WEEK 9 WHITE PAPER: OPEN SOURCE

For the past few weeks we've been touching at various points—not always happily!—on hacker culture. This week the roles of hackers, hacker culture, and the 'hacker ethic' are central to our topic, so it pays to consider these terms briefly. The Steven Levy book mentioned in the Week 7 readings is the classic text on hacking's early years from the 1950s through the 1980s. The first two chapters are available through Project Gutenberg, the second of which discusses Levy's conception of 'the Hacker Ethic':

- Access to computers—and anything which might teach you something about the way the world works—should be unlimited and total.
- Always yield to the hands-on imperative!
- All information should be free.
- Mistrust authority—promote decentralization.
- Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race, or position.
- You can create art and beauty on a computer.
- Computers can change your life for the better.

Levy's stories of early hackers made it clear that the original spirit of hacking was one of *exploration* driven by interest and a sense of fun:

> According to the standard thinking on computers, their time was [so] precious that one should only attempt things which took maximum advantage of the computer, things that otherwise would take roomfuls of mathematicians days of mindless calculating. Hackers felt otherwise: anything that seemed interesting or fun was fodder for computing.

For many programmers, this still *is* the spirit of hacking. To identify oneself as a hacker is to identify oneself as an explorer of technological potential for the sheer joy of it. Richard Stallman's 2002 piece On Hacking expresses this same view:

> It is hard to write a simple definition of something as varied as hacking, but I think what these activities have in common is playfulness, cleverness, and exploration. Thus, hacking means exploring the limits of what is possible, in a spirit of playful cleverness. Activities that display playful cleverness have "hack value".

But as Stallman and others complain, somewhere along the way the word became corrupted in the popular imagination to mean simply 'security breaker'. Although it's important to distinguish between the different connotations of the term, it's useful to think about why the confusion has persisted for so long despite objections. Many hackers are complicit in the confusion, encouraged as it is by statements like Stallman's:

> Hackers typically had little respect for the silly rules that administrators like to impose, so they looked for ways around. For instance, when computers at MIT started to have "security" (that is,

> restrictions on what users could do), some hackers found clever ways to bypass the security, partly so they could use the computers freely, and partly just for the sake of cleverness (hacking does not need to be useful).

All sorts of rules seem 'silly' if you look at them in a certain way; any rule will inhibit to some degree the kind of open and free exploration that informs the hacker spirit. Once again we come back to this ideological divide between liberty and control, with hacking very much being about the former.

Rulebreakers and outlaws—pirates, cowboys, bushrangers, highwaymen—bring out the romantic in many, and programmers can be just as susceptible to that romantic allure, with the security-breaking side of hacking falling into a greyer area in the hacker imagination than distinctions between 'black hat' and 'white hat' hackers or 'hackers' and 'crackers' would have us believe. This must partly be because of the age at which hackers are first formed and often at their most active: their teens and early twenties, before the responsibilities of adult life take hold. Old-school hackers look down their noses at script kiddies, 'wannabes' (often very young) who use existing programs to break into networked computers rather than write their own, but the existence of such wannabe behaviour is testament, surely, to the romantic draw of the hacker-as-outlaw.

Other attempts to explain hacking have also pointed to the time of life in which hackers usually become hackers. An early attempt to defend them from negative media portrayals (Brian Harvey [1985], What is a Hacker?) linked the 'aesthetics' of hacking with youth:

> Steven Levy ... talks at length about what he calls the "hacker ethic." This phrase is very misleading. What he has discovered is the Hacker Aesthetic, the standards for art criticism of hacks. For example, when Richard Stallman says that information should be given out freely, his opinion is not based on a notion of property as theft, which (right or wrong) would be an ethical position. His argument is that keeping information secret is inefficient; it leads to unaesthetic duplication of effort.
>
> The original hackers at MIT-AI were mostly undergraduates, in their late teens or early twenties. The aesthetic viewpoint is quite appropriate to people of that age. An epic tale of passionate love between 20-year-olds can be very moving. A tale of passionate love between 40-year-olds is more likely to be comic. To embrace the aesthetic life is *not* to embrace evil; hackers need not be enemies of society. They are young and immature, and should be protected for their own sake as well as ours.

