2

THEORIZING CRITIQUE AND RECUPERATION

Starting from the observation that "hacking has been hacked," in the following pages we develop an interpretative framework for studying the dialectical reversal of hacking. The core of our theoretical argument is borrowed from Luc Boltanski and Eve Chiapello's book *The New Spirit of Capitalism*. With them, we take as our point of departure the idea that capitalism advances by incorporating the critiques directed against it. Critique, once recuperated, is transformed into a source of innovation and legitimacy for capitalism (Boltanski and Chiapello 2005). It goes without saying that this is a two-way process. At the same time as criticism (and the critics) are incorporated into dominant structures, those structures are also transformed by the critique. A dialectic between critique and recuperation unfolds.

Boltanski and Chiapello developed a theory about the general, historical logic of capitalism. The dynamic that they describe in general and abstract terms is apparent in the empirical field of hacking. Silicon Valley owes its existence to a computer underground that was spawned from the American counterculture of the 1960s (Barbrook and Cameron 1996). A number of historians of technology have filled in the details about how cyberculture and counterculture merged and furnished the nascent computer industry with communication standards, working practices, consumer preferences, and so on (Zandbergen 2011; Turner 2006; Liu 2004). Contemporary

examples of the same kind of symbiotic relationship between the computer underground and various branches of the computer industry are easy to come by, as we will discuss in more depth in the empirical chapters (see also Delfanti and Söderberg 2018).

In addition to the work of Boltanski and Chiapello, our interpretative framework also draws upon a number of supplementary theoretical resources. Primary among these are social movement studies, autonomist Marxism, and labor process theory. Social movement studies, especially where it intersects with Science and Technology Studies (STS), offers a rich source for reflecting upon how activist groups achieve their demands at the price of becoming part of the institutional arrangement that they railed against. The contribution from autonomist Marxism is twofold. Firstly, it has advanced an interpretation of technological change as the outcome of capitalist restructuring in response to class struggle (Dyer-Witheford 2015). Secondly, it has theorized the "social factory" (Tronti 1979) or the "factory without walls" (Negri 1989). The basic idea is that the factory—and with it, the contractual employment relation—no longer delimits the site of capitalist value production. Scholars in new media studies are putting empirical flesh on these theoretical bones by describing how media companies have become structurally dependent on exploiting the free labor of fans, audiences, users, and so on (Gill and Pratt 2008; Fuchs 2014). Hackers fit neatly into this same line of argument. In the tradition of labor process theory, finally, we find empirically grounded insights into how management and workers wrestle over who is in control of the workplace and the paramount role of skills and technological design in determining the outcome of those struggles (Böhm and Land 2012; Söderberg 2019).

Based on a synthesis of these theoretical resources, we propose an interpretative framework for conducting fine-grained, empirical studies on the dynamic of critique and recuperation within hacker culture. Hacker projects brim over with tension and strife: alleged violations of licensing terms, bickering over the correct names to call things, endless negotiations over what software tools are proper, and so on. These skirmishes are not isolated, random events. They make up a larger pattern of struggle over the conditions under which hacker culture may enter into relations of mutual dependency with industry and government agencies. Through those struggles is determined the relative degree of functional autonomy

of individual hacker projects—and of hacker culture at large—vis-à-vis an exteriority.

By the word "autonomy," following an established tradition in political philosophy, we mean the capacity of a collective to determine its own goals and rules of conduct and then to remain true to those goals and rules no matter what the cost. A group of hackers enjoys a high degree of autonomy when they can determine the methods and purposes of their collaborative endeavors and future collective existence. The idea of autonomy is not falsified by the observation that hackers derive a livelihood and gain political leverage from entering into symbiotic relationships with industry, government, and academia. With our interpretative framework, however, attention is directed toward the risks and benefits of such alliances for the autonomy of the collective.

When autonomy is weakened, hackers can more readily be subsumed under an "open innovation" model. They are thus turned into a steady source of blue-sky ideas and problem-solving work. In saying this, we intend to turn the tables on the vast literature about "open" or "user" innovation (Chesbrough 2003; von Hippel 2005). In our reading of the situation, innovation is what happens when hackers fail to resist recuperation attempts. Indeed, the open model for procuring innovations from hackers (as well as users, fans, etc.) is the spirit of contemporary capitalism, to reconnect to Boltanski and Chiapello's terminology: it is how production is organized outside the factory walls.

In order to discuss the recuperation of critique with greater precision, we propose a number of analytical distinctions in this chapter. Firstly, we introduce three time horizons, or temporalities, within which the evolving relationship of mutual dependency between hackers and industry may be studied. Secondly, we survey the factors required for putting up an effective resistance to recuperation within the recuperative logic of history, which we elect to call the "three pillars of autonomy." These pillars are: technical skills, historical memory, and shared values. Thirdly, we propose a typology to talk about a budding social division of labor outside of formal employment relations. These are: communities of peer producers (such as hackers), crowds of users and audiences, and clouds of click workers. The refinement of this division of labor is one of the things to which recuperated hacker projects are contributing: for instance, by developing the

tools used by management to oversee a decentralized workforce. "Commodification" refers to the process of enclosing commonly held resources and inserting them into market circulation. The antonym of commodification is "communization." Resources nominally owned by the employer are rerouted by the employees to serve as common infrastructure and to support the reproduction of a shared culture.

These conceptual tools cannot stand on their own. Their analytical worth must be proven through empirical case studies, such as the ones we expound in the following chapters, in which we present four historical cases in depth. The point of this exercise is to alert us to likely developments ahead, and so we conclude the book with an anticipation of the upcoming circuit of struggle.

WHO COUNTS AS A "HACKER"?

Before we can start the theoretical discussion proper, we need to clarify the words and definitions that we will be drawing upon, starting with the key word, "hacker." Different uses of this word are circulating in the academic literature, often bearing witness to the disciplinary fields from which those studies derive. The different strategies for defining hackers can be grouped into four general types. In empirically oriented fields, scholars typically allow the people calling themselves "hackers" to define the word. If a scholar creates a definition based on some theoretical and disciplinary background knowledge, then hacking tends to be assigned a meaning stemming from studies of youth subcultures, social movements, or class analysis. We discuss the strengths and weaknesses of each approach in turn.

Definitions based on self-appointed hackers usually take their basis in an excerpt from the Jargon File, a widely recognized lexicon of hacker slang. The first entry for "hacker" reads: "A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary" (Steele and Raymond 1996, "hacker"). Three more entries follow, stressing the hacker's aptitude for programming. In addition, some general characteristics expected of an individual claiming to be a hacker are described, such as enthusiasm, curiosity, and the like. This offers a point of departure for studies of hackers, but it does not allow the scholar to advance

very far beyond what the practitioners already profess to know. Theoretical shallowness is a common weakness in many, otherwise empirically solid, exposés of hackers (Benkler 2006; Moody 2001; Weber 2004). This approach is problematic when the self-representations of the hackers are so faithfully conveyed by the academics that the practitioners' exclusions and omissions are reproduced and stamped with scholarly insignia. A case in point is the dismissive description of the "cracker" in the Jargon File, as someone who "breaks security on a system" (Steele and Raymond 1996, "cracker"). While free software development is associated with positive values, such as information sharing and transparency, this does not tell the whole story about a subculture that is just as much about secrecy and stealth. The latter aspects are strategically left out in the accounts given by many hackers, who are preoccupied with improving their tarnished public image.

A definition of hackers that is more loosely tied to their self-representations can be found in the field of cultural studies. Here, hackers are interpreted as one youth subculture among many. This perspective, put forward by Douglas Thomas (2002), has a lot to offer to our discussion. After all, subcultures are all about defining who belongs to the group and who does not. Of particular interest for our purposes is the fact that Thomas foregrounds how the hacker milieu differs from most other subcultures. The identity of a hacker is bound up with a practice rather than a style. He argues that this endows hackers with a greater amount of self-determination in relation to external influences than style-based subcultures, which are more easily swayed by commercial forces. In line with the cultural studies tradition, however, he understands this in terms of a generational struggle (Thomas 2002). The blind spot of the cultural studies field is political economy. Thomas's definition fails to take the measure of the successive integration of hackers into professional life as they come of age. Such a definition excludes the main topic of our investigation from the start, which is how hacker culture negotiates the perils and opportunities of subsumption under the computer industry's schemes of value production (Lakhani and Wolf 2005).

Social movement studies is more attentive to the political stakes involved in hacking. Another advantage of this approach—of which Paul Taylor and Tim Jordan are the chief proponents—is that it aims to describe how

hackers constitute themselves as politically conscious subjects capable of collective action. This outlook makes a lot of sense when studying how hackers mobilize against intellectual property laws, state surveillance, and the like. The approach is fitting for studies of WikiLeaks, Anonymous, and similar political groups with strong ties to hackers and a strong support base in hacker culture more broadly. There are many other aspects of hackerdom, however, that will be missed if it is studied using an interpretative grid borrowed from social movement studies. The stereotypical hacker does not perceive him- or herself as a political activist and is more likely to vehemently deny such comparisons. This is a problem for social movement theory, which stresses raised consciousness and articulation by its actors. One risks losing sight of what makes this setting special, what Gabriela Coleman once named the "political agnosticism" of hackers (Coleman 2004; Coleman 2012, 187–189). If pride of place is given to a relatively small group of activists who fit the bill, to the detriment of the rank-and-file, free software programmer, something important has been missed.

