This article was downloaded by: [Lancaster University Library]
On: 29 November 2013, At: 05:06
Publisher: Routledge
Informa Ltd Registered in England and Wales Registered Number: 1072954
Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK

Economy and Society
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/reso20

Free software production as critical social practice

Anne Barron

Published online: 09 Sep 2013.

To cite this article: Anne Barron (2013) Free software production as critical social practice, Economy and Society, 42:4, 597-625, DOI: 10.1080/03085147.2013.791510

To link to this article: http://dx.doi.org/10.1080/03085147.2013.791510

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the "Content") contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly

forbidden. Terms & Conditions of access and use can be found at http://www.tandfonline.com/page/terms-and-conditions

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

Economy and Society Volume 42 Number 4 November 2013: 597–625
http://dx.doi.org/10.1080/03085147.2013.791510

Free software production as critical social practice

Anne Barron

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

Abstract

This paper analyses the phenomenon of free and open source software (FOSS) in the light of Luc Boltanski and Ève Chiapello's *The new spirit of capitalism*. It argues that collaborative FOSS production by volunteer software developers is a species of critical social practice in Boltanski and Chiapello's sense: rooted in resistance to capitalist social relations, and yet also a source of values that justify the new routes to profitability associated with contemporary network capitalism. Advanced via collective projects that are sustained by hacker norms and privately legislated 'copyleft' law, the FOSS ethos is apparently antithetical to private property-based accumulation. Yet it can be shown to embody the 'new spirit of capitalism' in its most distilled form; moreover FOSS developers have instituted new forms of property and new modes of profit creation around software that are in the process of being adapted for use in other economic sectors. Meanwhile, the private law constraints on profit-seeking that have emerged from the FOSS movement are counteracting some of the social pathologies that accompany network capitalism only to consolidate others. The paper concludes by identifying likely bases for a renewal of critique given these realities.

Keywords: copyleft; free software; immaterial labour; new spirit of capitalism; open source; reputational capital.

Introduction

The term 'hacker' has come to have a pejorative connotation as a label for a malevolent system-breaker. Yet in computing circles it is a badge of honour (Nissenbaum, 2004, p. 197),[1] reserved for those who share an anti-authoritarian, anti-bureaucratic approach to innovation, who value freedom of inquiry and who are therefore opposed to all strategies oriented towards

Anne Barron, Department of Law, London School of Economics and Political Science, Houghton Street, London WC2A 2AE, UK. E-mail: a.barron@lse.ac.uk

Copyright © 2013 Taylor & Francis

Routledge
Taylor & Francis Group

keeping the material and intellectual tools of innovation inaccessible to those with an interest in using them. Incubated in the research institutions that fostered the early development of computing in the 1960s, hacker culture became both more diffuse and more widespread with the development of the internet in the 1990s. It is now closely associated with the free software movement, but radiates out from there to encompass innovators in any field who share the 'hacker attitude' (Raymond, 2001a). Hackers see themselves not as vandals but as programming enthusiasts, committed to maximizing access to computer equipment and software, and to the information necessary to improve these tools and innovate with them. Their ethos is organized around a faith in computing progress as the key to social progress, and it holds that sharing, openness, non-hierarchical decision-making and respect for technical excellence are essential conditions for continuing improvements in both dimensions. At the heart of the hacker identity, though, are experiences of the pure joy of computer programming, born of an intense curiosity about programming challenges and an enthusiasm for solving them well – that is, in ways that other hackers recognize and appreciate – typically in partnership with other programmers (see e.g. Coleman, 2013, Ch. 3; Sennett, 2008, pp. 24–51). A commitment to the value of community and collaboration is thus also associated with these hacker norms; indeed the most noteworthy feature of free software products is that they emerge from spontaneously generated collective 'projects' on which many volunteers co-operate, within loosely structured groupings facilitated by the internet, to produce complex but nonetheless integrated artefacts that are freely available for all to use.

The reason why these products are freely available for all to use is that the copyrights in them are licensed in ways that reverse the dominant meaning of private (intellectual) property as a right of exclusion. National copyright laws now invariably provide that software which is original to its author shall be protected by copyright,[2] and, for the major commercial software producers in the US, the EU and elsewhere, licensing these copyrights on restrictive terms has become crucial to achieving ever more finely tuned dominion over access to and use of code. By contrast, the free software strategy has been to transform the copyright licence into a charter of freedoms for software users. Some such licences – notably those based on the model General Public Licence (GPL) devised by free software pioneer Richard Stallman in the late 1980s – leverage software copyrights to propagate 'software freedom' norms beyond the immediate licensee. Consequently, these 'copyleft' licences have been described as viral: the norms they contain reproduce themselves every time derivative works are created from licensed material and in turn licensed. Copy*left* is thus itself an ingenious hack of copy*right*:

> Hacks . . . take advantage of characteristics of [a] system that may or may not have been obvious to the people who designed it. . . . To call Free Software a hack is to point out that it would be nothing without the existence of intellectual property law . . . (Kelty, 2008, p. 182)

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

Elsewhere (Barron, 2012, p. 40) I have supported Christopher Kelty's more general intuition about the free software movement: that its significance lies in its tendency to problematize the technical infrastructures underpinning today's digitally mediated public spheres (thereby drawing attention to the materiality of practices too often idealized as 'purely' communicative)[3] and subject them to processes of public debate and re-making (Kelty, 2008, pp. 27–63). The questions structuring the present paper, however, focus on the *work* that these processes involve for the movement's participants: What (aside from the general commitment just noted) mobilizes these individuals to engage – for no payment, and with no prospect of exclusively controlling the fruits of their efforts – in the labour of writing and maintaining computer code? How are their collective efforts co-ordinated? How do these mobilizing factors and forms of organization relate to the pursuit of profit by capitalist enterprises, in the software sector and beyond? The answers to these questions show that free software production is also intimately connected – though in complex and ambiguous ways – with the dynamics of contemporary capitalism. On the one hand, the movement emerged from a resistance to proprietary control of the electronic architectures on which capitalism increasingly relies, and an antagonism towards capitalist social relations that is reflected in practices and norms of inclusion, co-operation, volunteering and sharing. On the other hand, free software projects have been made to serve as laboratories for the development of organizational and managerial techniques tailored to a project-based capitalist economy, and the hacker ethos animating these projects has become distilled into a particularly pure form of the 'spirit' of this capitalism (Boltanski & Chiapello, 2005 [1999]). Inevitably, then, property relations have re-emerged in business models organized around 'open' innovation. Founded on mechanisms that curtail strategies of profiting from exclusive rights in one kind of intangible asset (authorial products), only to consolidate strategies of profiting from control of another (reputational capital), these relations institute social asymmetries that are in their turn underpinning new modalities of exploitation. At the same time, publicly legislated copyright norms are reversed here only by privately legislated 'copyleft' norms, which in turn rely on legal techniques that advance commodification processes rather than impeding them.

Code, copyright and commerce

In any history of the free software phenomenon, 1985 marks a critical juncture and Richard Stallman emerges as a central character. In 1984, Stallman was still associated with the Artificial Intelligence Laboratory at MIT, where over the previous decade he had established a formidable reputation as a brilliant programmer. Committed to the hacker principles that software should be freely shareable and that programmers should be able to build upon each other's work, he opposed the creeping tendencies towards software

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

propertization (the assertion of proprietary rights over software, such that source code could not be accessed and the distribution of modified versions was prohibited) that were becoming evident with the rise of the PC market and the emergence of firms specializing in the production of PC software products. Driven by these commitments and antipathies, Stallman had by 1984 embarked on the project of writing an operating system, dubbed 'GNU', which would be released as 'free' software (Stallman, 2010, pp. 25–36). For Stallman, free software was defined by the core user freedoms that were already taken for granted within the hacker community: the freedoms to use and modify the software, and to make and distribute copies and modified versions. Software was not free if any of these freedoms was in practice unavailable because the user had no access to the program's source code. (A program's executable object code is effectively only readable by computers. Access to the human-readable source code is therefore a precondition of the easy analysis and modification of software.) Software was also not free if the user had to pay for, or even ask for, permission to exercise any of the essential freedoms in relation to either its source code or its executable code. It followed that proprietary software was not free: subject to certain exceptions, intellectual property (IP) law (including copyright law) requires licences to be obtained from a right-holder for any use falling within the scope of the right-holder's exclusive rights, usually in return for royalties. Yet Stallman saw no contradiction between his free software definition and charging for *copies* of software programs: indeed his practice had always been to make copies of his code available to anyone who requested them – for a price but with no restrictions on use – and to invite feedback from users. In October 1985, he sought to institutionalize this process in relation to his GNU project, setting up the Free Software Foundation (FSF) to co-ordinate a growing band of follower-volunteers to write GNU code, distribute copies and incorporate suggested modifications.

In a context of legislative changes and judicial rulings consistently favouring the demands of proprietary software producers for strong IP protection,[4] the FSF had to find ways to advance its philosophy of software freedom from within a legal framework that seemed fundamentally at odds with that philosophy. One way was to encourage volunteer programmers to donate their software (with its source code) to the public domain, thereby renouncing their copyrights altogether. Another was to encourage programmers to use their copyrights to give legal effect to free software norms by releasing their software to the public subject to permissive general licences (i.e. licences granting all prospective users all of the essential software freedoms, for no royalty). From a starting-point of fundamental opposition to the idea that intellectual property rights (IPRs) could subsist in software, the former was the preferable strategy, but neither method of release could ensure that donated or licensed software remained free for users downstream of the first recipient. In donating code to the public domain, the donor disclaims any copyright in it, and in so doing pledges to relinquish the legal leverage s/he would otherwise

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

have had over users of it by virtue of that copyright. However, it remains possible for anyone else to take the donated material, add enough to it to create something that qualifies as an original work for the purposes of copyright law, claim copyright in that derivative (but 'original' in the legal sense)[5] work and thereafter exercise that copyright in conventionally restrictive ways. The same applies to permissively licensed code.

