REPUTATION IN OPEN SOURCE SOFTWARE

Andrew Watson

a.watson@neu.edu

College of Business Administration

Northeastern University

Working paper

This is version 1.0, sent to the Free/Open Source Research Community in September 2005

ABSTRACT

The 1990s and early 2000s have seen the dramatic rise of open source software, with the Linux operating system as the most salient example. This article focuses on the role of reputation in open source. It describes the importance of the reputations of hackers, software vendors, open source projects, and the open source movement. Although reputation has long been used as an explanation of hacker motivation, this article applies the concept of reputation at multiple levels, and identifies the inter-level relations. The article also emphasizes the particular importance of reputation in inter-firm competition within the open source arena, drawing on strategic management's resource-based view of the firm to do so.

INTRODUCTION

One of the most notable changes in information technology (IT) in recent years has been the rise of open source software. By 2002, almost one in every three server computers was running the open source operating system Linux. IBM spent a billion dollars on Linux-related research and development in 2001, despite the fact that Linux is not owned by IBM, or by any other IT vendor (Ante, 2001). Firms such as Red Hat are in a sense making an even larger bet on open source; these are the “pure play” open source firms, which concentrate solely on open source software and closely related products.

It may seem strange that these and other for-profit firms are investing heavily in Linux. After all, Linux is freely available, and so contributions to the Linux code made by IBM are available, at no cost, to IBM’s customers and competitors. Things become stranger when we compare the reputation of IBM with that of Linux. IBM’s reputation is that of solid supplier of information technology to the corporate world. In contrast, the Linux community was described on the cover of *Business Week* as “a ragtag band of software geeks” (Kerstetter et al., 2003; similar descriptions abound). We will see that despite—or because of—this contrast, reputation is a concept useful in explaining IBM’s investment decision.

Indeed, the concept of reputation is useful in explaining much of the story of open source software. It is the basis for Raymond’s (2001) answer to the question of why individuals contribute to the development of open source software. The question is similar at the individual and organizational levels of analysis: why contribute to open source, when it is by definition a public good?

This article uses the concept of reputation extensively, showing how it is relevant to the actions of many characters in the story, and not only to the actions of the “hackers” who write the software. In that it focuses on reputations at multiple levels (individual, firm, etc.) and on the interactions between these levels, it is distinctive within the research on open source. It contributes, not only to research on open source, but also to research on strategic management. Much recent research in strategic management discusses competitive advantage in terms of resources. Real advantage, according to the “resource-based view of the firm,” results from resources that are difficult to imitate. The argument is that if advantage-conferring resources are imitable, then they will be imitated by competitors, and the advantage will last only briefly before being competed away (Barney, 1991).

In this context, open source software is a particularly interesting phenomenon, since its defining resource—the software source code—is by definition open and hence perfectly imitable. The source code is what computer programmers write, and from which runnable programs are derived. We can similarly say that the cost of imitating “free software” is zero, or close to it. Due to the imitability of its key input, open source represents a field in which less readily imitated resources, such as reputation, become proportionately more important. Hence open source is the kind of extreme case (Eisenhardt, 1989) that may shed light on a phenomenon of interest—in this case, reputation.

Before moving to the body of this case study of open source and reputation, it is worth remarking that not every relevant reputation is a positive one. The open source community is one of the quarters in which Microsoft’s reputation is negative. SCO is “the most hated company in tech" (i.e. not just in the open source community) because of the legal actions it has taken against open source (Kerstetter, 2004).

The main section of this article describes the reputations of each of four entities: hackers, projects (e.g., Linux), vendors (e.g., IBM), and the open source movement itself. It is followed by a discussion of implications for futher research into open source software and reputation. But before describing the reputations of the various entities, it is helpful to introduce the terms, concepts, and sources on which this case study is built, and then to provide a brief history of open source software.

TERMS, CONCEPTS, AND SOURCES

We start with the term *reputation*:

> As normally understood, the essential features of reputation, whether of a person, a group, an organization, an object, an event or an activity, are that there is some sort of estimation of its nature and value, and that this estimation is widely shared in a group of people. (Bromley, 1993, p. 12).