All of this is intimately tied up with education, because hackers, at least in the past, usually became hackers at educational institutions: at school or at university, where they had access to equipment and networks unavailable to them at home. Hacking shifted from being something done on computers to something done on computer networks as educational institutions became more networked (although the earlier phenomenon of phreaking played a part); well into the 1990s, it was easier to explore a network on fast university machines than over slow home connections, and much of that exploration was of other networked educational sites. Once hacking became largely a case of hacking *networks* over hacking

hardware or hacking code, it was almost inevitable that the 'security-breaking' image of hacking would become dominant.

But those other forms of hacking persist, even if their exponents don't always see themselves as such. The hardware-oriented hacker might rebuild his or her PC on a regular basis whether it needs it or not, using parts from obscure sources; or may be into overclocking chips to increase performance; or may be a case modder turning his or her computer into an aquarium or a microwave. The software-oriented hacker may write freeware or shareware applications for commercial operating systems; or may be part of the open source movement, the most famous example being Linux (or GNU/Linux—once again, the naming is controversial).

So let's turn to the open source movement and where it's come from. Its historical roots go back to the early years of computing, when we consider that for most of that time most software was not a commercial proposition; it was just what made commercial hardware work. By the late 1970s, however, programmers writing for early personal computers were starting to use early forms of copy protection to maintain the commercial value of their work, and it was this that prompted Richard Stallman to found GNU, a volunteer project to produce free versions of all major Unix software from the kernel (the heart of the operating system) up. Stallman's initial announcement from 1983 reads like a call-to-arms for what we would come to know as open source, although not all the details were in place.

The open source philosophy gained its slogan a year or so later, when Stewart Brand (founder of the Whole Earth Catalog and co-founder of the influential San Francisco online community the WELL) coined the phrase 'information wants to be free':

> Information Wants To Be Free. Information also wants to be expensive. Information wants to be free because it has become so cheap to distribute, copy, and recombine—too cheap to meter. It wants to be expensive because it can be immeasurably valuable to the recipient. That tension will not go away.
>
> The pressure of the paradox forces information to explore incessantly. Smart marketers and inventors quietly follow—and I might add, so do smart computer security people.

This shift from the principle that information *should* be free to the anthropomorphized 'information *wants* to be free' came at the moment when hacking was moving from an untheorized activity to one with histories (like Levy's) and its own philosophical debates and underpinnings. The idea that information has its own wants and imperatives, beyond those of its individual creators, fits well with the notion of hackers as explorers of technologies and their associated information landscape: hackers as Magellan or Columbus, hackers as Newton or Einstein or Darwin; hacking as the technologist's scientific method, with the same open and sharing approach to discovering and testing knowledge. Just as scientists seek to discover the laws of the universe, hackers seek to discover the laws of information.

At this same point, Levy's history introduced the notion of a 'hacker ethic', with its elements of ethics, aesthetics, and observed belief. In the decades since, the hacker community has taken this notion and

modified it, refined it, elaborated it, and lived by it.

GNU grew throughout the 1980s, but by the early 1990s was still missing the crucial element of the Unix kernel. Linux, Linus Torvalds' project to write a free Unix kernel, filled that gap. As Linux evolved, its distributions (the various versions of the OS) took on elements of existing GNU software, which is why some (such as Stallman, but not Torvalds) label the result 'GNU/Linux'. By the late 1990s, Linux had become important enough that it was in turn inspiring similar models of software development across different operating systems. This 'open source' movement took its label from the key feature that its source code was open to others to copy, modify, and redistribute, subject to licensing conditions.