To forestall this re-propertization of code acquired as free software, Stallman devised an ingenious strategy for licensing software copyrights so as to guarantee that the licensed software and any derivatives remained free for all users, not just the immediate licensee. The term 'copyleft' reflects the recursive dimension of this strategy, which was adopted by other free software activists from the late 1980s once a template copyleft licence drafted by Stallman and others had been made available for general use as the GPL.[6] Licences based upon the GPL are non-exclusive, perpetual (i.e. subsisting for the duration of the copyright in the licensed work unless breached by the licensee), applicable worldwide and royalty-free. They permit any user to run, copy and modify the licensed software: the essential freedoms that define free software according to the FSF.[7] GPL-based licences also permit verbatim copies and modified versions of the licensed software to be 'conveyed' (propagated in a way that enables others to make or receive copies) subject to certain conditions, notably that source code be made available as well; there is no restriction on charging for copies, support services or warranty protection. Where modified versions are conveyed, the derivative work must identify the modifier and indicate when the changes were made. Most controversially, the GPL's copyleft clause stipulates that such downstream products must also be licensed under the same (GPL) terms. Effectively, then, a GPL-based licence is a legal mechanism for ensuring that any software which is acquired as, derived from or intermingled with free software can only itself be distributed to other parties as free software: it imposes a regime of reciprocal sharing.

The keys to understanding free software development are to be found not only in licences that liberate individuals to use and modify code as they wish for their own purposes, but also in those hacker norms that encourage individuals to participate gratis in collective projects with a view to producing integrated products collaboratively. A project is typically initiated by posting a first version of a software product to a host website[8] and inviting users/developers to upload additions or modifications – all under permissive general licences such as the GPL. Where additions or modifications are accepted, the project's initiators incorporate them and re-post the new version, thereby launching another phase of the same process. The best known product of this process is undoubtedly still the GNU/Linux operating system, so called because it combines an operating system kernel named Linux with GNU operating system tools. Initiated by Linus Torvalds (then a computer science student at Helsinki University) in 1991, it grew rapidly with the explosive growth of internet use from the 1990s, and as it became clear that its

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

decentralized development model was yielding an operating system that was at least as good as, if not technically superior to, the available proprietary equivalents, it increasingly became the focus of efforts to commercialize it. By the mid-1990s, companies such as Red Hat and Caldera were building profitable businesses out of packaging the components of the system onto CDs or downloadable files, and/or offering technical support and customization to its users.

Considered in terms of conventional economic analysis of the bases of individual and collective action and social co-ordination, the technical and commercial success of GNU/Linux was unexpected. Why were good programmers choosing to allocate time and energy to a project for which they knew they would not be compensated and on which they knew others would free-ride? Once they did commit themselves, how were the combined efforts of thousands of geographically dispersed individuals able to yield functioning products in the absence of the co-ordinating mechanisms typical of firms (hierarchical chains of managerial command) or markets (contracts for services)? In 1997 Eric Raymond, a free software developer and writer on hacker culture, explained these 'miracles' (Raymond, 2001b, p. 22) as arising from straightforwardly utility-maximizing behaviour by individuals, organized by reference to a few basic principles. Hackers, he argued, are no less self-interested than innovators in any other field but, for them, the prospects of ego-satisfaction and peer recognition are more powerful drivers of creative effort than the hope of direct monetary reward. The interactions constituting the Linux project amounted in effect to a market in these intangible goods, whereby intellectual outputs (chunks of code) were exchanged, not for royalties, but for acclaim and credibility. The project's 'great babbling bazaar of differing agendas and approaches' (ibid., p. 21) structured this market by bringing together a good initial software design with a modular structure, permissive licensing, effective use of the internet to harness the attention and brainpower of a globally distributed talent pool, the co-operative customs and 'play drive' native to hacker culture, and a leader with the programming skills to judge the quality of others' contributions to the project and the communication skills to 'keep his hacker/users constantly stimulated and rewarded' (ibid., p. 30). The result was 'a self-correcting spontaneous order more elaborate and efficient than any amount of central planning could have achieved' (ibid., p. 52). By 'maximiz[ing] the number of person-hours thrown at debugging and development' (ibid., p. 30), the Linux project had incubated a highly efficient process for identifying and fixing bugs with which the monoliths of the proprietary software world could not compete.

The impact of Raymond's intervention on the free software movement and its relationship with the commercial software sector was immense. Citing Raymond's essay as an influence on its decision, Netscape – a proprietary firm then locked into battle with Microsoft for control of the Web browser market – released the source code of its browser on the internet in 1998 under a

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

permissive general licence (lacking a copyleft clause). Its aim was to attract volunteers to work on the development of the browser as a collective project. Later that year, tensions that had been brewing within the free software movement over the movement's relationship with the proprietary software sector finally resulted in a split. Disaffected by what they perceived to be the dogmatism of Stallman's free software philosophy, self-styled 'pragmatists' within the movement – led by Raymond and Bruce Perens – launched the Open Source Initiative. This had the avowed aim of 'dump[ing] the moralizing and confrontational attitude that had been associated with "free software" in the past and sell[ing] the idea [of collaborative non-proprietary software development] strictly on the same pragmatic, business-case grounds that had motivated Netscape' (Open Source Initiative, n.d.). Proponents of what was now called 'open source' software (as distinct from 'free' software[9]) then began actively to court venture capitalists and large corporations, pleading the technical superiority and lower cost of software produced on the basis of openness and sharing, but eschewing a rigid adherence to 'copyleftist' ideology. Ever since, open source proponents have formed the numerically dominant camp within what is now generally referred to as the 'free and open source software' (FOSS) movement, and Raymond is widely regarded as the founding father and chief spokesperson of this branch of the movement – as important to its self-understanding as Stallman has been to that of the free software stem.[10]

For present purposes, the most noteworthy result of the FS/OS split was that the free software philosophy became transformed in the eyes of investors into nothing more than another business model: the 'open source' model. From this point on, a swelling chorus of voices – including those of Linus Torvalds, the technology publishing group O'Reilly Media, influential publications such as *Wired* magazine, the CEOs of FOSS distribution firms and a growing band of academic researchers – joined Raymond's in offering plausible accounts of the 'success of open source'[11] that explained the phenomenon of open innovation in the idiom of conventional economic analysis. Research in the economics of innovation was called upon to explain why permissive licensing strategies and hacker norms are in fact more *efficient* mechanisms for structuring software development than IPR-based exclusion. The propensity of exclusive IPRs to throttle product development in the information industries, or to absorb capital that might otherwise be productively deployed elsewhere, is well documented in this literature (see e.g. Heller, 1998, 2008). By opening up access to the intangible raw materials of innovation – source code, in the software development context – permissive licensing, it is said, can avert the 'tragedy' of enclosure,[12] forestalling the possibility of hold-ups by the owners of monopoly rights and obviating the need to duplicate others' efforts. Further, free-riding – using FOSS code but contributing nothing to its development – has been characterized as capable in fact of enhancing its value: by causing it to become a de facto standard, the network effects (positive

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

externalities) generated by free-riding on FOSS products can increase the market share of the corresponding commercial distributions.

A more far-reaching set of implications of the FOSS development model has emerged from the organizational and management science literatures on FOSS. Researchers in these fields have devised economistic explanations for the motivations that drive voluntary participation in FOSS projects and the mechanisms by which participants' contributions are co-ordinated. Echoing and expanding on Raymond's observations, 'incentives' to participation have been said to include opportunities to solve personal or business IT problems, hone skills and earn reputational benefits that can be cashed out in the form of job offers, access to venture capital and the ability to attract other excellent programmers as collaborators (see e.g. Gambardella & Hall, 2006; Roberts *et al.*, 2006; Shah, 2006; Tirole & Lerner, 2002; von Hippel & von Krogh, 2003). Another driver of participation is said to be the possibilities for self-expression (or simply fun) that coding offers. Unlike developers working for proprietary software enterprises, FOSS developers can choose what problems to work on, and how; and their work is available for their peers to see, celebrate and (most gratifying of all) to use. Steven Weber identifies as their major motivation the simple desire to create code that not only functions to achieve particular technical goals but is openly affirmed as functioning 'elegantly' (Weber, 2004, p. 136). The websites that host FOSS software and projects are said to play an important role here: they enable developers to display their talents, debate the merits of each other's work and give or withhold recognition.

How the distributed labour invested by the contributors to FOSS projects is co-ordinated has proved to be more difficult for organizational and management scientists to fathom. On one level the internet is seen as part of the solution to this problem, in that it enables geographically dispersed participants to communicate easily, instantaneously and cheaply. On another level the vigorous discussion it enables is seen as increasing the likelihood of conflicts between contributors – e.g. over the criteria informing assessments of a contribution's 'elegance', over how contributions should be attributed and over when defecting from a project to launch a competing one ('forking') is legitimate (ibid., p. 89). Yet researchers report that the process of project development is not in fact anarchic.[13] Failure to comply with norms or decisions that are widely perceived within the community to be authoritative is sanctioned, albeit in informal ways – ranging from 'flaming' (public condemnation) to shunning (refusals to co-operate with the wrongdoer thereafter) – but more importantly, the impact of non-compliance is minimized in advance by strategies of project modularization: '[a] well designed program has modules whose design limits interaction between modules' (Tuomi, 2002, p. 175; see also Baldwin & Clark, 2006) such that if one module becomes beset by irresolvable conflicts the others can proceed regardless. Further, most conflicts do prove amenable to resolution, invariably by recourse to generally recognized, albeit uncodified, norms establishing

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

quality standards organized around notions of technical efficacy; and where these standards prove inconclusive, conventions have emerged to determine where ultimate decision-making authority resides. Even in those projects that are overseen by formal governance institutions, authority on issues of technical design generally rests with those who have earned the respect of their fellow programmers as a result of the technical merits of their programming work, their responsiveness to the contributions and comments of others, and their ability to set achievable goals for the project (Weber, 2004, pp. 167–8). Significantly, Eric Raymond regards these conventions as rooted in a conception of ownership that can be characterized as 'an abstraction of [the] territoriality' (Raymond, 2001b, p. 98) underlying land claims under Anglo-American common law. Ownership in this customary sense is regarded as legitimately acquired in one of three ways: by founding, and remaining actively engaged in the development of, the project; by transfer (via explicit public declaration) from the founder (or from one of his 'successors in title'); and by a community-supported takeover of an abandoned project (ibid., pp. 74–5). Viewed in this way, hacker norms contest liberal-legal conceptions of the bases of property rights (investment, alienation and adverse possession) only to replicate them at a deeper level. Thus entitlements to fork, or approve changes to, a project are in fact held exclusively by those who are accepted by the community of contributors as the project's legitimate owners (ibid., p. 73).