It is important to note that reputational entities—that is, bearers of reputation—include not only people, but also organizations, and other entities. We will consider multiple reputational entities, not all of them people. We will consider multiple reputations for each of these entities, since reputations differ between groups that interact with the entity. Davies et al. (2003, p. 58) emphasized the importance of multiple reputations for corporations when they stated, as the first tenet of their "reputation paradigm," that "multiple stakeholders [shareholders, customers, etc.] need to be considered."

For the second key term, we turn to *The Open Source Definition* (Open Source Initiative). In order to qualify as open source, software must fufil multiple criteria, the first three of which are particularly relevant here. Each of these three criteria corresponds to a right conferred upon programmers, and indeed upon any user of open source software. The first is the right to redistribute the software. The second is the right of access to the source code; without such access, it is not possible to improve or otherwise change the software. The third right is indeed the right to improve the software, that is, to produce derivative works.

A few terms closely related to open source software also merit early clarification. This is particularly true of the term *free software*. Richard Stallman, founder of the Free Software Foundation (FSF), has always emphasized that the “free” is used as in “free speech,” rather than as in “free beer.” The FSF considers the above-described rights (redistribution, source code, and derived works) to have an ethical foundation, to the extent that “non-free software is a social problem and free software is the solution” (Stallman, 2002, p. 55). Since the word *free* is English and ambigious, the term *software libre*, with its unambiguous connotation of liberty, is used in many parts of the world.

The term *hacker* as used here refers to “someone who loves to program and enjoys being clever about it.” This particular definition is due to Stallman (2002, p. 15). Whether a hacker identifies with the free software movement or with the open source movement (or with neither) it is likely that he would agree with this definition. One of the many things on which the movements are in agreement is that *hacker* should be used in this sense, and not to refer to a computer criminal. The masculine pronoun is used to refer to hackers, since well over 90% are male (Ghosh et al., 2002).

As may be apparent by now, the writings of prominent hackers, notably Raymond and Stallman, provide some of the material for this case study of open source software. Open source is to some extent self-documenting via the web, where the pages of individuals, organizations, and discussion groups abound. Journalists' accounts of the phenomenon provide further material; Moody's (2002) book is particularly useful. Academic research on the web is burgeoning. This article draws on such research where appropriate, but does not include a review beyond the needs of the case study. The article similarly draws on, but does not review, the literature on reputation.

A BRIEF HISTORY OF OPEN SOURCE SOFTWARE

Lerner and Tirole (2002) identified two eras of open source software development. They also identified a prior era of "cooperative software development," but we can commence our history at the start of the first open source era, in the early-to-mid-1980s. It was then that Stallman, working at MIT's Artificial Intelligence lab, embarked upon an ambitious free software project, developed (with others) a license in order to ensure that the software remained free, and founded the FSF.

The name of the project is GNU, which stands for GNU's Not Unix. (The recursive acronym is a familiar device of hacker humor.) GNU is an operating system modeled on Unix. Stallman chose an operating system rather than any other type of software because the operating system is the fundamental software on which applications (such as word processors, spreadsheets, tax preparation packages, etc.) run; hence a free operating system could almost literally form the foundation for the free software movement. He chose the Unix operating system in order that the large base of Unix users could easily switch to the new, free operating system.

It was not sufficient for Stallman that GNU be free. It was also imperative that it remained free, and that works derived from it would also be free. This is where the concept of *copyleft* and the terms of the GNU General Public License (GPL) come in. Copyleft is a pun on copyright; it is a mechanism to safeguard the continued freedom of GNU, and software like it. The GPL grants the user the freedom to run GNU, to redistribute it, to modify it, and to redistribute the modifications—provided the modifications are also redistributed under the terms of the GPL. Hence the GPL forbids taking free software and modifying it to produce non-free software. While GNU comprises software code, the GPL uses the code of copyright law in order to keep GNU and other software free.

The FSF provides the organizational home for GNU and other free software. It is a tax-exempt charity. These three components of the free software movement—GNU, the GPL, and the FSF—to a large extent defined the first era of open source development, and remain crucial in the second era, which runs from the early 1990s to the present.