Stallman's objections to such labels was as much political as proprietary, as Smajda (2011, p. 314) indicates when discussing how 'open source' won out over the label 'free software':

> Tim O' Reilly (founder of O'Reilly Media, a technology book publisher) organized a "Freeware Summit," assembling a who's who of free software project leaders at the time—with one notable exception: Richard Stallman, who O'Reilly felt would "disrupt the effort to achieve a consensus" (Moody, 2001, p. 167). The group decided they needed to agree on a name if they were going to maximize their collective chance at business success, and after putting several options to a vote, "open source" was selected. The group decided that the crucial component was the sharing of source code. "Free software" got "almost no positive votes" (van Rossum, 1998). Stallman interpreted this as treason against the GNU mission: "free software" is fundamentally about an ethical, moral belief that proprietary software is a threat to human liberty and equality. "Open source" drained this of its moral content in Stallman's view, leaving only an argument about the practical benefits of sharing source code.

Sullivan (2011, p. 232) argues that "social justice issues have been at the core of the free software movement ever since Stallman crafted the notion of the communitarian ethos that prevents many software projects from being removed from the public domain by introducing copyright restrictions". His article provides a good history of the movement and an argument for treating it as a social movement:

> From its roots in the rise of microcomputing in the 1970s and 1980s, the free software movement has identified a set of core ideological beliefs and discourse about the basic freedoms of computer software. ... Evidence of the links between the free software movement and social justice issues can be seen via the plethora of nonprofit relief and development organizations that have adopted F/OSS technologies and, along with it, the shared ethos of the free software movement. (p. 237)

In the secondary readings, Lehmann (2004) looks at FLOSS (free/libre and open source software) as a 'social formation' rather than a community or movement, defined in the first instance by its licensing arrangements (the 'legal boundaries'), and then by a range of 'cultural boundaries' which overlap to a considerable extent Levy's hacker ethic:

> The goal is to write software and not—like in software companies—make a profit. Whether a

> product is ready for use at a specified time does not play an essential role. Rather, a new version should be released whenever it is fit for use. Software must not only work somehow, but should also be written well. There is a shared understanding that code should be elegant.
>
> Generally, FLOSS developers, in their self-image, tend to emphasise technical knowledge and a certain can-do-spirit to differentiate themselves from others.
>
> In addition to shared norms and values, the collective identity is underpinned by shared traditions. Common roots and role models are found in the early days of computing. The 1960s hackers at MIT with their spirit of exploration, or the early Unix users who developed a form of close and free cooperation are among them.

Lehmann also discusses motivations 'for joining, and the motivation to stay on' which again overlap hacker motivations:

> The motivation for joining is twofold: motivation for taking part in FLOSS development at all, and motivation to join a particular project. The reasons of individuals vary widely: to fight for the spread and growth of FLOSS; to give something back for the software you use (reciprocity); solving a specific problem; having fun (the FLOSS playground); improving your own programming skills; and, a service to others.

But she rejects the idea of FLOSS as a social or political movement:

> FLOSS projects are about creating software, not about social change. Political activities are mainly the task of other organisations, which may have close links to FLOSS projects and even the same members, but which are set apart from development projects precisely by their political activist character. Therefore FLOSS as a whole is not a social movement. To (nearly all) FLOSS developers, writing software is the focal point, not the political aspects. The structure of projects is shaped by this, not by the achievement of some social change. There is however, a set of organisations (such as the FSF or the OSI) which are more movement-like in character. But they belong to a social movement concerned with issues like free speech, free access, and data protection—not with writing good code.

Smajda (2011, pp. 306-7) writes about the "moral field" of computing, the two axes of which are "competing conceptions of identity and community (a community of experts vs. a community extending to all of humanity) on one axis, and competing moral and ethical visions (an ethic of community vs. an ethic of the market) on the other":

> There is more to open source than a vague commitment to "transparency, participation, and collaboration." Computing culture has a tradition of free, open, and contentious deliberation about the rights and responsibilities shared by members of a community, and the political culture of computing is more complicated than a simplistic distinction between open source and proprietary software suggests. It is a culture built upon a moral field in which actors situate themselves and others according to the positions they take on group boundaries, moral

> obligations, and the role of markets. "Open source" is one way of talking within this field of positions. (pp. 320-21)