A fourth approach to the study of hackers draws on the tradition of political economy and class analysis. One common variant that we want to avoid is to declare that a segment of the population constitutes a new class of hackers in its own right, on a par with the working class (Wark 2004; for a critique, see Barbrook 2006). Such claims, made in disregard of what self-described hackers say or do, bring no analytical clarity to the discussion. More promising in our opinion is a class analysis that starts out from labor process theory and empirically oriented workplace studies. In the tradition of labor process theory, starting with Harry Braverman's classic *Labour and Monopoly Capital*, there was always an astute awareness of the need to keep the definition of the working class as dynamic as the everchanging labor process. This warrants our analytical procedure of including free software developers, hackers, and related kinds of hobbyists as a subset within the occupational structure of waged programmers and engineers. We acknowledge that the status of being a nonemployee is foundational to their identities, and so our categorization is gainsaid by subjective testimonies. That being said, empirical support is readily found in a computer industry that has grown to be structurally dependent upon extracting value from their activities (Kirkpatrick 2018; Liu 2004). The large discrepancy between, on the one hand, hackers' identity of being nonworkers

and, on the other, the objective role that they fulfill within capitalist circuits of accumulation is what warrants our inquiry into the dynamics of critique and recuperation. Thus, we offer an interpretative framework for empirical studies of the processes whereby hackers are turned into sources of revenue and innovation for the industry, or resist such attempts.

Then, who are we talking about when we discuss "hackers"? The answer to this question is further complicated by the fact that the phenomenon of hacking is as fluid as the technology to which it relates. Hence, any definition of the hacker must not be too closely linked to the use of any single device or infrastructure. In the early days of hacker culture, tinkering with hardware and interfering with telephone networks were central features (Levy 1984; Lapsley and Wozniak 2013). In the 1990s and early 2000s, to do hacking became synonymous with writing software code and meddling with network protocols. The practice was often inscribed in the then very popular master narrative about a "coming of the information/network society" (following Castells 1996). Subsequently, the confinement of hacking to a realm of "bits" was positively affirmed and contrasted against the old, industrial "world of atoms." This dichotomy is not salient today. A definition of hacking that excludes tinkering with hardware and interventions in biology would miss out on much of what is going on in hacker spaces today (Seravalli 2012; Kostakis, Niaros, and Giotitsas 2015; Delfanti 2013).

Conversely, however, the specificity of the hacker vis-à-vis other subcultures and movements would be lost if all references to technical practices were abandoned. An example of this is when artists involved in what was once called "culture jamming" claim to be doing "semiotic hacking." We therefore insist on the connection to technical practices. Without it, we cannot make sense of the meritocratic values within hacker culture. Being skilled is key when hackers distinguish themselves from ordinary computer users, variously labeled as lamers, n00bs, AOLers, and so on. Furthermore, hacking cannot be extended to include just any technology whatsoever, as this would make the word "hacker" synonymous with "tinkerer" and "inventor" or "hobbyist." There must be a connection on the symbolic plane, however remote, with practices relating to information processing and information security.

Open hardware development qualifies, for instance, because it borrows its methodologies, licenses, values, and cultural tropes from the established

tradition of free software development and computer hacking (Ackermann 2009; Powell 2012; Söderberg and Daoud 2012; Lindtner, Hertz, and Dourish 2014). Continuing along the same lines, the main argument of Delfanti's monograph on biohackers is exactly that "[o]pen biology is embracing values and practices taken from the world of hacking and free software [so that] science is experiencing the same type of differentiation and complexity shown by hacker cultures" (2013, 12). We concur with his observation, that with each subsequent wave of objects to which hacking is applied, the milieu seems to be becoming ever more integrated into the capitalist system. Hardware development is ultimately reliant upon a global supply chain, which might explain why hacker culture has come to hybridize with the shanzhai innovation ecosystem—likewise with DIY biology, which has developed in tandem with spinoff companies eyeing up venture capital and seeking certification from the authorities (Keane and Zhao 2012; Tocchetti and Aguiton 2015). The title of an article written by Delfanti shortly after the first dedicated biohacking conferences took place and the first community spaces for biohackers became established puts the question upfront: "Is do-it-yourself biology being co-opted by institutions?" (2014). Each time there has been a shift in the center of gravity within hacker culture, from phone phreaking and hacking to free software development, and from software to hardware and biology, a shared body of cultural tropes and values has been passed on, although, we venture to say, in an increasingly attenuated form. Thus, it is possible to grasp the elusive subject of hacker culture as a historical formation rather than as a purely theoretically motivated conception. To summarize, we base our analytical distinctions on the genealogy of collective representations among the people who identify with the "hacker" moniker.

This pronouncement begs the question: On what analytical level do we intend our distinctions to be applicable? We are obviously not content with simply referring to the individuals who call themselves "hackers." We also analyze the distinct historical trajectories and group dynamics of various separate collectives. Our musings about functional autonomy hinge on how the collective entity has been defined in the first place. We make use of a variety of designations: "hacker culture," "free software movement" and "open hardware movement," "hacker projects" and "hacker communities," and "computer underground" and "geek public." The

different names carry different connotations and bring into play different analytical levels. We draw upon different terms depending on what aspect of hacking we want to emphasize at particular junctures in our argument. For instance, when we discuss a common identity that encompasses the full spectrum of phenomena associated with hackers, we use the phrase "hacker culture." Under its umbrella gather past and present instances of hacking, including phone phreaking, the warez scene, free software and open hardware development, hackerspaces, DIY bio, and so on. When the emphasis of the argument is instead on the contradistinction between them and the surrounding society, we refer to the "computer underground" or "hacker subculture." The variation in language use is motivated, as we argued in the paragraphs above, by the need to explain hacking from a combination of disciplinary and theoretical perspectives.

Our use of the terms "hacker project" and "hacker community" warrants some additional comments. When saying "hacker project," we are referring to a delimited subgroup within the larger hacker culture who are dedicated to the development of a single software or hardware artifact. A hacker project is closely associated with a recognized project leader, common engineering goals that have been defined in advance, a single license under which the output is published, a dedicated discussion forum or a cluster of such forums and websites, and a core set of developers and beta testers. When we want to bring the developers and users into focus, rather than the artifact and the development process, we say "hacker community." The word "community" is loaded with many associations and conceptual ambiguities. For lack of a better word, we use it to name the loose constellation of developers and users who frequent a discussion forum, know each other from before, and jointly care about the future of the development project in question.

Occasionally, we compare communities of hackers with gatherings of amateurs, fans, hobbyists, and so on. The latter coalesce around similar kinds of generative practices as hackers, but without identifying themselves with the overall hacker culture. The emergence of such practices in many different sectors of society has been intensely debated under the heading "commons-based peer production" (Benkler 2006, chap. 3; Rigi 2012). Keeping to the established terminology in the literature, we refer to a peer production community as a generic, catch-all term that is not

beholden to the subjective identity constructions of the practitioners. As we move on in this chapter to unpack our theoretical arguments, however, we will propose a tripartite typology for making distinctions within commons-based peer production. This typology is: "communities of peer producers," "crowds of users/audiences," and "clouds of click workers." All of these categories designate abodes of production located outside of the formal employment contract and the factory gates. For similar typologies inspired by different theoretical traditions, see Jenkins, Ford, and Green (2013) or Shirky (2008). In keeping with the overall, interpretative framework of this book, we argue that meaningful distinctions can be made in the degree of autonomy (or lack thereof) that each group formation exercises over its own collective being in relation to external interests (i.e., state and capital). The classification of communities, crowds, and clouds as different constellations of value-producing activities corresponds to the emerging trend toward a division of labor outside of the wage labor relation.

Finally, with the term "hacker movement," we indicate a broader sweep of hacker projects and hacker communities that are bound together by common goals and formal support structures. This claim can be illustrated with the free software movement. A wide range of hacker projects, dedicated to the development of a single free software application or operating system distribution (such as, for instance, the Debian project), unite under this movement. They share the same software tools, content management systems, and licenses, go to the same conferences, and so on. This shared technical and organizational infrastructure is duplicated in shared concerns and discussion topics. The free software movement is broader in scope than any individual, free software development project, while at the same time being more restricted and bounded than the reference to a "hacker culture," as the latter must also include open hardware, DIY biology, and so on.

As a final remark, we note that our conceptualization of hacking as a collective entity, operative at different analytical levels, comes close to the approach of Christopher Kelty. Acknowledging the heterogeneity of the phenomenon, he also makes a compelling case that scholars need to refer to the commonality of attributes that fall under the umbrella of "hacking." He then draws a parallel with the eighteenth-century notion of the

"public." Just as the public in those days defined itself in opposition to absolutism, the notion of being a counterweight to the powers that be looms large in the self-presentations and identity constructions of hackers. This is a crucial point, because, although the concept of a public is sufficiently vague to include a range of phenomena, it is coherent enough to allow for collective action. The common identity of hackers across innumerable variations and differences is verified by the fact that, from time to time, they can come together and act in concert. Kelty's reliance on collective action as the defining trait of hackers resonates with the approach that we are adopting in this book.

From Kelty's work, we have also borrowed the idea that the public acts "recursively" to create the material conditions upon which its own, continued existence depends. The eighteenth-century public emerged in close association with the technologies of its time, chiefly coffee houses, book printing, and newspapers. Likewise, hackers rely on free software, transparent and compatible network standards, open hardware, and so on. In contesting intellectual property rights and promoting open infrastructures, hackers are not solely striving to realize their vision of a good society. They are concurrently safeguarding the legal and technical conditions for their own continued existence as a collective of peer producers. This fits neatly with our primary concern in this book, to describe how hackers resist recuperation attempts and maintain a relative degree of autonomy vis-à-vis an exteriority upon which they are nevertheless dependent.

THREE TIME HORIZONS OF CRITIQUE AND RECUPERATION

The coupling of "critique and recuperation" describes a dynamic wave of historical change propelled by struggle. Critique can take many different forms, but in the context of this book we are chiefly referring to critical engineering practices (Oliver, Savičić, and Vasiliev 2011). Through such practices, dreams about a radically different society become embedded in alternative product designs and diverging pathways in the development of technology. Critique, thus understood, spurs capitalism to evolve in new directions. By transforming itself, capitalism absorbs the disturbance and turns it into a material and organizational infrastructure for

the continued accumulation of capital. So that we may talk about this dynamic with more precision, we proceed to distinguish three time horizons within which recuperation processes unfold.