This current era is characterized by growth in open source software along at least three related dimensions: the number of open source projects; the number of hackers contributing to these projects; and interactions between the domain of open source and the domain of for-profit IT (Lerner and Tirole, 2002). Linux was one of these new projects; its growth is the most prominent aspect of the rise of open source software. In 1991, Linus Torvalds started writing and running Linux; hence he was the only user, and the only developer. By 1994, there were half a million Linux users. By 2001, Linux had almost a third of the server market (Ante, 2001). It is important to note that this is the market for servers—the back-end computers on which heavy processing takes place. On the desktop side the progress of Linux—and of open source in general—is much slower.

Hence to provide further examples of the rise of open source, we remain on the server side. Apache is an open source web server. When we access a web page, our web browser (e.g., Internet Explorer, or the open source Mozilla Firefox) sends a request to the web server, which retrieves the page and sends it back. More often than not, the server in question is open source: more than 50% of web servers currently run Apache. This is more than double the share held by Microsoft's Internet Information Server (Lerner & Tirole, 2002).

Samba is also a server side open-source program. It emulates the file server capabilities of Windows. Hence client code written to interact with a Windows file server can, without being rewritten, interact with Samba. This allows a server running Samba and Linux (or one of the other operating systems on which Samba runs) to be plugged seamlessly into a network otherwise comprising Windows machines. This in turn allows performance benchmarking of open source file servers (Samba/Linux) against Microsoft file servers (Windows). Such benchmarking commenced in early 1999; as we will see, the ensuing spate of benchmarks boosted the reputation of open source software (Moody, 2002).

The second era saw not only new projects, but also a new organization: the Open Source Initiative (OSI). Bruce Perens (1999) has described how and why he, Eric Raymond, and others founded the OSI, and developed the above-mentioned Open Source Definition. One of their objectives was to make software such as Linux more palatable to business people, some of whom were deterred by the "libertarian fervor" of the FSF. The FSF and the OSI are the formal organizations at the core of two different movements; the former emphasizes ethical considerations, and the latter, practical considerations. Although distinct, the movements have more similarities than differences (Perens, 1999; Stallman, 2002). For example, each promotes Linux, even though the FSF insists that it is properly referred to as GNU/Linux (since the operating system in question was originally called GNU, and has many components, including the Linux "kernel").

Since this article focuses on business matters rather than on technical matters, it uses the terms favored by the explicitly business-friendly OSI, such as *Linux* and *open source*. Hence the term *open source community* is used here to include hackers who identify with the FSF, as well as those who identify with the OSI. This is consistent with the way in which by many vendors (e.g., Red Hat) and scholars (e.g., Feller and Fitzgerald, 2002) refer to the community.

Insert Figure 1 about here

To continue using business-oriented terms: we can discern the emergence during this second phase of a new supply chain. Figure 1 illustrates this new chain, and contasts it with the original (and still existing) open source supply chain. In the original chain (Figure 1.1), hackers contributed to open source projects, such as Linux, and used such contributions. As for-profit firms saw opportunities in open source, and other for-profit firms were founded to pursue such opportunities, a new chain emerged (Figure 1.2). To flesh out this new chain with examples: Linux is an open source project, Red Hat is a vendor, and Amazon.com is one of Red Hat’s customers; Amazon’s servers run the Linux operating system, with Red Hat providing not only the operating system, but also technical support and other services.

While Figure 1 shows relationships among key entities in terms of *supply*, Figure 2 illustrates relationships among these entities in terms of *reputation*. The next section describes these relationships.

Insert Figure 2 about here

REPUTATIONS IN OPEN SOURCE

There are multiple ways in which reputation will influence the flows along the value chain in Figure 1.2. For example, a hacker may be motivated to contribute to an open source project in order to advance his reputation among his fellow hackers. Reputation may be one of the factors influencing a customer's choice of open source vendor. Further, to introduce a reputational entity appearing in Figure 2 but not in Figure 1, the reputation of the open source movement may influence the customer's choice between an open source solution and a proprietary solution.