Understanding FOSS

Thus far, two powerful – but, I will argue, ultimately unsatisfactory – frameworks have emerged for understanding the relations between FOSS, capitalism and the critique of capitalism. One is Manuel Castells's sociology of the 'information age', connecting transformations in computing and communications in the 1970s to broader socio-economic changes, and ultimately to the emergence of a 'network' society to which information flows are central. Volume one of Castells's influential trilogy on these themes (first published between 1996 and 1998) focused on the technological changes and forms of economic restructuring involved with what Castells characterized as a shift from an 'industrial' to an 'informational' mode of capitalist development.[14] *The rise of the network society* (Castells, 2000) represented this shift as exemplifying two competing trajectories. The more firmly established of the two, Castells argued, deployed new information and communications technologies (ICTs) towards the end of short-term profitability and in ways that left old economic power structures intact – i.e. 'to save labour, to subdue unions, and to trim costs, rather than to improve quality or to enhance productivity by means other than downsizing' (ibid., p. 265). Yet Castells also seemed to discern another tendency, more potential than actual but nonetheless immanent within the most advanced sectors of informational capitalist

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

economies by the mid-1990s. Propelling this trajectory were innovative 'network enterprises' that were profitable yet non-hierarchical and harnessed ICTs to support organizational forms and cultures that privileged worker autonomy and consumer engagement; Castells later appeared to identify 'open source' enterprises as exemplary of this 'new' economy gestating within the old (Castells, 2004, p. 158). If fully unleashed, his analysis suggested, the values sustaining these enterprises could lead to the emergence of a network economy in the best sense: globally integrated (though also decentralized and differentiated), fully inclusive (capable of engaging every human mind as a direct productive force (Castells, 2000, p. 31)) and therefore marked by sustained growth, enhanced experiences of work, low unemployment and higher living standards everywhere.

Of particular interest in the present context is that Castells eventually came to see each of these trajectories as linked with a distinct 'spirit' of informational capitalism.[15] In Max Weber's famous formulation, the spirit of capitalism is the set of values motivating action conducive to capital accumulation: Weber himself saw these values as organized around a 'Protestant' conception of work as a vocation to be pursued with a dogged attention to duty (Weber, 2001 [1905]). In *The hacker ethic and the spirit of the information age* (Himanen, 2001), Castells's collaborator Pekka Himanen identified two competing spirits of the informational capitalism whose contradictions Castells had already mapped in *The rise of the network society.* The dominant one, Himanen argued, is directly descended from the Protestant ethic, though reformulated for the digital age. It comprises an implacable orientation towards the achievement of calculable goals, in relation to which all available time must be optimized and all available energy expended: individuals and enterprises that operate in accordance with this ethic will function like the very computers they ostensibly programme. Opposed to this is the hacker ethic, organized around a conception of work as a passion to be realized in community with others, and oriented towards 'the imaginative use of one's own abilities . . . and the giving to the world of a genuinely valuable new contribution' (Himanen, 2001, p. 141). In endorsing this alternative spirit of informationalism, Castells (2001, p. 167), seemed to affirm the centrality of the hacker ethos to the incarnation of the new, improved capitalism already heralded in *The rise of the network society.* Overall, however, his work on this theme suggests a commitment to a version of technological determinism (Castells, 2000, p. 257),[16] evident for example in the suggestion that although hackers may have initiated the IT revolution underlying 'the rise of the network society', they never controlled its subsequent development because 'once programmed [by social actors], information networks, powered by information technology, impose their structural logic on their human components' (Castells, 2001, p. 167). In short, although Castells intuited that the hacker ethos could exert some critical traction on the workings of the capitalist system, he ultimately represented it as a function of the 'technological paradigm' currently underpinning that system: 'informationalism'.

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

Ostensibly opposed to all forms of determinism is another influential perspective on FOSS, one also propelled to contemporary prominence by a widely read trilogy (Hardt & Negri, 2001, 2006, 2009). For the 'autonomist' Marxist – arguably better described as post-Marxist – internet theorists who take their bearings from the work of Antonio Negri, global electronic communications environments are exemplary sites of the 'immaterial' (cognitive and cultural) labour on which profits increasingly depend under the post-Fordist conditions that characterize advanced capitalist economies. The central theme of this literature, however, is that the process by which surplus value is extracted from labour is fundamentally contested. Labour in general, it is said, exceeds all attempts by capital to organize it: it manifests a capacity for insurgency that can transform instruments of capitalist exploitation into resources for opposition, and in ways that are autonomous of political parties, unions and the like. Immaterial labour is said to have a particular potential for autonomous self-organization because, valorizing itself in spontaneously generated co-operative networks, it does not need to be activated and orchestrated by capital in order to become productive. Some autonomist writers have explored how this assumed potential manifests itself online (e.g. Dyer-Witheford, 2005; Terranova, 2000, 2004, Ch. 3). Drawing on this literature, Johan Söderberg has characterized FOSS projects as experiments in 'hacking capitalism': as showing how the knowledge workers of the computer world can self-organize resistance to capitalist control of software production, and co-create alternatives to the infrastructures and networks on which capitalist exploitation in general relies (Söderberg, 2008). On Söderberg's account, then, the libertarianism often avowed by FOSS participants can be dismissed as epiphenomenal. Viewed objectively – that is, in terms of what is necessarily presupposed by the activity ('hacking as a verb') rather than by reference to what its practitioners ('hackers as a noun') subjectively consider themselves to be doing when engaged in it (ibid., p. 5) – FOSS development represents a radical critique of capitalism that is oriented towards its supersession, not merely (as on Castells's account) its repair or improvement. Being open in principle to all who wish to participate, FOSS projects undermine the social division of labour and give power over the trajectory of technological change to the people. Being voluntarily undertaken and collectively managed, they are premised on a rejection of the wage relation and the capitalist organization of the labour process. Being informed by an antipathy to private property-based exclusion, the method by which their products are circulated is incompatible with the commodity form. Above all, FOSS development proceeds from the drive to produce for the sheer joy of it, and to share one's creations freely with others. It thereby advances a conception of labour as a form of play, guided by an aesthetic motivation quite opposed to the urge to use labour power as a means of expanding capital: the aim of creating beautiful artefacts (here, computer programs) as ends in themselves.

Söderberg's account of FOSS production as critical social practice is provocative and insightful, and it has informed the present account in numerous ways. Yet his 'discovery' of the real meaning (anti-capitalist struggle) underlying the subjective meaning (pro-capitalist enthusiasm) most participants ascribe to their involvement in FOSS projects reveals a problem that characterizes autonomist writing in general. If (as the commercialization of FOSS suggests) capitalism *requires* the very practices that are supposed to manifest 'a kind of spontaneous and elementary communism' (Hardt & Negri, 2001, p. 294), and if the meanings the agents of these practices give to them are irrelevant, how is it possible to distinguish evidence of co-optation from evidence of contestation – or a resurgent capitalism from an emergent communism? The same practices seem to lend themselves to radically opposed interpretations, and autonomism offers no principled criteria by which to decide between these (Gill & Pratt, 2008, p. 19). Further, if Castells's structuralist representation of hackers seems mechanistic, the autonomist idea that they form the vanguard of a labouring 'multitude' – 'spontaneously' resisting or routing around capitalism – is positively mysterious. Conceptualized by Hardt and Negri as defining a global subject of revolutionary insurrection, the category of the multitude remains a highly abstract and metaphysical one for which no empirical warrant can be found (Shapiro, 2004). Consequently, autonomism too ultimately marginalizes human volition, in this instance because it represents the actions and intentions of human beings as mere symptoms of transcendent dynamics that lie beyond their comprehension and control.

The premise of what follows is that Castells's and Söderberg's insights about the relationship between FOSS, capitalism and the critique of capitalism are best read in the light of Luc Boltanski and Ève Chiapello's epic study of the business culture associated with the network economy, *The new spirit of capitalism* (first published in France in 1999). While acknowledging that capitalism constitutes a system with its own logic and laws, Boltanski and Chiapello nonetheless seek to integrate into their account of its workings the critiques that human actors direct at it (Boltanski & Chiapello, 2005, p. xviii). Critiques of capitalism, they argue, have historically taken two general forms: 'artistic' critiques (originally stemming from nineteenth-century bohemianism) denounce capitalism as requiring forms of life that are inauthentic and oppressive; 'social' critiques (long associated with the traditional labour movement) condemn it as immiserating and corrosive of social solidarity. The central thesis of *The New Spirit of Capitalism*, however, is that these critiques are always entwined with capitalism even as they oppose it, and that this entwinement is manifested in the 'spirits' that have accompanied successive phases of capitalist accumulation from the late nineteenth century onwards. Since each spirit of capitalism takes shape by co-opting what resists capitalism, each must restrain the accumulation process as well as mobilizing support for it: more particularly, each must represent the forms of life that are linked with the phase of capitalism to which it corresponds as *exciting* for

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

individuals (thereby accommodating the artistic critique), but also as guaranteeing individuals *security* and conducing to the *common good* of society as a whole (thereby addressing the social critique). To these ends, each spirit of capitalism needs mechanisms for its own concretization: the values it advances must be 'inscribed in institutions, bound up with actions, and hence anchored in reality' (ibid., p. 3) – not least by legal rules and other less codified measures such as particular management techniques and organizational arrangements within and between enterprises (ibid., pp. 399–400).