The shortest time horizon encompasses the life cycle of an individual technology development project and its associated community of developers. In the case of open-source desktop 3D printers (discussed in chapter 4), for instance, the starting point was the launch of the RepRap project in 2004. This project drew to a close in the early 2010s, when leading corporate players consolidated their control over the niche consumer market for desktop 3D printers. The vision that underpinned the development project in the early days, concisely captured in the project's byline, "wealth without money," had conclusively been recuperated when derivatives of the RepRap 3D printer started to circulate as commodities.

The second time horizon spans a landscape of hacker projects and communities evolving in concert with an associated branch of the industry. A case in point is the duration of the free software movement, from its inception in the mid-1980s until today. Although the free software movement has not come to an end, it can be meaningfully delimited from its successor, a social movement dedicated to the development of open hardware. Whereas the free software movement is closely associated with the computer industry, the open hardware movement is evolving in tandem with products and industry standards stemming from the consumer electronics industry. Recuperation processes operating within this time horizon shape the framing conditions for individual hacker projects. They tilt the balance of forces in upcoming, local struggles against recuperation attempts.

The third time horizon relates the phenomenon of hacking to epochal transitions in capitalism as an evolving whole. Hacker culture is at one and the same time a historical product of the capitalist economy and a—tiny but strategically placed—contributor to the further development of this economic system. We speak at this level of abstraction when, for example, we relate the birth of the computer underground in the 1960s and 1970s to the transition within capitalism from Fordism to post-Fordism.

These analytical distinctions are needed because the dynamic of critique and recuperation plays out simultaneously across many different geographical scales and time horizons. Recuperation attempts within a local hacker project advance in a piecemeal, iterative fashion. The microstruggles over

an alleged license violation or an ideologically informed design choice, for example, add up to shape capitalism as a whole. Short-term dynamics of action and change acquire their meaning within a wider frame of reference corresponding to the succession of capital's accumulation regimes. Recuperation attempts can be meaningfully resisted in a local setting—as, for instance, when hackers refuse to adopt one or another commodified and black-boxed industry standard. The catch is, however, that the significance of this refusal is conditioned by forces that surpass the local setting. Only with reference to the grander scale of things can judgments be made about whether or not to use a particular consumer product or choose one development fork over another.

Furthermore, recuperation does not exclusively impact upon the subjugated hacker project. Once a project has been recuperated, it furnishes capital with product innovations and ideas for organizational reforms that can then be exported to other sectors of the economy. Given the global and social division of labor, it is only to be expected that the most coercive side of recuperation is not experienced by those who are most directly concerned with it (i.e., the subjugated hacker community). The lowest tier of the labor market bears the brunt of recuperated hacker projects. Two examples will suffice to make the point: the digital platforms for dividing up and distributing piece rate work in the "cloud" and the new methods for surveilling and disciplining workers based on digital rating systems. In order for us to catch a glimpse of the geographically dispersed and protracted fallout from recuperation, we must not restrict our investigation to one-off case studies of technology projects. Hacking needs to be studied within an interpretative framework that can be scaled up to encompass capitalism as an evolving whole.

By historicizing the framing conditions of individual hacker projects, we arrive at the conclusion that the outsider position to which hackers often lay claim is always already inside the larger whole of historically developed, capitalist relations. Putting it differently, hacking is (partially) recuperated from the outset. The notion of a free-floating subject position located "outside" the social totality of capital is illusory. Connected to this ideological notion of the outsider is another, equally problematic, idea. Namely, that a vortex of disruptive innovations will flow from the hacker to uproot the constituted order and tear down incumbent interests. We

contend that this Schumpeterian fantasy works like a trap for capturing value. The disruptiveness of hacking has been anticipated by the open innovation model.

Something else that is ruled out by our historical approach is the notion of a pristine, golden age of hacking that at some point was forsaken. We endorse the commonplace notion that the past is an ex-post construction. Even so, we cannot choose to not posit a point of origin that has culminated in our present mode of existence. We need "usable pasts" as a baseline for making comparisons with the present, to pass normative judgments and take directions toward a more desirable future (Kelty 2008, 64–66). We reject the shallow wisdom of nominalism—whether it comes in the form of commonsense empiricism or as convoluted poststructuralism—and the unqualified celebration of contingency that is common to both. Often this outlook is an overreaction by those who previously longed for an absolute foundation, only to become disappointed and sink into despair. The right lesson to be drawn from the argument about contingency is that, although history is all that we have to hold onto, this is sufficient for the purpose of guiding collective action. In saying this, we endorse a key insight in Hegelian philosophy and in its modern-day heir, the immanent critique tradition (Antonio 1981). Although we do not dwell on these references in what follows, for the record, we declare this to be the philosophical backbone of our observations about the dynamic between critique and recuperation.

FIRST TIME HORIZON: RECUPERATION OF A SINGLE HACKER PROJECT/COMMUNITY

The first time horizon within which we can observe the dynamics of critique and recuperation in hacker culture is that of the life cycle of an individual development project and its concomitant community of developers. Recuperation attempts operating within this time horizon strive to enclose product innovations and other tangible goods stemming from the common development process. This corresponds to the "enclosure of the commons" idea, which is widely diffused among practitioners. One-off enclosures add up and work together as step-by-step advancements by the industry to subsume the hacker project in question under an open innovation model. Thus, hackers are turned into a steadier and less risk-inducing

supply of disruptive innovations and piecemeal development work. When we investigate hacking at this analytical level, we conduct case studies of the life cycle of a single technology. The temporal aspect is key, because it is in the displacements that occur from start to finish in the testimonies coming from the project's participants that we detect signs of a recuperation process.

The last point accords with the insight in STS that an adequate study of a technology should pay as much attention to failures and dead ends as to successes. It will then transpire that the stated goals of a project, the individual motives for being involved, and the design itself, have all metamorphosed during the course of the project's life cycle (Edgerton 2008; van Oost, Verhaegh, and Oudshoorn 2009). This point is warranted by the many success stories about start-up companies and innovations that fill page upon page of airport and management literature, as well as being the default narrative in some academic fields, such as innovation studies. These success stories are narrated with the commercial breakthrough as their referential cutoff point. What happens to the community of developers and users who incubated the innovation in the first place becomes unimportant after the product has been brought to market. This narrative contributes to the further marginalization of diverging perspectives and alternative trajectories of the technology that pointed away from the pressures of commercialization and mass production.

For the kind of inquiry that we have in mind here, perspectives from STS can be productively synthesized with social movement studies. The latter field is well acquainted with studying the dilemma faced by a social movement when it tries to have its claims and grievances institutionalized without losing itself in the process. Such a theoretical synthesis has been proposed by David Hess (2005). He coined the concept of technology- and product-oriented movements (TPMs) to describe civil society mobilizations that try to bring about social change through advocating alternative technologies and industry standards. Unlike many activist milieus, these are enmeshed in the corporate world. TPMs often have to accommodate some degree of co-optation by industrial actors for their political goals to be realized.

At the outset, it is commonplace for the goals of TPMs to be articulated in liaison with start-up firms, allies in the industry, or branches of

government, whose interests happen to coincide with those of the group in question. The symbiotic relationship with these external actors brings in resources, gives credibility to the cause, and opens up networks for disseminating the message or the product. In return, the social movement provides a niche market for the company's products, as well as being a pool for beta testing, market research, and so on. As Fred Turner (2006) shows, hacker culture was founded on such an alliance, which he called "legitimation exchange." The hippie movement and the nascent information technology industry cross-fertilized. The personal computer is an offspring of the "small is beautiful" philosophy of the hippies.

Gradually, as firms systematize their interactions with social movement actors, the design and meaning of the alternative technology is disfigured. New design considerations are gradually introduced into the development process, corresponding to a realignment of the group's priorities with market demands and mass-production standards. Following on from this, the transformation gives rise to what Hess (2005) terms "object conflicts" between the TPM and its for-profit allies, and between different factions within the TPM. Conflicts revolve around the proper design and/or adoption of a transformed technology, as measured against the original grievances and values claimed by (a fraction of) the movement. External actors may weigh in behind one or another side in the internal dispute.

Arguably, the free software movement could be considered a specific case of the more general TPM phenomenon. This synthesis of perspectives from STS and social movement studies offers a terminology and a point of orientation for studying the microprocesses of resistance to recuperation within hacker projects and hacker communities. What is missing from such a theoretical synthesis, however, is a vocabulary for speaking about the systematic character of these struggles. This systematic character is due to the industry's structural dependency on hackers as an external source of innovation and value production. Hence, perspectives from labor process theory and workplace studies need to be included in the discussion.

In a branch of media studies oriented toward a political economy style of analysis, it has long been debated whether or not fan fiction writers and television audiences should be considered producers of value for the culture industry. Several attempts have been made over the years to extend the labor theory of value to include these sites of value production (Smythe

1977; Fuchs 2015a, 2015b). Alongside fans and audiences, the same kind of argument can easily be extended to more technically oriented subcultures of hobbyists, DIY-tinkerers and "citizen scientists.". The commercial breakthrough of the GNU/Linux operating system in the late 1990s and early 2000s generated much enthusiasm because it suggested the feasibility of organizing programming work with a minimum of managerial control, with the programmers self-allocating tasks among themselves. In addition to spawning numerous empirical studies of the phenomenon, it inspired conceptually driven inquiries into an emerging mode of commons-based "peer production" (Benkler 2006; Rigi 2012). Free software development showcased the failure of capital on its home turf, in organizing production in the most technically advanced sectors of the economy. Waged and hierarchical labor relations proved to be inferior to a production process based on voluntary and nonmonetized contributions from peers, not only in moral terms, but also in terms of technical efficiency (Dafermos 2012).