Hence we will consider four reputational entities: the hacker, the project, the vendor, and the open source movement itself. For each of these entities, we will consider two key stakeholders. The resulting account of reputation in open source software is not exhaustive. Rather, it is compact enough to be summarized in a single figure (Figure 2) while aiming to be sufficient to illustrate the importance in open source software of reputations, and to highlight the most important reputational entities and stakeholders.

Reputations of Hackers

It is natural to start any account of open source software with hackers. They write, share, use, debug, and further develop the source code. We will start, then, with the *reputation of hackers in the eyes of other hackers*. Eric Raymond (2001), in one of his participant-observer accounts of the open source movement, highlighted the importance of reputation in this sense. He identified three reputation-related motives for hackers to contribute to open source projects. He mentioned first a motive grounded in evolution, and in the assertion that human beings evolved in such a way as to seek status within the group.

The second motive to contribute is grounded in anthropology, and on the proposition that open source is a “gift culture,” in which the gift of source code is a means of attracting attention and cooperation from others in the culture (see also Kollock’s (1999) account of gifts in cyberspace). These first two motives, and especially the second, highlight the importance to hackers of reputation in the eyes of their peers.

Raymond’s third motive for hackers to contribute to open source is economic. Open source, and its community and culture, do not exist in a vacuum; they are embedded in a larger economic system. A reputation built in open source may lead to rewards in and from that economic system, in the form of paid employment. Lerner and Tirole (2002) term this “the career concern incentive” for contributing to open source. When we turn to from the anthropological to the economic incentive, we retain focus on reputation as a motive for hackers. We move, though, from reputation in the eyes of fellow hackers to *the reputation of hackers in the eyes of vendors* (and other potential employers, such as customers).

The account of hacker motivation given by Raymond (2001), and drawn upon by the economists Lerner and Tirole (2002), suggests that it is useful to regard hackers as reputational entities, and to regard other hackers and vendors as the key stakeholders. This is not to assume that reputation-seeking is the sole explanation of open source contribution.

Fortunately, there is empirical evidence on hacker motivation, which allows a closer look at reputation as a motivation. The surveys by Lakhani and colleagues (2002) and by Ghosh and colleagues (2002) asked hackers about their motivations. When the motives are initially ranked by response, reputation is not among the leaders. In the former survey, reputation in the open source community was acknowledged as a motivation by 11% of hackers, and professional status by 17.5%. These reputation-based motives ranked far behind motives such as intellectual stimulation (44.9%) and skill improvement (41.3%), and also behind several other motives. In the latter survey, reputation in the community was acknowledged by 9.1% of hackers as a motive to join, and by 12% as a reputation to stay. Again, several other motives ranked higher, with skill improvement (70.5% to join, 78.9% to stay) at the top of the list. Improvement of job opportunities (23.9%/29.8%) is reputation-related, in that a hacker's reputation in the eyes of vendors (or other potential employers) will enhance job opportunities; but there are several other factors, notably skill improvement, that will lead to similar enhancement.

A closer look at the data suggests a stronger role for reputation-related motives. Lakhani and colleagues (2002) conducted cluster analysis, yielding four clusters of hackers: Believers, Skill Improvers, Fun Seekers, and Professionals. Professionals (21% of respondents), make up the most reputation-conscious of the clusters. Professional status was fourth among motives in this cluster, being cited by 25% of professionals. This is the cluster in which reputation in the open source community was highest (14%). Ghosh and colleagues (2002) asked hackers, not only about their motives, but also about their expectations of other hackers. 32.3% said they expected that they, and their work, would be respected by other hackers. This makes an interesting contrast with the far smaller percentage (9%/12.1%) who acknowledged reputation in the community as one of their main motives. It may be that more hackers expect respect and reputation than are motivated by such considerations. It may also be that hackers, when asked about their motives, understate reputation, preferring to acknowledge "worthy" motives such as skill improvement. In other words, the reported importance of reputational motives may be attenuated by social acceptability bias, and be revealed more fully when respondents are given the opportunity to acknowledge tham more comfortably.