The remaining two sections of this paper seek to show that the FOSS ethos has become distilled into a particularly pure form of the 'new' spirit of Boltanski and Chiapello's title. The latter is tailored to a post-Fordist regime of accumulation, arranged around '*lean* firms working as *networks* with a multitude of participants, organizing work in the form of teams or *projects*, intent on customer satisfaction, and a general mobilization of workers thanks to their leaders' *vision*' (ibid., p. 73, emphasis in original). The spirit of this phase of capitalism privileges the bohemian qualities that generate and sustain the projects on which profitability now depends – qualities such as creativity, openness, informality, adaptability, conviviality, charisma – and it celebrates the 'connexionist' form of life (a life of 'contacts [and] chance encounters' (ibid., p. 312)) that the cultivation of these qualities entails. In a project-based capitalist order, adept connexionists can find both excitement and security in their portfolios of projects, accumulating (social, reputational, informational and ultimately monetary) capital as they segue nimbly from one project to another. Yet their success, Boltanski and Chiapello argue, depends on a form of exploitation that is specific to network capitalism: not only are some more mobile than others, but the immobility of some is *necessary* to the mobility of others (ibid., p. 362). The bad citizens of network capitalism ('networkers') use mobility differentials opportunistically to expand their own networks and reap the benefits yielded by their connections at the expense of others who have helped to create them. The true exemplars of the spirit of network capitalism are its good citizens, connexionists who seek to advance the common good by *compensating* for mobility differentials. These 'network-extenders' share their connections and the resources yielded by these, and foster the potential of everyone linked into their networks at the same time as they cultivate their own. FOSS developers, I argue below, are archetypal network-extenders; more particularly, they have devised norms and mechanisms that institutionalize the network-extender's implicit code of behaviour, thereby 'anchoring the new spirit of capitalism in reality' in uniquely effective ways. Yet these same norms and mechanisms, I suggest, are also embedding new modalities of control over both production and consumption, and extending commodification processes rather than curtailing them. Further, they are proving insufficiently robust to counteract the opportunistic networking that in fact sustains the new economy.

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

FOSS and the new spirit of capitalism

In Boltanski and Chiapello's terms, Richard Stallman's early crusade for free software can be seen as having been fuelled by both an artistic and a social critique of capitalism. Stallman opposed software propertization on the ground that it was producing attitudes and orientations fundamentally at odds with the patterns of collaborative code-writing that had emerged organically among programmers before commercial imperatives changed the rules of the software development game, and that it was suppressing programmers' 'natural' tendencies to borrow from each other in the exercise of their own creativity (Williams, 2002, Ch. 1). He lamented the passing of what to him was an intensely enjoyable decade of playful exploration, alongside fellow programmers at the AI Lab who shared the hacker ethos, of the new possibilities opened up by computing in the 1970s. He resented the economic forces and management practices that lured hackers into the employ of proprietary firms and imposed unethical contractual (especially trade secrecy) obligations and 'ugly' programming conventions on them. He particularly disdained the passivity to which proprietary strategies subjected software consumers, by preventing them from adapting software to their own ends and thereby customizing their own computer equipment, and keeping them locked into dependency on producers (Moody, 2001, Ch. 2). And he saw the freedoms to participate in, and benefit from the fruits of, collaborative programming as important not only for individuals but for society as a whole, 'because they promote social solidarity – that is, sharing and cooperation' (Stallman, 2010, p. 83).

Nonetheless, it is clear from the trajectory mapped in the first section of this paper that although the free software movement originated in an antipathy to the proprietary software sector, the 'FOSS' movement has since become co-opted as an adjunct to that sector. The argument of the present section is that this outcome is unsurprising when considered in relation to Boltanski and Chiapello's account of the associations between capitalism and its critiques. From the end of the 1990s onwards, business analysts and organization theorists were able (thanks in particular to the interventions of Eric Raymond, Bruce Perens and Linus Torvalds) to interpret the avowedly confrontational 'free software' movement as a branch of a fundamentally business-friendly 'open source' movement; consequently, they could – and, as the first section above shows, did – subsume the former within the latter (now dubbed the 'free and open source', or simply 'open source', movement), and read the 'FOSS' ethos as a set of instructions for managing production in every sector of a connexionist market economy. So construed, FOSS projects have demonstrated the viability of the project form as a mode of co-ordinating persons, intangible assets and material arrangements towards the end of capital accumulation, while the norms that hold these projects together have provided the blueprint for a new spirit of capitalism adapted to post-Fordist conditions – a spirit that takes the free software movement's critiques of capitalism as its

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

very ingredients. These points are elaborated briefly in what follows; the next section identifies likely bases for a renewal of critique given these realities.

Boltanski and Chiapello observe that what distinguishes projects from other modes of action co-ordination is that they temporarily assemble disparate actors within a network, 'stabilizing certain connections and making them irreversible' (Boltanski & Chiapello, 2005, p. 105) while they last, and putting in place more enduring links 'that will be put on hold while remaining available' (ibid., p. 104) after they have come to an end. In this way, projects crystallize relatively durable 'forms' – subjects, objects, goals, patterns of production and accumulation of value – that would otherwise be dissipated through the networks constituting social and economic life; at the same time, their transience and mutability render them perfectly adapted to a network logic (ibid., p. 111). In effect, FOSS initiatives have served as laboratories for the development of a 'projective' grammar of social organization that transcends capitalism yet is also amenable to being tailored to its demands. By taking full advantage of digital networks, these initiatives are taken to have shown how projects can have a global reach and involve many dispersed participants without disintegrating or losing focus. Their modular arrangement, together with the meritocratic norms of conflict resolution that proceed from hacker culture, are seen as facilitating virtually horizontal decision-making structures that allow input by any participant with relevant expertise. A particular style of leadership is emphasized as crucial to co-ordinating this input and ensuring its quality. Central to this style is open communication and creativity (though not in the manner of the solitary Romantic genius – FOSS leaders are seen as recombining what they find on the way rather than creating from nothing, and distributing responsibility for innovation through the network rather than monopolizing it themselves); most importantly, FOSS projects are taken to show that effective 'team' leadership requires charisma: the ability to mobilize actors over whom one has no formal power towards a project's satisfactory completion depends on enthusing them with one's vision (Giuri *et al.*, 2008).

FOSS projects thus exemplify the very approach to organization and management that business analysts have, since the mid-1990s, perceived as crucial to profitability in a reticular economy (on 'visionary' leadership generally, for example, see Bennis & Biederman, 1998). Yet two features mark them out as distinctive relative to straightforwardly commercial co-operative ventures. The first is the heavy involvement within them of unpaid volunteers. The second is the conception of property that animates them: those who own the IPRs in FOSS project outputs see their exercise of these rights as governed by their role as 'stewards' of the project (O'Mahony, 2005) and guardians of its constitutive norms of sharing and collaboration. In Boltanski and Chiapello's terms, however, these non-market and anti-exclusionary aspects of the FOSS phenomenon are best regarded, not as positioning FOSS beyond the accumulation imperative altogether, but as expressing variants of (respectively) the artistic and the social critiques that, as such, were ripe for incorporation

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

within the new spirit of 'leftist' capitalism (Boltanski & Chiapello, 2005, p. 202) – a laid-back, cool capitalism – that was beginning to emerge in the 1990s.

The artistic critique embedded in the hacker ethos has proved particularly easy to assimilate because it is readable as a development of the themes of the 1960s counter-culture, themes that had themselves been invoked in the 1980s to justify the upheavals consequent upon the economic shifts of the 1970s: organizational restructuring, 'flexible specialization', labour casualization and the commodification of objects and services (including intellectual/creative objects and services) that had previously circulated beyond the market. In so far as these changes were 'sold' to those affected on the basis that both production and consumption would thereby become occasions for experiences of autonomy, personal development, creativity and authentic human relations, the erosion of the production/consumption divide was bound eventually to seem both natural and inevitable. This erosion is what FOSS development has primarily come to represent. As a form of voluntary and self-organizing collaborative endeavour that participants experience as pleasurable but that can also be seen to yield socially useful (and saleable) outputs, it erases completely the lines between work and play (or art), production networks and friendship networks, economic exchange and social interaction. From the hacker-participant's perspective, this erasure may well be experienced as the subsumption of the first term of each of these poles into the second. But from the business strategist's perspective, the lesson taught by the hacker's participation has been that exactly the reverse is possible: not that all labour can become play, but that all play can become labour.

Admittedly, the social critique presupposed by the hacker norms governing FOSS development has proved more difficult for capitalist enterprises to absorb. These norms extol good (in Boltanski and Chiapello's terms, network-extending) behaviour that is conducive to the alleviation of inequality and the building of solidarity, and condemn 'evil' (networking) behaviour that is self-regarding and opportunistic. By providing that anyone may contribute to a FOSS project, they open up prospects of developing skills and benefiting in other ways from connections with other programmers. Permissive licensing ensures that all may help themselves to FOSS outputs, thereby combating the forms of exclusion and asymmetry that are bound up with the conventional exercise of IPRs.[17] Copyleft clauses in permissive licences are designed to give legal force to the norm that no one may seek to recreate IP-based asymmetries in relation to code first licensed as free software, and so guard against this form of opportunistic networking. Co-operation is encouraged because it is the glue that holds together the projects that yield the most socially useful innovations. Nonetheless, the norm of individual software freedom, legally secured by every FOSS licence, entails the recognition that each individual is at liberty to fork a project (to take a project's source code base and launch an incompatible project from it), and to distribute modified versions of project outputs which are compatible with earlier versions. These liberties enable project participants to depart from a project in order to apply their skills elsewhere: they thereby preserve contributors' mobility, an asset deemed especially necessary in case

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

project leaders show themselves to be overbearing, uncommunicative or incompetent. Finally, attribution norms ensure that all contributions are accredited[18] so that no one's work goes unacknowledged.