The dark side of this promise, which was quickly pointed out by critics of the literature on peer production, was that firms could now tap into the unpaid, collective labor of free software developers. Pressure was thus put on wages and working conditions of employed computer programmers (Terranova 2000), who are now also called the "programming proletariat" (Jordan 2016). That this warning was warranted has since been amply demonstrated by the surge in what is euphemistically known as the "sharing economy." A thin line separates the utopian promise of overturning capitalist relations of production on the one hand and the intensification of commodified and exploitative relations on the other.

The academic debate on peer production versus free labor revolves around the question: Which one of the two scenarios will prevail? Instead of attempting to settle this question as though it was an either/or proposition, we offer for consideration a theoretical synthesis drawing on the aforementioned work on TPMs. On which side a particular hacker project will end up must be decided on a case-by-case basis, depending on whether the hackers can detect and resist recuperation attempts in the local setting. To the hacker community, defeat means that its autonomy is curtailed and its collective labor is subsumed under a value-extracting, open innovation model. To everyone else, the subjugation of the individual hacker community will be experienced (if at all) as an intensified commodification of

everyday life, an intensified exploitation at work, and an entrenchment of the division of labor outside of the contractual employment relation.

Using this interpretative framework, we can make sense of the regularly observed outbursts of strife and contention within hacker communities and, furthermore, explain why those outbursts tend to coincide with the crystallization of a potentially marketable product. Accusations and counteraccusations about breaches of the free/open license give a clear signal that recuperation is underway within a hacker project. The textbook example of this is when software published under a free or open license is copied by an entrepreneur or a firm and locked away behind traditional intellectual property rights or a black-boxed hardware casing (Brodkin 2016). Enclosure of the information commons by legal and/or technical means in open confrontation with the norms of the hacker community constitutes a highly visible form of direct recuperation. For the very same reason, it is often met with fierce resistance from hackers.

Inevitably, a moral register will be drawn upon by practitioners, who will accuse each other of hijacking, betrayal, and so forth. Naming and shaming are an indispensable tool in guarding the information commons from hostile encroachments. It goes without saying that an analyst drawing on the interpretative framework that we are proposing here will not be a neutral observer of events. We do not believe, however, that the predictive and analytical effort is helped by the analyst becoming an advocate siding with one or another camp. This would misconstrue the contradictions and constraints under which people subsist under capitalism. What is more, it puts the analyst at risk of being hoodwinked along with their chosen camp of practitioners, in case they are captured and subsumed under an open model of capital accumulation.

This last point is crucial, because the concept of recuperation is not exhausted by overt attempts at hijacking projects and communities. More insidious are the processes of indirect recuperation that work "behind the backs" of hackers, molding the goals and values of their community to make a better fit with the needs of the industry. Through such slow-working processes, the ground is prepared for the enclosure of the project's output in the future. This can happen without anyone even noticing, since the norms of the community are not being challenged head on. Rather, the local setting is carried away by a landslide that displaces a whole

movement of hacker projects. It takes a longitudinal and comparative analysis to register such historical forces at work. The present-day project with its stated aims can be held up against the standards of its own, earlier self. Hence our insistence above on the necessity of studying the entire life cycle of a development project. Some turning points in the trajectory of a project that could warrant analytical interest include: changes in the licensing terms, terminological shifts, challenges to project leadership, and forks in the development stream.

In the case of the open-source desktop 3D printer called RepRap, mentioned above, and to be discussed in chapter 4, the growing influence exerted by the consumer market over the development process meant that waning efforts were being devoted to designing the parts for the machine in such a way that another, similar machine could print them. Coupled with the engineering goal of building a self-reproducing machine was a utopian promise, as expressed in the motto of the project: "wealth without money." The twisting of the development process around a different set of engineering trade-offs reflected a transformation in the very purpose of the project. When this happens, there are often some in the community who remain loyal to the original vision and sound the alarm. However, the losing side in an internal dispute tends to quit the community, after which the traces of there ever having been discord are obliterated and forgotten. Therefore, just as much attention must be brought to the absence of conflict as to its presence. Saying this is only to repeat a time-honored insight from the study of ideology construction and the engineering of consent.

Processes of recuperation are difficult to detect because they preserve the content but alter the form of whatever it is that has been recuperated. To illustrate this rather abstract claim, we offer for consideration the example of hackathons. This is the name given to gatherings of programmers during which they shut themselves away for several nights and days to do spurts of coding. Hackathons reflect the intense and withdrawn lifestyle of the original hacker culture (Levy 1984, chap. 4). In recent years, however, hackathons have metamorphosed into a method for brainstorming start-up ideas and for overcoming hurdles in commercial project development (Irani 2015c). Even though the practices remain exactly the same as before, the overall purpose and meaning of a corporate-initiated hackathon are very different from those of a self-organized one (D'Ignazio et al. 2016).

The lessons drawn from the case of hackathons resonate with the historical emergence of the factory as the key site of production during industrialization. Karl Marx teased out the analytical distinction between the formal and the real subsumption of labor under capital from the development of factories. Formal subsumption corresponds to an early phase in the industrial revolution when the putting-out system predominated. Merchants provided workers with raw materials and bought back the refined products at a discount. Crucially, they left it to the workers to choose their own production methods. The introduction of the factory system did not immediately deprive workers of this discretion. To begin with, the factory was just an empty building within which the workers gathered. Only gradually was the internal composition of the labor process dissolved and reorganized at the behest of capital (as documented in further detail by Thompson 1963). The human worker was transformed into a mere appendix of the factory machine, in Marx's iconic expression. In order for this machine to operate smoothly, its unruly, human cogs had to be pacified through (despotic or scientific) management.

Hacker communities are caught up in the same whirlpool of forces as factory workers once were. Hackers clash with industry (upon which they nevertheless depend as a source of income, as a provider of critical services, products, and infrastructure, and in order to gain political leverage) over the relative degree of autonomy that they enjoy in defining common goals and choosing the proper methods for developing a technology. Defeat means that the hacker community is annexed to a firm's open innovation model. Methods for asserting managerial control over the external source of value production are introduced step by step by corporations. Thus, the hacker project is turned into a more reliable site of value production for capital. Innovation is the outcome of failure to resist a recuperation attempt.

SECOND TIME HORIZON: RECUPERATION OF THE HACKER CULTURE

The second time horizon within which we can observe the dynamics of critique and recuperation encompasses a range of different hacker projects united under a single movement that evolves in tandem with a whole

branch of the industry. If the direct counterpart to the hacker project is the entrepreneur and the firm, the counterpart to a movement of projects is an industry. When we study hacking at this analytical level, we look at how a landscape of interoperability standards, embedded infrastructures, intellectual property laws, alternative licenses, and so on, all of which have since taken on the appearance of "second nature," have come into being (see Russell 2014 for an emblematic study in this vein). Recuperation processes operating within this time horizon shape the framing conditions for individual hacker projects. Rather than resulting in a single, marketable product, as is the case in the first time horizon, the interplay of critique and recuperation within the second time horizon results in organizational innovations, leading to structural reforms across several sectors and/or the birth of new industries.

It is necessary to complement case studies of hacker projects/communities with a longer historical perspective, otherwise we would lose sight of the framing conditions of those individual projects. When moving up to this level of abstraction, the analysis is conducted at one remove from the locality of the practitioners and their direct experiences of recuperation attempts. Constructivist scholars in the field of STS will object to the deterministic slant of such an argument, which they believe rules out agency. We concede that our interpretative framework does rule out agency, provided that this word is interpreted in strictly individualist terms, as the freedom of an individual to act unhindered by constraining forces. However, our interpretative framework accords with a collectivist understanding of agency, although we prefer to talk about it in terms of a political, collective subject. Indeed, the purpose of historicizing the framing conditions of a hacker project is to dismantle the appearance of (second) nature that those conditions tend to acquire over time. Looking backward, the legal and technical landscape within which hackers find themselves can be shown to be a remnant of past struggles. Looking forward, the outcome of past struggles can be shown to determine the balance of forces in the present and in struggles yet to come.

Implied in the reference to the naturalness of a second nature is that the critical reading of the situation that we put forward here will not be immediately apparent. Consequently, when working at this level of abstraction, our theoretical claims about recuperation find less support in the

testimonies of practitioners, compared to when analyzing hostile enclosure attempts within a particular hacker community. That being said, a window of opportunity for observing recuperation processes within this intermediate time horizon is opened up when hackers face strategic junctures with a bearing on the future of the whole movement of projects and communities. A few examples include: the splitting of the free software and open-source movement, the introduction of a third, updated version of the General Public License, and the competition between different licenses dedicated to open hardware development. However, the turning points in hacker culture can often only be discerned with the benefit of hindsight. It takes a long historical perspective for the slow-moving processes of recuperation to become detectable within the empirical material. Hacker historiography plays a strategic role in the struggle against recuperation.

Steven Levy's classic work about hacker culture offers a point of departure for developing such a historical approach (1984). He identifies generational shifts in the history of hacking, spanning from the MIT hardware hackers of the 1960s to the emergence of free software in the 1980s. The common ethos, cultural references, and political goals that hackers rally behind have undergone shifts from one generation to the next. Levy also shows that each generational wave of hacking was closely intertwined with developments in the computer industry and government (Zandbergen 2011), a conclusion that has since been reaffirmed by many historians. By following the biography of Stewart Brand and his associates, Turner has documented how the West Coast counterculture of the 1960s branched off into a cyberculture, from which grew the nascent Silicon Valley (2006). As mentioned above, the idea of building a small, personal computer to foster free communication practices and new modes of community grew out of the "small is beautiful" philosophy that underwrote the hippie communes and the political protests that were taking place at the same time (Flichy 2007). These mutual influences went deeper than the mere exchange of legitimacy described above. Cultural and organizational elements of the counterculture, which already resonated with tropes from information theory and cybernetics, were foundational for the computer industry.

