from home and working elsewhere? How do you decide to work from home on any given day?

Concluding photojournal question: Compare your experience of work in this place versus that place. Does your sense of time change? Sense of effort? Sense of the overall quality of your work (both quality of experience and quality of work performance)?

General questions to ask when participant has shared all photos contained in the journal:

1. **Please tell me about your experience during Hurricane Dorian. Did you lose power? Was your work interrupted?**

2. **What are the advantages of being a freelance developer/designer in Halifax? What are the disadvantages?**

2. **How do you decide when and if to work in specific places?**

3. (If participant has not photographed any home-based worksite): Do you ever work from home? Why or why not?

4. **How do your workspaces inform what type of tasks you work on, if at all?**

5. **What are the benefits of working in particular types of places (elucidate each type, depending on those shown in photos: library, coffee shop, etc.)? What are the drawbacks of each?**

6. Do you have a favorite workplace? Why?

7. **What are the most important characteristics of a "good" workplace?**

8. Are certain places/spaces important to your work?

9. **Do you ever select workplaces for the possibility for interaction**

with or the presence of other people?

10. Would you prefer to have a space of your own outside of the home in order to conduct work? Have you ever considered joining a co-working space? (Explain the concept if participant is unfamiliar)

11. **What was the experience of taking photos of your workplaces like? Can you offer some reflections on the experience? Was there anything that stood out to you?**

12. Is there anything you'd like to add?

Conclusion

Offer gratitude for their time and remind them of researcher's contact information if they have any questions or concerns.