And yet, however opposed to exploitative economic relations they may seem, these norms are now giving a solid shape to a spirit of post-Fordist capitalism that, on Boltanski and Chiapello's own account, was only dimly discernible in the management discourses of the 1990s. From the mid-2000s on, this spirit has been elaborated in a business literature organized around the idea of an open and 'collaborative' capitalism that relies on communication as the crucial source of wealth, but also on communication media that enable 'everybody' to contribute to wealth creation and share in its fruits (see e.g. Anderson, 2009; Grossman, 2006; Howe, 2008; Leadbetter, 2009; Shirky, 2008; Surowiecki, 2005; Tapscott & Williams, 2008). Two lessons – about the folly of hoarding and the wisdom of branding – are repeatedly expounded in this literature's prescriptions for how (legitimate) wealth could be possible, now, for firms and individuals alike. Giving assets away for free, it is said, can generate more value than locking them up or pricing them – for example, where innovation in a firm's ecosystem is sparked by doing so, thereby creating value for the firm,[19] or where sharing can establish credentials and build goodwill with potential partners or customers (Tapscott & Williams, 2008, pp. 302–3). Yet the dominant view is that sharing should never displace, but should always complement, proprietary approaches: 'the future of mass collaboration lies in hybrid models, in which participants share and appropriate at the same time' (ibid., p. 303).[20] Trends in the software sector are consistent with this view. Since the end of the 1990s, both free software and open source software development methods and projects have attracted steadily increasing interest from formerly wholly proprietary firms, notably IBM, Sun Microsystems, Novell, Oracle, Hewlett Packard and Nokia. To this day, commercial firms continue to support FOSS development by donations of money, legal advice, business consultancy services, training, equipment, code and staff time (see Berdou, 2011, Ch. 4–5). These donations are clearly motivated by strategic considerations. IBM's support of the Linux project, for example, was prompted by its realization that the aggressively proprietary strategies on which it had previously relied had failed to produce an operating system that could disturb the market dominance of established rivals like Microsoft. Its investments in the project (a fraction of the cost of developing its own system) have enabled it to profit from sales of Linux-based services and hardware and undercut competitors who charge for operating system software.

The same investments have also greatly improved IBM's public image, because Linux is now a powerful brand as well as a formidable operating system. The centrality of conventional trademark rights to FOSS business models is readily understandable against this backdrop.[21] Trademark rights are IPRs subsisting in signs (names, logos and so on) that function as indicators of trade origin for the products to which they are applied. A trademark helps to position a product in a crowded market-place (and media space) and so assists the process of building and maintaining a brand: its holder can prevent

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

unauthorized traders from marketing their own products under the protected sign, thereby filching the holder's goodwill. Significantly, even the Free Software Foundation has no objection in principle to the conventional exercise of trademark rights. FOSS project leaders and foundations frequently register trademarks with a view to co-ordinating the process of monetizing their projects' reputations: these marks can be licensed (in return for royalties) to enterprises wishing to identify their goods and services with those projects (see e.g. Varghese, 2005).

The perception that the most important currency within the FOSS universe is reputational capital in turn enables an interpretation of the hacker norms that organize FOSS development as oriented towards nothing other than structuring the competitive struggle to acquire it. It is by reference to the imperative to accumulate reputational capital that Eric Raymond, for example, resolves the apparent contradiction between his picture of the FOSS project owner as a kind of yeoman homesteader and the hacker norm that project owners should act as stewards of 'their' property for the project's contributors. For Raymond, this norm makes sense because contributors' investments in a project give them a stake in it as well: at stake for them are the 'reputation returns' (Raymond, 2001b, p. 103) yielded by their efforts. The strong social pressure against forking projects can, he argues, be explained in these terms (ibid., p. 73), as also can the widespread recognition within developer communities that products emerging from forks should be distributed under a different name so that they are distinguishable from the parent version: illegitimately forking a project, and publicly distributing unauthorized modified versions of it that purport to be authorized versions, attract condemnation because they expose both the project owners and its individual contributors to the risk of reputational damage. On Raymond's reading, then, the FOSS ethos – notwithstanding that it appears to reject property – is itself oriented towards the propertization (albeit informal) of an intangible asset: not code but reputation, both collective and individual.

Brand equity has become a significant source of shareholder value since the 1990s, as digital ICTs have both increased global information flows and enabled financialization to drive economic growth (Willmott, 2010), and business analysts generally have become attuned to branding's growing importance. To these observers, Raymond's reading is apt to suggest that the true secret of FOSS enterprises' success in attracting and retaining expert volunteers is attribution. Attribution can be seen as incentivizing creativity, partly because of the psychological benefits that recognition brings, and partly because it facilitates mobility: by 'signalling human capital' (Fisk, 2006, p. 53), countable credits reduce employers' search costs in contexts of rapid labour turnover and thereby enable their recipients to find their way into (better) paid employment. So if one key lesson taught by free software activists' social critique of capitalism is that valuable outputs should be shared even with those who have not paid for them, the other is that valuable inputs can be harvested even from those who have not *been* paid for them if the latter are offered routes

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

to profitable self-commodification: equipped with a portfolio of 'braggables [they] can take credit for' (Peters, 1997; see also Botsman, 2012), contributors can counteract structural economic insecurity by monetizing their own human capital.

New economy; new critiques

It follows from the foregoing that the distinctions to which the mythology of FOSS appeals – between exclusivity and openness, the sale and the gift, the proprietary and the non-proprietary – are blurred in practice. Not only are 'proprietary' commercial enterprises increasingly resorting to 'non-proprietary' strategies, but exclusivity in some form – if not via the copyrights in software, then via the trademarks under which FOSS products and services are sold and the credits by which individual developers track and measure their achievements and market their reputations – underpins profitability across the board. Whether the norms embedded in FOSS production practices can continue to be associated with *legitimate* profit creation – and so remain as the ethical foundation of contemporary capitalism – is, however, an open question. A paradox can be discerned here which can be simply stated. On the one hand, it is not at all clear that FOSS licensing regimes are robust enough either to sustain the enterprises that have been built around them, or to institutionalize the norms that constitute the hacker ethos. On the other hand, if they do turn out to be capable of yielding the requisite degree of commercial certainty in the long term, renewals of the social and artistic critiques of capitalism will inevitably be prompted by FOSS business models themselves.

Questions abound, first of all, as to the legal status of FOSS licences. How should they be characterized in relation to the available legal categories of licence? In what circumstances, if at all, can they be revoked? Is the viral effect of copyleft clauses legally enforceable, and if so, on what doctrinal basis? What law applies to these licences in a context where users could be based in multiple jurisdictions? What legal implications follow from distributing products whose elements have been licensed under incompatible licences? What legal and commercial implications follow from the disclaimers of warranty and limitations of remedy contained in most FOSS licences? The lack of clarity regarding these and other fundamental issues (see e.g. Hass, 2007; McGowan, 2001; Nadan, 2002; Narodick, 2010; Loren, 2007) reveals the essential fragility of the licensing strategies peculiar to FOSS communities. To a large extent, in fact, the commercialization of FOSS has proceeded on the basis of a series of gentlemen's agreements and tacit understandings. If these were to unravel, FOSS licences could play no significant role in structuring the software economy thereafter. Similar uncertainties bedevil their 'free culture' analogues (such as Creative Commons licences) albeit to a lesser extent (Dusollier, 2007).

Even assuming that they are legally valid, open licences (FOSS licences and their derivatives) have significant limitations as mechanisms for institutionalizing

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

the more striking features of the hacker ethos. Concretizing only the individual freedoms to use and modify licensed material and distribute copies and modified versions, they cannot be deployed to enforce the other-regarding hacker norms of sharing and co-operation.[22] So although these latter norms could potentially serve as resources for challenging opportunistic profit-seeking, they lack a firm institutional underpinning. Meanwhile, the liberties granted to users by open licences are real, but paradoxical. As liberties, they carry no restrictions on the functional goals towards which users may choose to direct the further development of licensed material – goals that could be freedom-denying in other ways.[23] At the same time, copyleft licences – requiring those licensees who do choose to share their modified versions to share *alike* – have been aptly characterized as imposing 'servitudes' (Van Houweling, 2008) on users, obligations that 'run with' the licensed object as it moves from one user to another (Radin, 2000, pp. 1132–3).[24] The terms of an open licence are physically encoded in the digitized object itself, so that the latter's conditions of use (together with other details, such as attribution information) move with it as it circulates online; when a new user encounters the object, the embedded code automatically indicates the applicable terms. In so far as they thereby mark digitized information products with servitude-type restrictions that bind users even though consent to them is effectively absent, copyleft licences are no different from the *über*-copyright end-user licences resorted to by the most aggressively profit-seeking firms in the information industries.[25] In other words, the same technique that uses copyright to limit copyright is also being used to make it broader, e.g. by means of 'running' pre-emptions of important defences to copyright infringement (Radin, 2000, p. 1133).[26] Further, the technique cannot be legally valid in one context without also being valid in the other; and in either case validity will, of legal necessity, depend on licence terms being readable as delivering economic benefits to licensors considered as utility-maximizing rational actors.[27] Copyleft licensing is thus parasitic on the very licensing model that the FOSS movement was set up to oppose (Madison, 2004; Elkin-Koren, 2005; Narodick, 2010) – a licensing model that is, moreover, crucial to the information industries' overarching strategy at present: to track and monetize every move made in lives increasingly lived online, and to achieve this dominion by means of legal rules that are formulated privately, and applied and enforced by automated technical systems that are privately designed.