Continuing the list from where Levy left off, the free software movement has been complemented in recent years by movements in physical hackerspaces, in open hardware development, and in DIY biohacking. It

goes without saying that the reality is messier than this compilation of examples. No sharp line separates these movements, nor do they follow one another in neat succession. The free software movement has not disappeared from the map because of the rise of a movement around open hardware development. That being said, it is useful to distinguish between them analytically. Each one consists of a constellation of different hacker projects, held together by the common pool of resources and tools from which they draw. They can also be told apart by the distinct legal and regulatory spaces particular to each field. Corresponding to these are different branches of the industry. The role that the computer industry plays for the free software movement is equivalent to that of the home electronics industry for open hardware developers. Biohacking relates to various agricultural and medical branches of the industry, in addition to entertaining a special relation with US federal authorities.

From the history sketched out above, it must be clear that there never was a pristine, golden age of hacker culture that was captured at a later point by the industry and governments. The antagonists evolved in tandem from day one. Putting it differently, the starting point of hacking is always already recuperated to some degree. Saying this does not rule out the possibility that there are ideas and values in a local context that need to be defended from corporate or government involvement. In this defense, practitioners rely on "usable pasts" (Kelty 2008, 64–66), of which a key idea is that of a forlorn golden age of hacking. It is with the help of such usable pasts, projected backward in time, that hackers may construct baselines for historical comparisons and rally support behind their cause. This is most visible at times when hackers face a strategic junction in the road. One such junction came with the launch of the "open source" initiative in the 1990s, which sought to steal the leadership flag from the ideologically stringent free software movement. The schism revolved around the strategic choice of how accommodating hacker culture should become toward external, corporate, and government interests.

It is not incidental that the two sides clashed over the terms of the free/open license. Changes in the license are key, because they lay down the conditions under which firms are sanctioned by community norms to extract profit from the collective labor of hackers. Although individuals and firms are in principle entitled to make profit from free and openly licensed goods,

the reciprocal obligation to disclose information puts a de facto limit on profit maximization. The struggle over the terms of the license, and the vigilance by which those terms are then enforced in hacker projects, sends a signal about where the balance of forces is at any given point in time.

We believe it is meaningful to talk about a long-term trend in the coevolution of hackers and industry. The interactions between them are modified by the growing awareness on both sides of their mutual dependency. The historical lessons about the computer industry's indebtedness to the counterculture have been reprocessed and incorporated into an endless array of strategies and counterstrategies. On the part of hackers, the anticipation that their projects will be assimilated by corporations feeds into their representations, values, and individual choices. The aforementioned open-source initiative is a case in point. On the part of industry and government, methods and routines are developed to render interactions with hackers more manageable, rentable, and secure.

A turning point was the general realization in the corporate world that, by making their source code public, some of the waged programming processes could be put out to hackers, something that had previously happened by serendipity. Numerous experiments with reorganizing in-house programming labor have followed, clearly inspired by the project-based and community-centered model of free software development (Auray and Kaminsky 2007; Remneland-Wikhamn et al. 2011). For instance, companies organize hackathons and open competitions to solve difficult computer problems or to find security loopholes, headhunters use free software content management systems to scout for talent, and methods of coding free software have been detached from that context and reintroduced into corporations under the label "agile methodology."

A lesson of the same import, which corporate executives already understood from the first round of battles over filesharing in the early 1990s, is that innovations and profits may also be procured from customers and users hostile to a firm's goals. In defying a firm's prerogatives to define prescribed uses for its branded products, hackers may stumble across new uses for old products, untapped consumer demand, and novel business models. The illegal status of these activities does not prevent firms from benefiting from the outcome. A case in point is distributed and anonymous data retrieval systems, which were initially developed to usurp the

enforcement powers of intellectual property holders. Nowadays it is an industry standard, because it is far more efficient to retrieve data from multiple sources than from a single, centralized source. Academics in the field of innovation studies collect more such examples and offer their advice to companies on how to become even better at "harnessing the hacker" (Flowers 2008; Tapscott and Williams 2006).

The computer industry exports its production methods, along with its algorithmic services, to subservient sectors of the economy. A case in point is the platforms for involving users and audiences in firms' innovation processes on a systematic basis. With the open innovation model, peer production communities are transformed into pools of free labor. Peer production in its inverted, nightmarish form goes under the name of "the sharing economy." The truth behind this euphemism can most clearly be seen in the lower tier of the labor market. Experiments with self-managed, nonmonetized information systems that hackers undertook with utopian aspirations provide the backbone for capitalist restructuring in other sectors. Two examples will suffice. Information billboards that were set up by commuters in order to allocate empty car seats among themselves (ride sharing) bore fruit in Uber. A bottom-up initiative for coordinating travelers with empty couches in residential homes (couch-surfing) foreshadowed Airbnb. All it took was to add a cash nexus to the information service.

As suggested by the two examples above, the recuperation of hacking has consequences that extend far beyond the affected hacker community. Subsumed under an open innovation regime, hackers become a source of innovation and organizational restructuring for capital and, thus, an engine for intensifying commodified and exploitative relations everywhere in the economy. In order to catch sight of the framing conditions that precede and envelop hacker culture, the analysis needs to scale up to a third time horizon.

THIRD TIME HORIZON: THE NEW SPIRIT OF HACKING

The third time horizon relates hacker culture to capitalism as an evolving whole. From this perspective, the dynamics of critique and recuperation are seen as the motor, not only of an individual firm's marketing and innovation strategies, nor of an industry that undergoes restructuring,

but also of epochal transitions within capitalism. During these epochal shifts, the predominant logic of how capital accumulates is rewritten from the ground up, affecting production, circulation, consumption, and distribution. It is within this third time horizon that the first (the life cycle of an individual hacker project) and second (the coevolving landscape of hackers and industry) acquire their determinate meaning. When we move up to this level of abstraction, explanandum and explanans are reversed. Instead of explaining hacking with theories about capitalism, it is capitalism that is examined through the lens of hacking.

We are indebted to Luc Boltanski and Eve Chiapello's *The New Spirit of Capitalism* in making this suggestion (2005). The book title alludes to Max Weber's famous argument that capitalism was made possible by the ethical foundations created by Protestantism. Extending this idea to modern times, Boltanski and Chiapello argue that the transitions that periodically take place under capitalism are triggered by the critiques that are leveled against it. In order for the economic system to operate smoothly, it needs to appear legitimate and even inspiring to both workers and consumers. Lacking an ethical or affective source of its own, capital must draw legitimacy from external sources. The incorporation of critique through a period of restructuring allows capitalism to overcome its own impasses.

This transhistorical scheme neatly fits the current spirit of capitalism, which Boltanski and Chiapello name "connexionist." They trace the latest iteration of capitalism back to the anticapitalist critique that emerged from the political upheavals of the 1960s. The values associated with May 1968, individual freedom, self-expression, nonconformity, authenticity, and so on, were quickly depoliticized and transformed into an ethical foundation for capitalism in its present-day, consumerist, and financialized form (see also Liu 2004; Harvey 2005).

The 1960s and 1970s have been designated a turning point in capitalism by many otherwise unrelated schools of thought. Often, the argument is mapped onto the transition from Fordism to post-Fordism. Under Fordism, mass production was the defining trait of every economic activity, including such phenomena as mass party and mass union activism. Under post-Fordism, the flexible, just-in-time production model saturates every corner of society, not even sparing cultural expressions or political activism. Approaching it from a structuralist perspective, the regulation

school spoke of these epochal shifts as "regimes of accumulation" (Aglietta 1976; Jessop 1995).

The calling card of the structuralist approach is a deep suspicion directed toward the cognitive capacity of practitioners to make sense of the situation in which they find themselves. For most of his career, Boltanski has polemized against such a demeanor. Boltanski and Chiapello are aware of the fact that, by proposing a transhistorical scheme for interpreting practitioners' behavior and claims, they end up vulnerable to the same kind of accusations that they previously directed against structuralist sociology in general, and Pierre Bourdieu in particular: "Stressing historical structures, laws and forces tends to minimize the role of intentional action. Things are what they are. Yet the critical approach becomes meaningless if one does not believe that it can inflect action, and that this action can itself help to change the course of things in the direction of further 'liberation'" (Boltanski and Chiapello, 2005, x).

Undoubtedly, this warning is also applicable to the interpretative framework that we are developing here. Our musings about critique and recuperation within the third time horizon find scant support in the testimonies of hackers.

At this point in the argument, we take our cue from the autonomist Marxist tradition to explain how historical reflection on a grand scale can be developed without the analysis succumbing to deterministic explanations. Autonomist Marxists conceptualize the transition from Fordism to post-Fordism as a change in the "technical and political class composition." The stress is placed on class struggle as a driver of capital's restructuring processes. The autonomist Marxists apply this line of reasoning directly to the instruments of production. This we consider to be an advance over Boltanski and Chiapello's scheme. The latter have surprisingly little to say about the role of technology in their interpretative framework. Perhaps they dodged the question out of concern that they may become associated with the genre of writing that hails the coming of "postindustrialism," the "information age," or "network society." A common feature of this genre is that information technology is designated as the motor of historical development. With a focus that locates the root cause of technological change in the struggle between labor and capital, we can reconstruct long time horizons without having to resort to the notion that history is

propelled by an "innate trajectory of technology." Furthermore, empirical support for the autonomist Marxist claim is easy to come by in hacker culture. As we discussed previously, a long list can be compiled of disruptive innovations in information systems that emerged from contestations between hackers and their corporate or state adversaries.