In sum, the importance to hackers of reputation in the community may not be as widespread a motive as Raymond (2001) suggested, but it is probably more prevalent than would appear at first glance from the data. There is an identifiable subset of hackers (Lakhani et al's (2002) Professionals) for whom reputation, particularly professional reputation, is a stronger motivator than for the open source community as a whole.

Reputations of Vendors

The main difference between the supply chains in Figure 1 is the presence of the vendor-customer relationship in the "newer" chain. The *reputation of vendors in the eyes of customers* is a key aspect of this relationship. It is worth spending a few sentences to clarity the relationships amongst *reputation,* one of the key terms defined earlier in this article, and two other terms more specific to vendor-customer transactions: *corporate image* and *brand image*. Again, we turn to Bromely in matters of definition. The terms corporate image and corporate reputation carry much the same meaning (Bromley, p. 13; although other writers on reputation, including Davis (2003), relate the terms differently). Brand image is different, because it refers to a specific range of products. The extent of the difference between brand image and corporate reputation depends on the corporation's level of diversification: the wider the diversification, the wider the difference between the reputation of the whole corporation and a particular brand of that corporation (Bromley, p. 159).

The gap between brand image and corporate reputation is very extremely narrow for highly focused firms. Red Hat is such a firm, with a mission to be "the single trusted Linux and open source supplier" (Red Hat, 2003). Its firm commitment to open source—and to reputation-building—is embodied in the Red Hat "always open" promise: all code produced by Red Hat will be open source (Moody, 2002, p. 241). It is by building a reputation as the open source leader that Red Hat seeks to be the vendor of choice for Linux—and the high-margin services that complement the software itself. Bob Young (1999), who at the time headed the firm, argues that brand is just as important for Red Hat and Linux as it is for Heinz and ketchup.

IBM presents a contrast with Red Hat, here as elsewhere. It is less focused than Red Hat, in that it is a large, diversified firm. In particular, it can offer a wider range of complements to Linux, including hardware as well as services. IBM makes no promise to open the source code of all its software. Nevertheless, its commitment to make all its hardware platforms Linux-friendly established its reputation as a vendor serious about Linux. IBM's more general reputation was established long before there was an open source software movement. The adoption of Linux by IBM is notable for its effect on Linux, and so will be discussed in under the next section, which deals with reputations of open source projects.

Before leaving this account of vendor reputation, it is appropriate to consider the *reputation of vendors in the eyes of hackers,* since vendors cannot afford to take hackers for granted. Hackers have a choice whether to write open source code. They have a choice as to which features to work on. Hence "outsourcing" the development of vital software to the hacker community represents a risk to vendors. There are several ways in which vendors can reduce this risk.

Cultivating a good reputation in the eyes of hackers is one of these methods of risk reduction. For Feller and Fitzgerald (2002, p. 120), the most important characteristic of open source vendors such as IBM and Red Hat is that they seek "to establish a trust-filled relationship with the OSS [open source software] community." Such vendors do not just use the code produced by the community; they also give back to the community. One way in which they do this is simply making it the job of some of their employees to contribute code to open source projects.

It might be just as accurate to say that these vendors contribute by employing hackers who already have a track record in open source, and paying them to continue their contributions. A notable example of this is Red Hat's employment of Alan Cox, one of Linus Torvalds' lieutenants on the Linux project. Moody (2002, p. 308) states that this "acts as guarantee that Red Hat will always be a responsible guardian of the Linux community's interests." This reinforces Red Hat's "always open" promise, which is just as powerful in terms of reputation in the eyes of hackers as it is in terms of reputation in the eyes of customers.

There are important links here with the above-discussed issues of hacker reputation. Cox's reputation is a more powerful incentive for a vendor to hire him than any resume could ever be. His reputation among his fellow hackers is such that there is no question that his contract with Red Hat will undermine his allegiance to the open source movement (Moody, 2002).