Eric Raymond is a central figure in the open source movement, and his writings have been much read and discussed by hackers and those studying them. Rather than his first major paper 'The Cathedral and the Bazaar', I've included in the further readings the later 'Homesteading the Noosphere' (1998) for its explicit connections of hacker ideologies to political philosophy (Lockean theories of property) and anthropological theories of 'gift cultures':

> The ideology of the Internet open-source culture (what hackers say they believe) is a fairly complex topic in itself. All members agree that open source (that is, software which is freely re-distributable and can readily be evolved and modified to fit changing needs) is a good thing and worthy of significant and collective effort. This agreement effectively defines membership in the culture. However, the reasons individuals and various subcultures give for this belief vary considerably.
>
> One degree of variation is zealotry; whether open source development is regarded merely as a convenient means to an end (good tools and fun toys and an interesting game to play) or as an end in itself.
>
> Another ... is in hostility to commercial software and/or the companies perceived to dominate the commercial software market.
>
> To pragmatists, the [GNU General Public Licence] is important as a tool rather than an end in itself. Its main value is not as a weapon against 'hoarding', but as a tool for encouraging software sharing and the growth of bazaar-mode development communities. The pragmatist values having good tools and toys more than he dislikes commercialism, and may use high-quality commercial software without ideological discomfort.

Raymond discusses 'an elaborate but largely unrecognized set of ownership customs' that 'regulate who can modify software, the circumstances under which it can be modified, and (especially) who has the right to redistribute modified versions back to the community':

> As these customs have evolved over time, they have done so in a consistent direction. That direction has been to encourage more public accountability, more public notice, and more care about preserving the credits and change histories of projects in ways which (among other things) establish the legitimacy of the present owners. These features suggest that the customs are not accidental, but are products of some kind of implicit agenda or generative pattern in the open-source culture that is utterly fundamental to the way it operates.

Raymond relates these customs to John Locke's 18th century theories of common-law land tenure: *homesteading* on the frontier to establish new titles, *transfer of title* from one owner to the next, and *adverse possession* of abandoned land.

> The 'noosphere' of this paper's title is the territory of ideas, the space of all possible thoughts. What we see implied in hacker ownership customs is a Lockean theory of property rights in one subset of the noosphere, the space of all programs. Hence 'homesteading the noosphere', which is what every founder of a new open-source project does.

The 'hacker as frontiersman' imagery again appeals to the romantic side of hacker culture, as well as tapping into American ideals of freedom and liberty, where government and laws get out of the way while frontiersmen create something out of nothing.

This work, Raymond says, is done to build *reputations* rather than fortunes, although economic value can result from a good reputation among hackers. This leads to his description of hacker culture as a gift culture:

> Gift cultures are adaptations not to scarcity but to abundance. They arise in populations that do not have significant material-scarcity problems with survival goods. We can observe gift cultures in action among aboriginal cultures living in ecozones with mild climates and abundant food. We can also observe them in certain strata of our own society, especially in show business and among the very wealthy. Abundance makes command relationships difficult to sustain and exchange relationships an almost pointless game. In gift cultures, social status determined not by what you control but by *what you give away*.

He goes on to examine various angles of these variables of reputation, territory and ownership, including the 'joy of hacking' theme previously mentioned, before closing the paper with a meditation on links with academia:

> Respondents to this paper too numerous to list have pointed out that hacker ownership customs seem intimately related to (and may derive directly from) the practices of the academic world, especially the scientific research community. This research community has similar problems in mining a territory of potentially productive ideas, and exhibits very similar adaptive solutions to those problems in the ways it uses peer review and reputation. Since many hackers have had formative exposure to academia (it's common to learn how to hack while in college) the extent to which academia shares adaptive patterns with the hacker culture is of more than casual interest in understanding how these customs are applied.
>
> I suspect academia and the hacker culture share adaptive patterns not because they're genetically related, but because they've both evolved the most optimal social organization for what they're trying to do, given the laws of nature and the instinctive wiring of humans. The verdict of history seems to be that free-market capitalism is the globally optimal way to cooperate for economic efficiency; perhaps, in a similar way, the reputation-game gift culture is the globally optimal way to cooperate for generating (and checking!) high-quality creative work.