Clearly, FOSS licensing is oriented towards a radically different goal than that of proprietary licensing regimes: not profit maximization through the global enclosure of the common, but the global common good through the construction of a global software commons. It is for this reason that the free software movement also seeks to route around the states and international organizations that enact public copyright laws, anchoring the hacker ethos instead in made-to-measure software freedom codes that have been engineered from the bottom up by hacker communities. The guiding idea is to mobilize the power of open networks against those forces that currently aim to propertize them and the

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

content flowing through them: to pit a transnational hacker movement against an enclosure movement that is similarly indifferent to national boundaries and state-political accountability (on this see Dusollier, 2007). Yet this is to endorse the use of under-the-radar techniques of governance to regulate public access to and use of software and networks – techniques that, as shown above, can just as easily be used to achieve contrary goals. And since ultimately the enforceability of FOSS licences depends on national courts and national laws, the notion that this movement could trump the state-political instance through the spontaneous self-generation of a normative order 'beyond' the state is arguably illusory (Marrella & Yoo, 2007; cf. Teubner, 2004).

Assuming, however, that the legal and commercial uncertainties associated with free licences can be overcome, two further sources of instability in the ethical foundations of the new economy are foreseeable. The first emerges from the fact that FOSS-based business models are beginning to appear at least as exploitative as those associated with closed source software. If '[t]he networker has succeeded when, at the end of a project, something [not produced by him] can be attributed to him and publicly associated with his name' (Boltanski & Chiapello, 2005 [1999], p. 359) then the activities of enterprises that commercialize FOSS outputs arguably qualify as paradigm cases of the form of exploitation specific to network capitalism. Yet by sharing their work under FOSS licences, volunteer developers relinquish the sole leverage that they would otherwise have against such firms: the power to veto the commercial use of the licensed code. So far, few FOSS activists have foregrounded this as a problem. It is deeply embedded in the philosophy of both the free software and open source branches of the movement that those who distribute FOSS products should be encouraged 'to charge as much as they wish or can' (Stallman, 2010, p. 65). The pursuit of profit by this means is seen as a 'good and legitimate activity' (ibid.), and there is no norm requiring that contributors receive remuneration, although the Free Software Foundation (FSF) encourages distributors to donate some of their profits to it or to other free software development projects. The concern that some consumers might be priced out of the market for distributions is recognized as valid, but is countered by the argument that – FOSS being what it is – users who cannot afford the price demanded by a particular distributor will always be able to acquire the software from some other source for a price at or near zero (although software unaccompanied by the high-quality technical support typically provided by distributors is effectively only useable by adepts).

Few FOSS developers harbour personal objections to the commercialization of products incorporating their own unpaid labour. Yet if reputational incentives are as powerful in the FOSS universe as has been suggested, these same developers also imagine that their chances of (better) paid employment will increase on the strength of high-quality FOSS project work that can be attributed to them. In fact, the availability to software firms of pools of expert volunteer labour is highly likely to exert a downward pressure on wages and worsen conditions of employment for programmers generally (Söderberg, 2008, p. 40), and firms' recourse to open source licensing can lessen the

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

mobility of their own employees.[28] Likewise, the circulation of zero-priced software products alongside equivalent products that are available for a price on the software market may seem to benefit consumers, but its effect on the market for programming labour can only be deleterious for programmers – and ultimately for FOSS communities, whose members rely on the wages they can earn as programmers to subsidize the FOSS work they undertake in their spare time (ibid.). In short, it is difficult to find in the legal and ethical norms governing FOSS development and commercialization adequate guarantees of security for those producers whose participation is required to sustain the business models that FOSS has inspired. The same can be said of other contexts to which these norms have been 'ported'. In the cultural industries, too, free or under-compensated creative labour is increasingly being enlisted in ways analogous to the mobilization of volunteer programming expertise in the software sector,[29] and the precariousness of cultural workers is arguably being intensified thereby. A revival of the social critique, exposing contradictions in the values of sharing, collaboration and equality to which the new spirit of capitalism appeals, therefore seems timely; indeed it is already underway (see e.g. Burston *et al.*, 2010; Scholz, 2013).

A second source of likely future upheaval in the new (software) economy is that FOSS projects preserve rather less room for authentic human relations, spontaneous self-organization and creative work than the mythologizing would suggest. Commercial enterprises can annex and manipulate FOSS projects in which they become involved. Donations of staff time in particular can help to position FOSS developer communities as a firm's 'complementary' (as distinct from its core) assets: when employees can gain access to these communities, they can influence the direction of project development and legitimize the firm's commercial exploitation of project outputs (Dahlander & Wallin, 2006; Fosfuri *et al.*, 2008). The popular image of networked innovation as proceeding from an undifferentiated 'hive mind' is also in tension with the realities of project governance which, as indicated in the first section of this paper, tends to disintegrate further the fragmented labour processes, and mimic the (property- and expertise-based) power structures, characteristic of the post-Fordist enterprise;[30] invariably, too, these structures reflect multiple intersecting social inequalities,[31] and often they are held together by a form of charismatic domination (Dafermos, 2012; cf. Arvidsson, 2009) that is difficult to square with the artistic critique's commitment to liberation. Most importantly, the premium placed on the competitive pursuit of 'reputation returns' is at the very least in tension with the hacker spirit's privileging of creativity and community. In a reputational economy, creative production becomes a means to the end of forging a publicly recognizable identity: the goal is not so much to produce a body of work that can take on an existence beyond oneself as to become an entrepreneur *of* one's self by associating as much activity as possible (preferably including that of others) with one's name. If unchecked, this will yield a culture in which (self-) promotion takes priority over production; it is also liable to obscure the collective effort that sustains

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

every project, erode mutual trust and loyalty, and ultimately undermine the FOSS spirit itself. A revival of the artistic critique, exposing how ideals of autonomy, authenticity and mutual recognition are simultaneously invoked and betrayed in the (brand) new economy, therefore also seems timely; and it too is underway (see e.g. Arvidsson, 2009; Arvidsson & Colleoni, 2012; Fleming, 2009; Hearn, 2008, 2010; Lair *et al.*, 2005; Willmott, 2010; Zwick *et al.*, 2008).

Notes

1 Participants in these circles tend to disapprove of the 'malicious meddlers' – whom they prefer to label as 'crackers' – that give hackers properly so called a bad name: see Raymond, n.d. The classic account of hacker values is still to be found in Levy, 1984.
2 The World Trade Organization's member states must so provide under Article 10 of the WTO's Agreement on Trade-Related Aspects of Intellectual Property Rights (1994).
3 On the materiality of all forms of public association and political engagement, see generally Marres & Lezaun, 2011.
4 In 1980, the US Congress amended the 1976 Copyright Act to protect computer programs (Pub. L. 96-517, 94 Stat. 3015, 3028, 12 December 12, 1980, amending the US Copyright Act, 17 U.S.C. §101 and §117). In 1981, the US Supreme Court definitively established computer programs' eligibility for patent protection (Diamond v. Diehr, 450 U.S. 175 (1981)).
5 In general, the requirement of originality in national copyright laws demands no more than a non-mechanical and non-trivial exercise of independent skill and judgment in the production of a work (Judge & Gervais, 2010).
6 The current version of the GPL is available at http://www.gnu.org/copyleft/gpl.html.
7 The current definition of free software, as codified by the FSF, is available at http://www.gnu.org/philosophy/free-sw.html. The FSF's account of how 'free software', 'copylefted software' and other categories of software relate to each other is available at http://www.gnu.org/philosophy/categories.html.
8 See e.g. http://sourceforge.net/ and http://freshmeat.net/.
9 The 'open source definition' can be found at http://www.opensource.org. Despite the rift, there remains a large degree of consensus between the free software and open source software camps about how software development should proceed – hence the emergence of the 'FOSS' label to designate this common ground. All agree that permissive licensing is crucial, although OS proponents are agnostic about the concept of copyleft.
10 For an exploration of Raymond's role in the FOSS movement, see Berry, 2008, Ch. 5.
11 Steven Weber commences his study of that name with the remark that '[o]pen source code does not obliterate profit, capitalism, or intellectual property rights. Companies and individuals are creating intellectual products and making money from open source software code, while inventing new business models and notions about property along the way . . .' (Weber, 2004, p. 3). Torvalds has insisted that far from being a profit-killing 'socialist' experiment, 'open source is what makes capitalism in software possible [by undoing "feudal" IP-based monopolies and barriers to entry]' (Torvalds, quoted in Tapscott & Williams, 2008, p. 92). Similar views are reported by the 'geeks' (FOSS developers and activists) that were the focus of Kelty's research (Kelty, 2008, esp. Ch. 2). See also Young, 1999.

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

12 Institutional economists have charged Garret Hardin's famous fable about the tragedy of the commons (Hardin, 1968) with ignoring the inefficiencies that can arise from private property, and the efficiency gains linked with the institutions that communities develop to manage common pool resources (see e.g. Ostrom, 1990). For interpretations of FOSS communities as engaged in the management of common pool resources, see O'Mahony, 2003; van Wendel de Joode, 2005 and Schweik & English, 2012.

13 For varying explanations, see e.g. Coleman, 2013, Ch. 4; O'Mahony & Ferraro, 2007; Tuomi, 2002.

14 Although much of Castells's terminology carries traces of his earlier structuralist Marxist leanings, it also echoes theorists of post-industrialism such as Daniel Bell, who effectively separated out what Marx had wedded together in the concept of the mode of production – i.e. 'forces' and 'relations' of production (Harvey, 2006, p. 25, Ch. 4; cf. Cohen, 1978, Ch. II) – and redefined the former as *purely* technological forces that could in themselves be regarded as the main levers of social change (Bell, 1973). Castells has accordingly insisted on an analytical distinction between capitalism as a mode of economic production and informationalism as a mode of economic development. A mode of production is organized on the basis of a structural principle determining how any unconsumed surplus arising from human action on nature is controlled (Castells, 2000, pp. 15–16). The structural principle of the capitalist mode of production is the maximization of profit, which involves a perpetual growth in the proportion of the social surplus that is controlled by the capitalist class on the basis of private ownership of the means of production. A mode of development is constituted of the technological arrangements by which a surplus is generated: these can be more or less productive and so can vary in the level and quality of surplus that they yield (Castells, 2000, p. 16). Each mode of development, Castells has claimed, is centrally defined by the elements that are fundamental in fostering productivity. These are labour and natural resources in the agrarian mode of development; energy and machines in the industrial mode of development; and technologies of information processing, knowledge generation and symbol communication in the informational mode of development (Castells, 2000, p. 17). A 'technological paradigm' is a conceptual model of how particular technologies relate to each other so that their performance may be optimized: it guides the synergetic integration of otherwise disparate technologies and directs their future trajectories (Castells, 2001, pp. 160–1). 'Informationalism' is the technological paradigm that underpins the informational mode of capitalist development.