We are not the first to suggest that the new spirit of capitalism can be found in a condensed form within hacker culture. Drawing directly on Max Weber's writings, Pekka Himanen (2001) argues that hackers embody a new work ethic that is about to supersede the protestant work ethic of the past. He praises this trend in what amounts to a new take on an old trope in management literature, according to which the next industrial revolution will abolish alienated and deprived working conditions. Anne Barron (2013) also points to hackers as representatives of a new work ethic, but she argues in a more critical vein, starting from Boltanski and Chiapello's reinterpretation of Weber. We concur with her that the values embodied in free software development qualify as a particularly pure form of the new spirit of capitalism. The reverse side of the critique against proprietary software and other forms of closed innovation systems is an emotional investment in "open" forms of capital accumulation. Thus, the oppositional stance of hackers has been turned into an ethical foundation for contemporary capitalism (Barron 2013).

The subjective counterpart to the open innovation model is the outsider. Identification with the position of the outsider goes to the heart of hacker culture. The same idea of freedom as having "no strings attached" underpins the promise that emancipation will flow from the repurposing of tools and the circumvention of social constraints by technological means—that is to say, from hacking. Alas, such acts of technology-mediated transgression have always already been anticipated by the open innovation model. As sources of value production external to the firm become increasingly important for capital, so the more pervasive the methods will become to reassert management control over this distributed chain of production. Furthermore, open innovation is only a subcategory of a more general trend in which value production is relocated from contractual employment relations to peer production communities, crowds of users and audiences, and clouds of click workers. This is the spirit of capitalism that is embodied in the work ethic of the hacker.

THE FUNCTIONAL AUTONOMY OF HACKER PROJECTS AND HACKER COMMUNITIES

The outcome in the struggle against recuperation decides to what extent hackers will shape technology and, conversely, to what extent hackers themselves will be shaped by and contribute to predominant structures, including predominant technologies. Struggles over recuperation center on the functional autonomy of hacker culture vis-à-vis its exteriority. By the concept of "autonomy," we understand the ability of a collective to give itself its own laws and future-oriented goals. Autonomy only exists in relative measure. The notion of absolute, unconditional autonomy is a misnomer, just as much as the fantasy of the free-floating position of the outsider, as discussed in the paragraphs above. Hence, we aspire to give equal weight, on the one hand, to the dependence of hackers on preexisting structures (such as, for instance, industrial standards, economic relations, cultural values, etc.) and, on the other, the discretion that hackers nevertheless possess to pursue pathways in technology development that diverge from predominant trends.

We build on a tradition within political philosophy in which autonomy is understood as a phenomenon that first emerged in struggles for political emancipation. Only much later was autonomy defined in analytical-philosophical terms. The point zero of autonomy is not, in other words, Kant's philosophy, although he is an important stopover in the modern history of the concept. Rather, the idea of autonomy dawned on humankind during the process of resisting imperial and dynastic domination. Two milestones in this history are ancient Greek poleis and city-republics during the periods of the Renaissance and Reformation, who mobilized against oligarchic coups and armed interventions. From this, there follows a point of utmost importance. Autonomy is not a mark of individual cognitive and moral beings, but rather a property of collective, political subjects that is consolidated through struggle (Rosich 2019).

Likewise, we take issue with an influential tradition in political philosophy, represented by Hannah Arendt among others, according to which "autonomy" is a sphere of freedom that is antithetical to needs, work, and technology. If autonomy is conceptualized as ontologically opposed to labor, then one makes oneself oblivious to contestations over autonomy

that are fought out on a terrain of material needs, economic dependency, and technological infrastructures. In the idealist school of political philosophy, "autonomy" is understood in such a way that it rules out the possibility of an "extension of the franchise" to the working class. This philosophical outlook resonates with millennia-old class prejudices.

As a corrective to both the individualist and idealist readings of the word "autonomy," we draw inspiration from how struggles over autonomy have been conceptualized and studied in labor process theory (Hanlon 2016). In case studies of worker resistance against management prerogatives and Taylorism, it is a given that the autonomy of workers is limited by the contractual employment relation in which they find themselves. This contrasts with the abstract notion of "absolute freedom." Furthermore, in the context of workplace struggles, autonomy must be understood as embedded in a material substratum: the distribution of skills among the workforce, the layout of the factory and design of the machinery, the economic bargaining power of the class antagonists, and so on.

From David Montgomery's seminal study of labor activism during the second half of the nineteenth century, at a time when factory operations were largely in the hands of self-governing craftworkers, we derive the notion of "functional autonomy" (Montgomery 1987). Due to their intimate familiarity with the production process, craftworkers exercised a functional autonomy over the factory, even though, nominally speaking, that same right to dispose of their time and effort was held by the factory owner. It is precisely this discrepancy between the nominal rights assigned to the capitalist, and the practical control exercised by the craftworkers, that transformed skill levels and machinery design at the point of production into a major battleground in the ensuing cycles of struggle.

The continued relevance of investigating the functional autonomy of workers, however much it has been encroached upon compared to that of nineteenth-century craftworkers, is attested to by the battery of countermeasures deployed by the representatives of capital, such as, for instance, scientific management, the automation of work tasks, and union-busting tactics (Delfanti 2021). From labor process theory, we derive an analysis of technological design and skills that are understood to be through and through determined by the balance of forces between labor and capital.

Technology is pivotal in this interpretative frame, but not exclusively so. We have also taken a cue from labor process theory in directing our analytical attention toward the intersection between artifacts and skill, on the one hand, and group solidarity and norms, on the other. The two are equally decisive for the outcome of industrial conflicts (Montgomery 1976).

The tug-of-war between workers and managers over who is in effective control over the shop floor provides a blueprint for studying the functional autonomy of hacker culture within a relation of dependency upon the industry. Although, by definition, hackers have no contractual ties to an employer, they are nevertheless dependent on the industry in numerous respects: in order to be paid commission, access technical products, gain political leverage, and so on. For the hacker, just as for the employee, the distribution of productive skills and the design of machinery are decisive factors. The autonomy of hackers is curtailed, for instance, when software tools and software libraries are fenced in behind intellectual property rights. Conversely, their autonomy expands as tools and libraries are brought into the information commons. Whatever else is being produced by a thriving, autonomous hacker project, it concurrently furnishes the material conditions for its own continued existence as an autonomous group entity.

We label a positive outcome in struggles over recuperation "communization." We take this word from the discourse of the self-described ultraleft tendencies, where it refers to steps toward the direct realization of Marxian communism within the context of everyday social relations (e.g., Dauvé 2011; Friends of the Classless Society 2016). This is in line with our conception of the term, but we use it in a narrower sense here, referring to the rerouting of work time and resources, nominally owned by the employer, to sustain commons-based peer production communities. The resources may also be procured through sponsorship from companies. Even so, it indicates that the hackers are in a strong position to define the terms of engagement with industry actors. In chapter 6, we discuss Internet Relay Chat as an example of communization. For decades on end, system administrators working for internet service providers and university departments have allocated their employers' servers to run public Internet Relay Chat networks. Their ability to do so, with or without the tacit

agreement of line managers, hinges on them having technical know-how and access rights. Hence, processes of communization often interlock with a high degree of functional autonomy.

A bone of contention in labor process theory, which has a bearing on our discussion about hackers, concerns the double-edged implications of workplace autonomy. Coercion is not the sole tool in the manager's toolbox. Another way for managers to ensure consent to work is to give workers some leeway (Burawoy 1979). Indeed, it is commonly observed in this literature that the smooth operation of a factory presupposes that the workers have some discretion over task selection and that they feel somewhat responsible for the quality of their output. When managers tighten their control over the production process, often in response to an outbreak of industrial conflict, productivity tends to be negatively affected.

The same qualification applies to the subsumption of a hacker project under an open innovation model. The reason why capital relocates production from employed, in-house labor to commons-based peer production communities is not merely to tap into free (as in gratis) labor. Equally important is the higher productivity and greater inventiveness that is fostered outside of the wage labor relation. As much is suggested by the technical superiority claimed for GNU/Linux over proprietary operating systems. The coerciveness of waged employment is suboptimal for organizing productive activities, at least in the upper segment of the value chain, where value is extracted from the workers' subjectivity. The open innovation model seeks to remedy this situation. Managerial control over the innovation process is weakened in order to set productivity free. That being said, some degree of managerial control must nevertheless be upheld, in order for capital to valorize the collective endeavors of hackers. Furthermore, the imperative of profit maximization spurs firms to reassert control over commons-based peer production communities, occasionally killing the proverbial goose that lays the golden eggs.

As has already been established in labor process theory, there is a tension between the need for capital to, on the one hand, assert managerial authority over the labor process and, on the other, give workers some margin of freedom in order for them to determine the most productive application of the factory machinery. This tension is typically resolved by refining the division of labor among the workers. Tasks are unevenly distributed

among different classes of workers, a separation reinforced by status hierarchies at multiple levels (education, gender, ethnicity, and nationality, to mention the most important ones). Building on this historical lesson, we suggest later in this chapter that there is an analogous development taking place outside of the contractual employment relation. A division of labor is emerging between three different classes of nonemployees: communities of peer producers, crowds of users, and clouds of click workers. We base this classification scheme on the relative degree of functional autonomy that is exercised (or not) by these collective formations.

Obviously, hackers sit at the top of this value chain. By saying this we have not, however, exhausted all that can be said about them. Rather, this observation serves as our starting point for investigating the shifting degrees of autonomy of different hacker communities, all of them privileged in comparison to the click workers. In our understanding, it qualifies as a high degree of autonomy when hackers can dictate the terms under which they (collectively and individually) enter into a symbiotic relationship with industry and government actors. On this hinges their ability to cultivate critical opinions and imaginaries about technology that deviate from mainstream engineering practices. In the absence of such autonomy, hackers will develop technology that reproduces predominant structures and proclivities. Concurrently, however, in order to gain traction, hackers must produce something of value to society, which is what gives them leverage against their more powerful allies. In the absence of such allies, the hacker project will become isolated and lose its societal relevance. The symbiotic relationship between hacker culture and the computer industry is not optional—the real question is on whose terms this symbiosis is based.