It's here that we can turn to Himanen's chapter from *The Hacker Ethic and the Spirit of the Information Age* (2001), an interesting short book that reads as much as political or social theory as an account of

hackers. Himanen seeks to contrast the 'hacker ethic' with the Protestant work ethic: so, not just a collection of tropes about ethical and moral or 'aesthetic' behaviour, but an ideology that motivates hackers to do what they do, the central idea of which is not that *work* is the focus of life and an end in itself (the work ethic), but that *passion* is:

> For hackers, *passion* describes the general tenor of their activity, though its fulfillment may not be sheer joyful play in all its aspects. Thus, Linus Torvalds has described his work on Linux as a combination of enjoyable hobby and serious work: "Linux has very much been a hobby (but a serious one: the best type)." Passionate and creative, hacking also entails hard work. (Himanen 2001, pp. 18-19)

The short chapter on 'The Academy and the Monastery' that you've seen links the open model of software development to this hacker ethic through its discussions of Linux and Raymond, and then argues for the academy and the scientific method as an allegory for open source development:

> Scientists, too, have chosen this model not only for ethical reasons but also because it has proved to be the most effective way of creating scientific knowledge. All of our understanding of nature is based on this academic or scientific model. The reason why the original hackers' open-source model works so effectively seems to be—in addition to the facts that they are realizing their passions and motivated by peer recognition, as scientists are also—that to a great degree it conforms to the ideal open academic (pp. 68-69)

Himanen goes on to discuss the ways in which the 'hackers' open learning model' resembles the original model of Plato's Academy, where 'it was not the teacher's task to inculcate the students with preestablished knowledge but to help them give birth to things from their own starting points' (p. 76), and suggests that:

> We could also use this idea to create a generalized Net Academy, in which all study materials would be free for use, critique, and development by everyone. By improving existing material in new directions, the network would continuously produce better resources for the study of the subjects at hand. Members of the networks would be driven by their passions for various subjects and by the peer recognition for their contributions. (p. 77)

MOOCs and other open learning initiatives immediately spring to mind, as does Wikipedia, and this is an obvious topic to discuss in an educational context, in tension with the economic imperatives we looked at earlier in the semester. Sullivan (2001) notes that "Stallman’s emphasis on reinvigorating a sense of common good via artistic and other cultural expression has become the philosophical foundation for the larger 'free culture' movement," (p. 233) and draws connections between "open source hackers", "software engineers in general" and "teachers, artists, and others who work in the cultural industries" as "knowledge workers" (pp. 234-35).

We could turn here to the article by Deimann (2013), which focuses not on open source but on its educational partner in openness, open educational resources. Deimann sees OERs as a partner or flipside

to the German educational concept of *Bildung*:

> Since the eighteenth century, *Bildung* has been known as a German 'invention' with a variety of political, economic, philosophical, religious, and educational connotations. It can only be loosely translated into English with the terms 'formation' or 'edification', but these fail to capture its multifaceted and multilayered significance. ... In basic sense, *Bildung* refers to the free, dialogical and dialectical interplay between the individual and the world in such a way as to allow for the individual's self-realization—the full unfolding of his or her innate potentials. It was conceptualized as a fundamental right for every human being, regardless of social class. (p. 192)

> It appears that both [OER and *Bildung*] affirm education as a common good, and both seek to support the prosperity and well-being of the individual and society. Furthermore, OER and *Bildung* present special visions of the future and propose means to reach this: Whereas OER are aimed at increasing access to the world's digital knowledge, *Bildung* emphasizes both the individual and social potentials and processes of engaging with these resources. (p. 193)

> Open Educational Resources and open online courses, in increasing access to educational resources of all kinds, can be seen as renewing the value of *Bildung* as an educational construct that affirms the universality of education as a common good, realized in different ways in society and the individual. (p. 196)