15 For his first – and somewhat cryptic – effort to specify the culture, or ethical foundation, of informational capitalism see Castells, 2000, pp. 210–15.

16 On Castells's technological determinism, see Van Dijk, 1999. It should be said that Castells himself has consistently resisted this characterization: see e.g. Castells, 1997.

17 Permissive licensing strategies thereby seek to enlarge the information 'commons'. As Yochai Benkler has explained, 'the core characteristic of property as the institutional foundation of markets is that the allocation of power to decide how a resource will be used is systematically and drastically asymmetric. That asymmetry permits the existence of "an owner" who can decide what to do, and with whom. . . . The salient characteristic of commons, as opposed to property, is that no single person has exclusive control over the use and disposition of any particular resource in the commons' (Benkler, 2006, pp. 60–1).

18 The social norm that prohibits the removal of a contributor's name from a project's credits list without his or her consent is reinforced by rules requiring contributors and project maintainers, respectively, to log every change made to a program and record the identities of those who wrote and/or revised each portion of a program. It is also underpinned by FOSS licences. All such licences provide that if the licensee distributes the licensed software, s/he must include with each copy a copyright notice: one effect of this is to ensure that credit accrues to the author(s) of the licensed software. Most

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

FOSS licences also provide that if a licensee modifies the licensed software, s/he must ensure that the changes cannot be misattributed to the author of the original.

19 O'Mahony & Ferraro, 2007; Tapscott & Williams, 2008, p. 315 ('[w]inning companies today have open and porous boundaries and compete by reaching outside their walls to harness external knowledge, resources, and capabilities. They're like a hub for innovation and a magnet for uniquely qualified minds. They focus their internal staff on value integration and orchestration, and treat the world as their R&D department'). Accordingly, firms like Eli Lilly and Proctor & Gamble are now partnering with web-based enterprises such as InnoCentive, who claim to enable their clients to innovate faster and more cheaply by connecting them to 'networks of millions of problem solvers': see http://www.innocentive.com/.

20 See also on this point, Benkler, 2006, pp. 122–7; Fosfuri *et al.*, 2008; Henkel, 2006; Lerner & Schankerman, 2010.

21 For a prescient early analysis of how, already in the 1990s, a regime of cultural rights organized by reference to copyright and original authorship was being displaced by one underpinned by trademarks and branding, see Lury, 1993.

22 Even GPL-based licences impose no positive obligation on licensees to distribute any derivative products they make using the licensed software; nor do they require such derivatives actively to contribute to the advance of any FOSS project in which the author of the licensed software is already involved.

23 Chopra and Dexter note, for example, that the GPL does not prevent the inclusion of privacy-violating code in software acquired under a GPL-based licence (Chopra & Dexter, 2008, p. 45).

24 See e.g. GPL Cl. 10 (providing that '[e]ach time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License').

25 For a history of proprietary software licences and an account of the factors that led to the inclusion of servitudes within the system of software distribution under licences, see Hemnes, 2004. Recent US and EU jurisprudence validates recourse to licensing strategies oriented towards restricting the licensee's ability to transfer and use licensed software: see e.g. Vernor v. Autodesk, 621 F. 3d 1102 (9th Cir. September 10, 2010), cert. denied 132 S. Ct 105 (2011); cf. UsedSoft GmbH v. Oracle Int'l Corp, (Case C-128/11) (Court of Justice of the European Union, 3 July 2012) (disallowing transfer restrictions where the licensee has paid a royalty commensurate with the economic value of the licensed copy of the program, but acknowledging that the licence continues to determine the use to which the transferee may put the software after transfer).

26 For an important analysis of the affinities between proprietary and open licensing, see Elkin-Koren, 2005.

27 Jacobsen v. Katzer, 535 F.3d 1373 (Fed. Cir. 2008). Here a senior US Court ruled that the re-use of code from a program licensed under an open source licence was a breach of copyright because the licence conditions – notably those relating to attribution and the logging of modifications – had not been complied with. Although US copyright law does not in general give authors attribution rights, the licence effectively did, but only because its conditions were adjudged to be calculated to deliver economic benefits to the licensor by drawing attention to his project (ibid., pp. 1379, 1382). See further Zhu (forthcoming).

28 Chopra and Dexter note that firms' strategic use of open source licensing can be a means of undercutting any bargaining power that might otherwise accrue to employed developers from their capacity to depart the firm, taking their specialized knowledge with them, before their employers can fully extract all the value generated by their work (Chopra & Dexter, 2008, p. 22).

29 The remix contests now routinely organized on behalf of record companies are exemplary: see e.g. http://findremix.com/about/.

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

30 On the bases of hierarchy in open source projects, see e.g. Garcia & Steinmueller, 2003.
31 The well-documented tendency of young, highly educated males to dominate FOSS project credits lists provides a clue as to whom has been consigned to the margins of FOSS communities; the fact that these communities' much-lauded worldwide reach extends only to members of the cosmopolitan elite in Southern countries indicates who is effectively disconnected from them.

References

Anderson, C. (2009). *Free: The future of a radical price*. London: Random House.
Arvidsson, A. (2009). The ethical economy: towards a post-capitalist theory of value. *Capital & Class*, *33*(1), 13–29.
Arvidsson, A. & Colleoni, E. (2012). Value in informational capitalism and on the internet. *The Information Society*, *28*(3), 135–50.
Baldwin, C.Y. & Clark, K.B. (2006). The architecture of participation: Does code architecture mitigate free riding in the open source development model? *Management Science*, *52*(7), 1116–27.
Barron, A. (2012). Kant, copyright and communicative freedom. *Law and Philosophy*, *31*(1), 1–48.
Bell, D. (1973). *The coming of post-industrial society*. New York: Basic Books.
Benkler, Y. (2006). *The wealth of networks: How social production transforms markets and freedom*. New Haven: Yale University Press.
Bennis, W. & Biederman, P.W. (1998). *Organizing genius: The secrets of creative collaboration*. New York: Basic Books.
Berdou, E. (2011). *Organization in open source communities: At the crossroads of the gift and market economies*. London: Routledge.
Berry, D.M. (2008). *Copy, rip, burn: The politics of copyleft and open source*. London: Pluto.
Boltanski, L. & Chiapello, È. (2005 [1999]). *The new spirit of capitalism* (G. Elliott, Trans.). London: Verso.
Botsman, R. (2012, September). Welcome to the new reputation economy. *Wired*. Retrieved 19 April 2013 from http://www.wired.co.uk/magazine/archive/2012/09/features/welcome-to-the-new-reputation-economy?page=all
Burston, J., Dyer-Witheford, N. & Hearn, A. (Eds.) (2010). Digital labour: Workers, authors, citizens [Special issue]. *ephemera*, *10*(3/4), 214–539.
Castells, M. (1997). An introduction to the information age. *City*, *2*(7), 6–16.
Castells, M. (2000). *The rise of the network society* (2nd ed.). Oxford: Blackwells.
Castells, M. (2001). Epilogue: Informationalism and the network society. In P. Himanen, *The hacker ethic and the spirit of the information age* (pp. 155–178). New York: Random House.
Castells, M. (2004). The information city, the new economy and the network society. In F. Webster, (Ed.), *The information society reader* (pp. 150–164). London: Routledge.
Chopra, S. & Dexter, S. (2008). *Decoding liberation: The promise of free and open source software*. London: Routledge.
Cohen, G.A. (1978). *Karl Marx's theory of history – a defence*. Oxford: Clarendon Press.
Coleman, E.G. (2013). *Coding freedom: The ethics and aesthetics of hacking*. Princeton: Princeton University Press.
Dafermos, G. (2012). Authority in peer production: The emergence of governance in the FreeBSD project. *Journal of Peer Production*, *1*(1), Retrieved 15 October 2012 from http://peerproduction.net
Dahlander, L. & Wallin, M.W. (2006). A man on the inside: Unlocking communities as complementary assets. *Research Policy*, *35*(8), 1243–59.
Dusollier, S. (2007). Sharing access to intellectual property through private ordering. *Chicago-Kent Law Review*, *82*(3), 1391–435.