In order to speak about functional autonomy with more precision, we now proceed to identify three pillars upon which it rests. The first pillar is the technical expertise of hackers. The second is their shared values and norms, cultivated and sustained in partial isolation from mainstream engineering culture and society at large. The third is historical memory, or, put differently, a common narration of events in the past with a bearing on the collective's future development. All of these pillars must be in place for hackers to successfully reproduce the social and material conditions for their own continued existence as an autonomous and self-directing collective. On this, in turn, hinges the capacity of hackers to pursue

alternative pathways of technological development, as well as their cultivation of independent opinions about policymaking related to information systems.

TECHNICAL EXPERTISE

Hackers' ability to understand and produce technology is important in a number of respects. Understanding how something works is a prerequisite for judging its wider significance to one's community and society at large. Such judgments inform and enable critical engineering practices, by which we are referring to hands-on practices and design choices that seek to promote social relations different from the hegemonic ones. Hacker politics chiefly consists of the creation of artifacts (software, hardware, protocols, or biological processes). It is on this score that, similarly to TPMs more generally, they distinguish themselves from traditional social movement activism, such as street protests and the petitioning of politicians. Furthermore, the spread of hacker culture lowers the threshold for accessing technical expertise. In the best-case scenario, disenfranchised groups may thus acquire the necessary know-how for contesting issues of immediate concern to them, above and beyond the information systems that hackers care about.

The same skills that circulate in the computer underground are also on offer on a regular engineering curriculum. Even so, these different training grounds give rise to very different results. The prescribed career path of the engineering profession is, firstly, to have been a hobbyist in one's teens, then to get a university degree in informatics or physics, and, finally, to get a job in a tech company or start one's own firm. Many computer engineers walk this career path without ever coming into contact with hacker culture. The educational system gives them the competences to assess and intervene in technology, which amounts to the first pillar of autonomy in our analytical scheme. However, having been drilled in a hegemonic conception of what technology should be, professional engineers tend, even in the absence of financial incentives and managerial dictates, to channel their efforts into the reproduction of predominant social relations (Noble 1977). The criteria for deciding the success or failure of a development project imbued with mainstream engineering culture is laid down by "instrumental rationality." That is to say, the whole endeavor will be

directed toward the optimization of cost efficiency relative to technical performance.

Hackers are not oblivious to performance in the narrow, technical sense, but aesthetic and ethical considerations also carry great weight when they assess whether or not to adopt a new artifact. A premium is often placed on simplicity and transparency, which occasionally overrule the imperatives of cost efficiency, speed, and user convenience. This can be observed in the common practice among hackers of sticking with outdated and, in terms of technical performance, "inferior" products and standards, in comparison to the up-to-date versions. Noteworthy is the widespread rejection (Wyatt 2010, 9) and critique of smartphones, commonly referred to in hacker lingo as "tracking devices." Just by being slightly out of sync with the latest marketing scheme, outdated technology may give its users some extra leeway. Furthermore, the slower pace of diffusion of older technology is conductive to its integration into moral economies, social conventions, and norm systems that have not been fully subsumed under corporate and elite control. As the gap widens between the old standard and the latest iterations, however, inoperability takes its toll, and the alternatives promoted by hackers become isolated from the wider society. The goals of ideological purity and political relevance are in tension. Hackers must refine their taste in judging the right balance between the two goals and realizing when to switch platforms, along with reproducing more conventional skills in soldering, programming, and so on.

SHARED VALUES AND NORMS

Cultural tropes and values are reproduced where hackers gather. The most frequented of such meeting places are the ubiquitous asynchronous and synchronous online social spaces, such as mailing lists and chat rooms (the latter analyzed in chapter 4). Complementing online interactions, physical meeting places serve a critical function in disseminating and entrenching a specific hacker culture. Hacker conventions—dubbed by Coleman "a ritual condensation and celebration of a lifeworld" (Coleman 2010)—allow hackers to convene at regular intervals from many different places, while shared machine shops provide fixed spatial coordinates for the hackers in a city or region to meet and forge bonds over time.

Shared machine shops, notably the hackerspace, as described in more detail in chapter 3, are a materially and symbolically constituted milieu of hacker culture. In this setting, subcultural symbols and rituals are transmitted via everything from culinary preferences to references from popular culture. Some examples include: the fridge in a hackerspace is typically stocked with Club-Mate, the default drink of hackers (Thomas 2014); the soldering iron stall warns that "if it smells like chicken, you're holding it wrong," which is a reference to Mitch Altman's soldering workshops; names of people and artifacts are taken from Discordianism, the Cthulhu mythos, or *The Hitchhiker's Guide to the Galaxy*. Thus, specific tastes, habits, and even metaphysics are reproduced in hackerspaces that lend support to and legitimize the core technical practices. Hackerspaces, although varying greatly in cultural and ethical preferences, constitute the most stable physical manifestation of hacker culture.

Technical expertise (the first pillar) and shared cultural tropes and values (the second pillar) will not on their own suffice to sustain the functional autonomy of hacker culture. As much is suggested by the Asian and Chinese hacker scene. From the research literature, we learn that "hacking with Chinese characteristics" draws on the same skill set and roughly the same cultural tropes as those circulating globally, but the antiauthoritarian and confrontational outlook of the original hacker identity is largely absent (Lindtner and Li 2012). The same observation can be made about the geographical displacement of shared machine shops, from Europe to North America and then to every corner of the world. The anarchist politics of the original hacklabs was lost in translation when the idea caught on in the United States and they were rebranded as "hackerspaces" and "makerspaces." This prompts us to stress the third pillar upholding the functional autonomy of hacker culture: the transmission of shared memories and lessons learned from the defeats and victories of older generations, which can provide points of orientation for future-oriented, collective action.

HISTORICAL MEMORY

Familiarity with past waves of technology is passed on to new hackers in part through the floating debris of obsolete gear that piles up in most hackerspaces. Recycling is the backbone of their political economy, the junkyard

furnishing them with both spare parts and inspiration for new projects. Antiquated machines, unfinished projects, and random electronic parts are stored on the shelves of hackerspaces and serve as a reference library of engineering solutions. When wondering about the proper way to wire a chip, one can find such chips already wired into some other device and use it as an example. Aside from the pragmatic aspects of recycling, old artifacts also elicit sentiments of nostalgia and even veneration. Some hackerspaces end up playing the part of technology museums. A case in point is Hack42 in Arnhem, the Netherlands. Housed in an old military barracks modeled after German countryside cottages, the three-story hackerspace includes several thematic collections of obsolete hardware: cameras both analog and digital, overhead projectors and beamers, typewriters, calculators, and computers, many of which are kept in working order.

Another anecdote suggestive of how old artifacts are called upon to articulate a critique of current trends in technological development is Mitch Altman's signature invention, TV-B-Gone. This is a universal remote control with a single button that turns off any television. Hackers wielding the TV-B-Gone convey the antitelevision sentiments of an earlier generation of computer users, for whom the television symbolized passive media consumption bordering on corporate mind control. The rejection of mass media was a thematic core for the hardware hackers who first envisioned the "small is beautiful" personal computer in the 1970s (Levy 1984). At the present moment, however, the TV-B-Gone device doubles as a critique of the ongoing convergence between the internet and broadcasting media, culminating in corporate-controlled video streaming platforms.

Appreciation for bygone computer architectures and artifacts sensitizes hackers to the alternative pathways that technology could have taken under different circumstances. It provides them with a baseline for comparison with actual developments in information systems. Nostalgia serves as a much-needed antidote to the presentism that is rampant in the high-tech sector. Tied to this backward-looking sensibility is a diagnosis of the actors and structures that impinged upon the development process in the past and, hence, continue to do so today. The interpretation and transmission of past events with a bearing on the collective identity of hackers is key to shaping their political outlook and guiding future-directed, concerted action.

COMMUNITIES OF PEER PRODUCERS, CROWDS OF USERS, CLOUDS OF CLICK WORKERS

Thus far into our description of how hackers become caught up in struggles over recuperation, we have postponed the crucial question of if, and if so, how, solidarity across petrified and compartmentalized identity boundaries can be constructed on the basis of hacker culture. We have no illusions about this being an easy task that is just waiting to happen. Both subjective and objective factors pull hacker culture in the opposite direction. Still, unless such bonds of solidarity can be forged with other social movements and disenfranchised constituencies, the hacker culture is destined to become a subservient incubator of innovation for the benefit of capital.

In terms of the subjective side of hacker culture, the political agnosticism of hackers is renowned (Coleman 2004). This does not rule out activism, but it does put an inward spin on their version of politics. As we discussed above at length, the orientation of hacker culture is toward expanding the material conditions for perpetuating its own existence as an autonomous collective. In the literature, this trait of hacking politics has been described as "recursiveness" (Kelty, 2008). The advantage of recursive politics is that, when hackers take a stand on policies that have a bearing on their continued existence as a collective, for instance, regarding extensions of intellectual property rights or deviations from the principle of net neutrality, they act in concert. The downside of the same thing is that hacker culture tends to be unreceptive toward larger political issues that do not arise from within this recursive loop. A case in point is the coarse welcome given to female hackers in some discussion forums and hackerspaces. The first kind of topics are typically construed as apolitical, a mere optimization of given engineering parameters, even when applied to policymaking and legislation (Gillespie 2006). The second kind of activism comes across to many hackers as ideological foul play by intruders. In sum, the political agnosticism of hacker culture does not promote the forging of solidarity bonds that extend beyond its own, insular concerns.