> *Bildung* emphasizes that the process of inner formation or self-realization is one requiring a dialectical balance and space of play between the individual and society—something that can't simply be realized through arbitrary choices and preferences. It is thus the task of the individual to direct the process of navigating through open complex worlds and to acquire orientational knowledge that has value because of its sensitivity to others' values and understandings. (pp. 196-97)

Deimann sees a threat to this spirit in the form of the filtering of our information and communication streams by social media and search giants:

> Facebook's personalized news stream and Google's personalized search results ... can be said to offer a kind of 'anti-*Bildung*'—the gradual elimination of the space for free dialectical interplay between self and world. Instead, what occurs is a self-reinforcing encounter of the self with itself —or with that which has been determined as compatible with this self and its world view. (pp. 197)

Not everyone sees open source as a 'gift economy' driven by romantic hacker ideals. David Lancashire (2001) pointed out that Raymond's writings are very much political manifestos, and that 'Homesteading the Noosphere' draws 'the most revolutionary conclusions found in the literature'. Lancashire challenged this by surveying representative open source activity across different countries and finding that the ones actually doing the work were not those that 'gift culture' arguments would suggest:

> Our findings are remarkable. The United States consistently drops to a position of relatively inactive development—a very puzzling discovery. The Northern European social democracies hold their position at the head of the list, and appear to be undertaking a completely disproportionate amount of actual work. ... In traditional economic lingo, the United States appears to be free riding on a collective good provided predominantly by non-Americans. The stark nature of this contrast is a devastating blow to arguments about post-scarcity "gift cultures". If the wealthiest country in the world is one of the least active (relative) contributors to open source development, it seems ludicrous to explain said development as a function of post-materialism.

So, does open source ideology reflect the real world, including the real economic reasons for open source development, or is it a mix of idealism and other political ideologies that masks what's actually going on and could lead governments, corporations and educational institutions down blind alleys? In relation to the last of these we should consider not just Himanen's 'Net Academy' , but the competition between commercial VLEs and open source alternatives. Which are sustainable, and which should institutions use?

Next week we'll be looking at the missing piece of the open source story: copyright and intellectual property (or 'property'). You will have seen mentions in some of this week's readings of a key player in this debate, Lawrence Lessig, who I'll leave with the last word for now:

> The argument that government has not played an important role in bringing about the environment within which the revolution of the Internet was possible is just wrong—historically wrong. The argument that we don't need the government to play the same role in the future of the Internet is also just wrong—pragmatically wrong. The movement has got to wake up to this reality, or this reality will crush the movement.... While the movement turns its energy away from government, while it cocoons itself with this lullaby about how little from government we need, others less principled and others less convinced dominate the debate in Washington. While the open-source crowd has scorned those who would speak of regulation, regulations abound. Bad regulation, no doubt, but bad regulation is the product of a process where the good was not heard. (Lessig 2000)

References

Deimann, M. (2013) 'Open Education and *Bildung* as Kindred Spirits', *E-Learning and Digital Media*, Vol. 10, No. 2, pp. 190-99.

Himanen, P. (2001) *The Hacker Ethic and the Spirit of the Information Age*, London: Secker & Warburg, chapter 4, pp. 63-81.

Lancashire, D. (2001) The Fading Altruism of Open Source Development, *First Monday*, Vol. 6, No. 12 (December 2001).

Lehmann, F. (2004) FLOSS developers as a social formation, *First Monday*, Vol. 9, No. 11 (November

2004).

Lessig, L. (2000) Should Public Policy Support Open-Source Software? Round Two, *Amercian Prospect*.

Raymond, E. S. (1998) Homesteading the Noosphere, *First Monday*, Vol. 3, No. 10 (October 1998).

Smajda, J. (2011) 'Open Source and the Moral Field of Computing', *Journal of Information Technology & Politics*, Vol. 8, pp. 304-22.

Sullivan, J.S. (2011) 'Free, Open Source Software Advocacy as a Social Justice Movement: The Expansion of F/OSS Movement Discourse in the 21st Century', *Journal of Information Technology & Politics*, Vol. 8, pp. 223-39.