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

Dyer-Witheford, N. (2005). Cyber-Negri: General intellect and immaterial labor. In T.S. Murphy & A.K. Mustapha (Eds.), *The philosophy of Antonio Negri* (pp. 136–162). London: Pluto Press.
Elkin-Koren, N. (2005). What contracts can't do: The limits of private ordering in facilitating a creative commons. *Fordham Law Review*, *74*(2), 375–422.
Fisk, C.L. (2006). Credit where it's due: The law and norms of attribution. *Georgetown Law Journal*, *95*(1), 49–117.
Fleming, P. (2009). *Authenticity and the cultural politics of work*. Oxford: Oxford University Press.
Fosfuri, A., **Giarratana, M.S.** & **Luzzi, A.** (2008). The penguin has entered the building: The commercialization of open source software projects. *Organization Science*, *19*(2), 292–305.
Gambardella, A. & **Hall, B.H.** (2006). Proprietary versus public domain licensing of software and research products. *Research Policy*, *35*(6), 875–92.
Garcia, J.M. & **Steinmueller, W.E.** (2003). *The open source way of working: A new paradigm for the division of labour in software development?* SPRU Working Paper Series #92, Brighton.
Gill, R. & **Pratt, A.** (2008). In the social factory? Immaterial labour, precariousness and cultural work. *Theory, Culture & Society*, *25*(7–8), 1–30.
Giuri, P., **Rullani, F.** & **Torrisi, S.** (2008). Explaining leadership in virtual teams: The case of open source software. *Information Economics and Policy*, *20*(4), 305–15.
Grossman, L. (2006, December). You – yes, you – are *Time's* person of the year. *Time*. Retrieved 19 April 2013 from www.time.com/time/magazine/article/0,9171,1570810,00.html
Hardin, G. (1968). The tragedy of the commons. *Science*, *162*(3859), 1243–8.
Hardt, M. & **Negri, A.** (2001). *Empire*. Cambridge, MA: Harvard University Press.
Hardt, M. & **Negri, A.** (2006). *Multitude*. London: Penguin.
Hardt, M. & **Negri, A.** (2009). *Commonwealth*. Cambridge, MA: Harvard University Press.
Harvey, D. (2006). *The limits to capital*. London: Verso.
Hass, D.A. (2007). A gentlemen's agreement: Assessing the GNU general public license and its adaptation to Linux. *Chicago-Kent Journal of Intellectual Property*, *6*(2), 213–79.
Hearn, A. (2008). 'Meat, mask, burden': Probing the contours of the branded 'self'. *Journal of Consumer Culture*, *8*(2), 197–217.
Hearn, A. (2010). Structuring feeling: Web 2.0, online ranking and rating, and the digital 'reputation' economy. *ephemera*, *10*(3/4), 421–38.
Heller, M.A. (1998). The tragedy of the anticommons: Property in the transition from Marx to markets. *Harvard Law Review*, *111*(3), 621–88.
Heller, M.A. (2008). *The gridlock economy: How too much ownership wrecks markets, stops innovation, and costs lives*. New York: Basic Books.
Hemnes, T.M.S. (2004). Restraints on alienation, equitable servitudes, and the feudal nature of computer software licensing. *Denver University Law Review*, *71*(3), 577–606.
Henkel, J. (2006). Selective revealing in open innovation processes: The case of embedded Linux. *Research Policy*, *35*(7), 953–69.
Himanen, P. (2001). *The hacker ethic and the spirit of the information age*. New York: Random House.
Howe, J. (2008). *Crowdsourcing: Why the power of the crowd is driving the future of business*. New York: Random House.
Judge, E. & **Gervais, D.** (2010). Of silos and constellations: Comparing notions of originality in copyright law. *Cardozo Arts and Entertainment Law Review*, *27*(2), 375–408.
Kelty, C.M. (2008). *Two bits: The cultural significance of open source*. Durham, NC: Duke University Press.
Lair, D.J., **Sullivan, K.** & **Cheney, G.** (2005). Marketization and the recasting of the professional self: The rhetoric and ethics of personal branding. *Management Communication Quarterly*, *18*(3), 307–43.
Leadbeater, C. (2009). *We-think: Mass innovation, not mass production*. London: Profile Books.

Lerner, J. & **Schankerman, M.** (2010). *The comingled code: Open source and economic development*. Cambridge, MA: MIT Press.
Levy, S. (1984). *Hackers: Heroes of the computer revolution*. New York: Doubleday.
Loren, L.P. (2007). Building a reliable semicommons of creative works. *George Mason Law Review*, *14*(2), 271–328.
Lury, C. (1993). *Cultural rights: Technology, legality and personality*. London: Routledge.
Madison, M.J. (2004). Reconstructing the software license. *Loyola University Chicago Law Journal*, *35*(275), 275–340.
Marrella, F. & **Yoo, C.S.** (2007). Is open source software the new lex mercatoria? *Virginia Journal of International Law*, *47*(4), 807–37.
Marres, N. & **Lezaun, J.** (2011). Materials and devices of the public: An introduction. *Economy and Society*, *40*(4), 489–509.
McGowan, D. (2001). Legal implications of open-source software. *University of Illinois Law Review*, 241–304.
Moody, G. (2001). *Rebel code: Linux and the open source revolution*. London: Penguin.
Nadan, C.H. (2002). Open source licensing: Virus or virtue? *Texas Intellectual Property Law Journal*, *10*(3), 349.
Narodick, B.I. (2010). Smothered by judicial love: How *Jacobsen v. Katzer* could bring open source software development to a standstill. *Boston University Journal of Science and Technology Law*, *16*(2), 264–93.
Nissenbaum, H. (2004). Hackers and the contested ontology of cyberspace. *New Media & Society*, *6*(2), 195–217.
O'Mahony, S. (2003). Guarding the commons: How community managed software projects protect their work. *Research Policy*, *32*(7), 1179–98.
O'Mahony, S. (2005). Non-profit foundations and their role in community-firm software collaboration. In J. Feller, B. Fitzgerald, S.A. Hissam & K.R. Lakhani (Eds.), *Perspectives on open source and free software* (pp. 393–414). Cambridge, MA: MIT Press.
O'Mahony, S. & **Ferraro, F.** (2007). The emergence of governance in an open source community. *Academy of Management Journal*, *50*(5), 1079–106.
Open Source Initiative. (n.d). History of the OSI. Retrieved 18 May 2012 from http://www.opensource.org/history
Ostrom, E. (1990). *Governing the commons: The evolution of institutions for collective action*. Cambridge: Cambridge University Press.
Peters, T. (1997). The brand called you. *FastCompany*. Retrieved 19 April 2013 from http://www.fastcompany.com/28905/brand-called-you
Radin, M.J. (2000). Humans, computers, and binding commitment. *Indiana Law Journal*, *75*(4), 1125–62.
Raymond, E.S. (n.d.). Hacker. Retrieved 18 May 2012 from http://www.catb.org/jargon/html/H/hacker.html
Raymond, E.S. (2001a). How to become a hacker. Retrieved 18 May 2012 from http://www.pavietnam.net
Raymond, E.S. (2001b). *The cathedral and the bazaar*. Sebastopol, CA: O'Reilly Media.
Roberts, J.A., **Hann, I.H.** & **Slaughter, S.A.** (2006). Understanding the motivations, participation and performance of open source software developers: A longitudinal study of the apache projects. *Management Science*, *52*(7), 984–99.
Scholz, T. (Ed.) (2013). *Digital labour*. New York: Routledge.
Schweik, C. & **English, R.** (2012). *Internet success: A study of open-source software commons*. Cambridge, MA: MIT Press.
Sennett, R. (2008). *The craftsman*. London: Yale University Press.
Shah, S.K. (2006). Motivation, governance and the viability of hybrid forms in open source software development. *Management Science*, *52*(7), 1000–14.
Shapiro, K. (2004). The myth of the multitude. In P.A. Passavant & J. Dean (Eds.), *Empire's new clothes: Reading*

Downloaded by [Lancaster University Library] at 05:06 29 November 2013

Hardt and Negri (pp. 291–317). London: Routledge.
Shirky, C. (2008). *Here comes everybody: How change happens when people come together*. London: Penguin.
Söderberg, J. (2008). *Hacking capitalism: The free and open source software movement*. London: Routledge.
Stallman, R.M. (2010). *Free software, free society*. (2nd ed.). Boston, MA: Free Software Foundation.
Surowiecki, J. (2005). *The wisdom of crowds*. New York: Anchor Books.
Tapscott, D. & **Williams, A.D.** (2008). *Wikinomics: How mass collaboration changes everything* (2nd ed.). London: Penguin.
Terranova, T. (2000). Free labor: Producing culture for the digital economy. *Social Text*, *18*(2 63), 33–58.
Terranova, T. (2004). *Network culture: Politics for the information age*. London: Pluto Press.
Teubner, G. (2004). Societal constitutionalism: Alternatives to state-centred constitutional theory? In C. Joerges, I.J. Sand & G. Teubner (Eds.), *Transnational governance and constitutionalism* (pp. 3–28). Oxford: Hart.
Tirole, J. & **Lerner, J.** (2002). Some simple economics of open source. *Journal of Industrial Economics*, *50*(2), 197–234.
Tuomi, I. (2002). *Networks of innovation: Change and meaning in the age of the internet*. New York: Oxford University Press.
Van Dijk, J.A.G.M. (1999). The one-dimensional network society of Manuel Castells. *New Media and Society*, *1*(1), 127–38.
Van Houweling, M.S. (2008). The new servitudes. *Georgetown Law Journal*, *96*(3), 885–950.
Van Wendel de Joode, R. (2005). *Understanding open source communities: An organizational perspective*. Enschede: Febodruk BV.
Varghese, S. (2005, 19 August). Linus backs trademark charge. *Sydney Morning Herald*. Retrieved 18 May 2012 from http://www.smh.com.au/news/breaking/linus-backs-trademark-charge-says-mad-dog/2005/08/19/1123958221375.html#
von Hippel, E. & **von Krogh, G.** (2003). Open source software and the 'private-collective' innovation model: Issues for organization science. *Organization Science*, *14*(2), 209–23.
Weber, M. (2001 [1905]). *The protestant ethic and the spirit of capitalism* (S. Kalberg, Trans.). London: Fitzroy Dearborn.
Weber, S. (2004). *The success of open source*. Cambridge, MA: Harvard University Press.
Williams, S. (2002). *Free as in freedom: Richard Stallman's crusade for free software*. Sebastopol, CA: O'Reilly Media.
Willmott, H. (2010). Creating value beyond the point of production: Branding, financialization and market capitalization. *Organization*, *17*(5), 517–42.
Young, R. (1999). Giving it away: How red hat software stumbled across a new economic model and helped improve an industry. In C. DiBona, S. Ockman & M. Stone (Eds.), *Open sources: Voices from the open source revolution* (pp. 56–61). London: O'Reilly & Associates.
Zhu, C.W. (forthcoming). 'Copyleft' reconsidered: Why software licensing jurisprudence needs insights from relational contract theory. *Social & Legal Studies*.
Zwick, D., Bonsu, S.K. & **Darmody, A.** (2008). Putting consumers to work: 'Co-creation' and new marketing governmentality. *Journal of Consumer Culture*, *8*(2), 163–96.

Anne Barron is a Reader in Law at the London School of Economics, where she teaches and researches in the fields of intellectual property and media law, modern European philosophy and critical social theory.

Downloaded by [Lancaster University Library] at 05:06 29 November 2013