To the subjective side of hacker culture, we must add the objective class position of the computer engineer, to which the hacker belongs at one remove. Without question, the engineering profession is one of the most

privileged segments of that part of the population that has to earn its means of subsistence on the labor market. In addition to the high salaries in the high-tech sector, engineers are overwhelmingly white, college-educated males living in metropolitan areas in the Global North. From a world system perspective, programming labor sits at the top of a global value chain that descends to miners who excavate lithium in Bolivia, maquila workers producing home electronics in Mexico and China, migrant workers in metropolitan tech centers providing supportive functions, and slum dwellers around the planet recycling electronics waste (Dyer-Witheford 2015). At the point of production, finally, computer engineering can serve various functions, but the decisive one is to reinforce management proclivities. The history of computer programming goes back to numerical control machinery that was deployed by managers to extend their control over factory machinery, and, by extension, over the machine operators (Noble 1977).

At best, it may be granted that computer programmers belong to a modern version of the "labor aristocracy." If so, two partially countervailing lessons can be teased out from the history of organized labor. The first lesson is that with privilege comes susceptibility to hegemonic ideas and values. One may easily draw parallels between, one the one hand, the exclusionary practices and macho jargon that saturate much of hacker culture (Bardzell, Nguyen, and Toupin 2016; Dunbar-Hester 2019) and, on the other, the strategies by which craftworkers once protected their working conditions by excluding female and casual laborers from workplaces and/or union membership (Cockburn 1985). There is, however, another story to be told about the labor aristocracy. Spared from absolute destitution and precarity, they were in a better position than many other elements of the working class to organize themselves and their fellow workers into labor parties, trade unions and consumption cooperatives (Moorhouse 1978).

Extrapolating this historical lesson to present-day hackers, we note that the autonomy of hacker projects is sustained in large part thanks to the resources that are pulled into the hacker movement by overpaid programmers. Furthermore, the alternative design choices made in this milieu, such as, for instance, the decision to make source code public, is beneficial to the majority of computer users, although they will never engage directly with computer programming. The point is that there are contradictory potentialities contained within the subject position of the hacker. Which of these

potentialities will be actualized is decided over the course of an ongoing struggle, hence the outcome cannot be told in advance. How we choose to study hacking contributes in some small measure to this outcome.

Academics like ourselves who criticize hackers for being technophilic and for engaging in exclusionary practices have a point insofar as those critiques are made with the intent of encouraging hackers to incorporate previously disenfranchised groups. The interpretative framework that we are developing in this book, in which hacking is situated within a larger whole of evolving capitalist relations, is not meant to be explanatory only. By connecting the lines between seemingly unrelated dots, we want to suggest that the "loop of recursiveness" that delimits the agnostic politics of hackers ought to be cast much wider. Implied in this analysis is the conclusion that hackers strengthen their own position by extending the bonds of solidarity to other groups who are caught up in the same forces of recuperation.

In order to continue along this train of thought, we must zoom out from the special case of hacker culture and revisit the discussion from the opposite direction, that of the spirit of capitalism. The open model of capital accumulation is mirrored by a working class that, even after the factory walls have been demolished and the collective identity of the Fordist mass worker has been dissolved, is still obliged to sell its labor in order to earn a subsistence. It is therefore urgent to investigate the subjective experiences of class within a dispersed, capitalist chain of production with the following question in mind: What collective representations can emerge from a setting where the means of living must be earned on the labor market, but where the everyday experiences of class antagonism are no longer framed by the bipolar conflict of interests between employer and employee, as codified in the employment contract? Employer and employee confront each other over working hours, pace of work, remuneration levels, and so on. Such pedagogical support is missing in the new forms whereby capital extracts value from labor. With the exception of a handful of rare cases (Postigo 2004), fans, gamers, users, and so on, even though their hobby has become integrated into capital's circuits, continue to refer to themselves as something other than exploited workers (Lee and Lin 2011). The fragmentation of class antagonism is fueled by the never-ending stream of neologisms invented by the platform owners to name their workers,

such as "taskers," "runners," and "Turkers" (Irani 2015a). The argument comes full circle when we insert the "maker" and "hacker" into this context. Hacker projects are not only a showcase for how commons-based peer production communities can be put to work by firms. Once they have been put to work in this way, hackers make substantial contributions to the organizational and material infrastructure of the emerging social factory.

When a hacker project has been subsumed under an open innovation model, it furnishes capital with ideas and innovations that can then be deployed in other sectors of the economy. At first, it might seem as though those who stand to be most negatively affected by the free offerings of programming labor would be waged labor in the same sector. However, the salaries and working conditions of waged computer programmers have not been markedly affected by the surge in free software development. The case could even be made that their collective bargaining position has been strengthened thanks to there now being an alternative forum and labor market to which they can turn. Ultimately, this comes down to them acting from a position of strength. The objective class position of programmers in the global and social division of labor is such that they stand to benefit from the transfer of resources to the high-tech sector from all the other sectors of the economy.

It is in the lower tiers of the labor market that the coercive side of recuperated engineering utopias come to the fore. Euphemisms such as "the sharing economy" and "the gig economy" cast a veil over the exploitative and precarious working conditions of a new generation of workers (Scholz 2016). This is the mirror side of capital's growing dependence on sources of value production external to the firm and, consequently, external to the contractual employment relation. With this comes the need to assert managerial control over the decentralized labor process. Corresponding to this development, a new division of labor is cropping up between different classes of developers, beta testers, users, audiences, and, further down the chain, taskers, Turkers, runners, and so on. We propose the following typology: "communities of peer producers," "crowds of users," and "clouds of click workers."

At the pinnacle of the open innovation model is the commons-based peer production community, of which free software developers are the

paradigmatic example. This community is a self-initiated and voluntarily entered association with a large capacity for collective action. Autonomy is a necessary condition for incubating the kind of innovative and problem-solving activities from which capital derives the highest value. Next in line come the crowds of users and audiences of various sorts. They swarm together on what seems to be a voluntary (noncontractual, nonremunerated) basis. Upon closer inspection, however, it turns out that they have often been algorithmically herded into environments where their activities can be mined for data or for other kinds of long-tail derivatives (beta testing, computing power, etc.). The unpredictability of this extraction model is that, under exceptional circumstances, the crowd may explode with political energy and, at least for a brief moment, turn into an angry mob. At the bottom end are clouds of click workers, who perform predefined, routine tasks on corporate-owned digital platforms under strict surveillance. Just as with the layout of the factory, the digital platform has been conceptualized from the outset to minimize communication and self-initiated coordination among the individual members of the cloud.

The list "community, crowd, and cloud" is made up of names that we give to different, nonemployed work constellations. These words have long-established uses and connotations predating the rise of information systems. What motivates our chosen terminology, however, is the common usage of these words in the context of online communication technologies. In keeping with the classificatory work previously undertaken in social movement studies, we distinguish community, crowd, and cloud according to their differentiated capacity for collective action (Dolata and Schrape 2016), while adding an additional layer to this analytical scheme by considering how capital extracts value from them. Hence, the defining criterion of communities, crowds, and clouds consists of how much discretion (if any) these constellations exercise over the purpose to which their labor is put. The relative degree of functional autonomy enjoyed by these constellations stands in an inverse relationship to their subsumption under capital. The cloud of piece rate click workers offers a near perfect image of what the real subsumption of labor under capital looks like outside of the contractual employment relation. The commons-based peer production community, in contrast, showcases a highly autonomous and self-directed labor force, to the point where one may easily forget the

influence of capital and capitalism in such a setting. It is for precisely this reason that a theoretical reconstruction of the structural interdependencies of these different work constellations is called for. In short, autonomy is a precondition for communities of peer producers to furnish capital with disruptive ideas and innovations, so that capital may better subjugate the crowd and the cloud (together with the regularly employed workforce, of course) under managerial control, consumer surveillance, and intensified exploitation.

This bleak scenario is not the only one possible. The legacy of hacking contains contradictory potentialities, some of which point toward a broader political-economic analysis and the forging of solidarity bonds with the working class at large. We are reminded of this possibility by the concluding words of one of the founding documents of hacker culture, the GNU manifesto: "We have already greatly reduced the amount of work that the whole society must do for its actual productivity, but only a little of this has translated itself into leisure for workers because much nonproductive activity is required to accompany productive activity. The main causes of this are bureaucracy and isometric struggles against competition. Free software will greatly reduce these drains in the area of software production. We must do this, in order for technical gains in productivity to translate into less work for us" (Stallman 1993).

Resistance to the Current

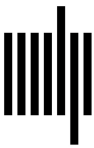
The Dialectics of Hacking

By: Johan Söderberg, Maxigas

Citation:

Resistance to the Current: The Dialectics of Hacking
By: Johan Söderberg, Maxigas
DOI: 10.7551/mitpress/13466.001.0001
ISBN (electronic): 9780262372008
Publisher: The MIT Press
Published: 2022

The open access edition of this book was made possible by generous funding and support from MIT Press Direct to Open



The MIT Press

The MIT Press would like to thank the anonymous peer reviewers who provided comments on drafts of this book. The generous work of academic experts is essential for establishing the authority and quality of our publications. We acknowledge with gratitude the contributions of these otherwise uncredited readers.

This book was set in Stone Serif by Westchester Publishing Services.

Library of Congress Cataloging-in-Publication Data

Names: Söderberg, Johan, 1976– author.
Title: Resistance to the current : the dialectics of hacking / Johan Söderberg and Maxigas.
Description: Cambridge, Massachusetts : The MIT Press, [2022] | Series: Information policy | Includes bibliographical references and index.
Identifiers: LCCN 2022007904 (print) | LCCN 2022007905 (ebook) | ISBN 9780262544566 (paperback) | ISBN 9780262372015 (epub) | ISBN 9780262372008 (pdf)
Subjects: LCSH: Hacking. | Computer crimes. | Capitalism.
Classification: LCC HV6773 .S638 2022 (print) | LCC HV6773 (ebook) | DDC 364.16/8—dc23/eng/20220711
LC record available at https://lccn.loc.gov/2022007904
LC ebook record available at https://lccn.loc.gov/2022007905