Although Red Hat and IBM are useful examples of open source vendors, it would be premature to conclude an account of vendor reputation in open source software without reference to the reputations of two further vendors: SCO and Microsoft. The former has already been referred to as hated in the industry. The hatred is of course most intense in the open source community. The reputation of Microsoft within this hacker community has long been, to say the least, strongly negative. Raymond (2001: 183) described Microsoft as "corrupting the open protocols on which open source depends and choking customer choice." Although the phrasing is loaded, the assertions are firmly based on internal Microsoft communications; these are the "Halloween documents," so called because of the date in 1998 around which they were leaked. They were subsequently published, with commentary by Raymond, on the OSI web site.

The Halloween documents noted that open protocols enable open source to establish a presence in the server software market. (HTTP, the protocol that enables the Apache server to communicate with any HTTP-compliant web browser, is an example of such a protocol.) They suggested Microsoft-proprietary protocols (which may be existing protocols, extended so that they can no longer be considered open, or new protocols) as a barrier to future inroads by open source software. Such protocols would prove a barrier to any software, open source or not, written without Microsoft's cooperation. It is hard to see how documents advocating such closed systems could fail to damage Microsoft's reputation in the open source community—and beyond.

Reputations of Open Source Projects

The reputations of specific open source projects are important for several reasons. One concerns hacker motivation. If a hacker is interested in contributing to open source but is constrained as to how much time he can devote, he has to decide which project(s) to join. Reputation may be among the criteria he uses in making the decision. Once again, this is not a claim that reputation is the only variable of interest. For example, the fit between the project and the hacker's skills—either those he already has, or those he wishes to develop—will be a key consideration. But projects within the open source movement do differ as to reputation, and this may make the difference as to choice of project.

A project's reputation may relate to process, to output, or to both. Its process-based reputation relates to such matters as ease of getting on with existing contributors. Its output-based reputation relates to the code itself, and the use to which it is put. In what Raymond (2001) describes as the hacker gift culture, how fine a gift is the project's code to the community? The answer depends on several factors, perhaps the most obvious being how widely used the code is.

Wide use is one of the reasons that the reputations of Linux and Apache are as high as they are. Another reason is that each attains market share at the expense of a Microsoft product (Windows and IIS respectively).

Samba's reputation among hackers also comes in part from a contribution to the struggle against Microsoft. In order to emulate a Windows file server, it was necessary to reverse engineer a Microsoft protocol. Hence Samba opened what would otherwise be a closed protocol. Moreover, in doing so, it enabled the benchmarking of open source against Microsoft file servers (Moody, 2002; Raymond, 2001).

While the *project's reputation in the eyes of hackers* relates to the supply side, the *project's reputation in the eyes of customers* relates to the demand side. In reputation, as in market share, the most prominent story is the the rise of Linux. Here is one description of Linux's reputation as of early 1999.

> [M]ost people thought of Linux as a strange tool created and used by hackers in dark rooms lit by computer monitors… most people thought that Linux was the work of a mad genius and his weirdo disciples. (Wayner, 2000: 4).

The source of the quotation is Wayner's book on competition between open source and proprietary software. The book opens with an account of the changes brought by the US Department of Justice against Microsoft. In January 1999 Richard Schmalensee (an economist of high reputation, and Dean of MIT's Sloan School of Management) testified that Microsoft did not have a monopoly on microcomputer operating systems, on the grounds that competitors to Windows existed. He offered Linux as an example of a competing operating system. Wayner's point is that the reputation of Linux seemed likely to undermine its credibility as a competitor to Microsoft Windows: the "weirdoes" did not sound like a significant competitive threat.

As quotes elsewhere in this article show, the reputation of the Linux *community* may not have changed radically since early 1999. But the reputation of the Linux *operating system* has certainly gone up, along with its use, in the business world. There is a positive feedback loop: the better the reputation of Linux as an appropriate platform for key applications, the more likely firms are to adopt Linux, and the better its reputation.

In terms of reputation, Linux has been helped by its relationship with vendors. Its relationship with Windows, and hence with Microsoft, is that of competitor. Linux, whether it is referred to as open or as free, presents a sharp contrast with Microsoft and its reputation for anticompetitive action. An executive at Merryl Lynch, where some vital trading systems run on Linux, states the need for “the right competitive dynamic” and “choices going forward” (Kerstetter, 2003, p. 80). The implication is using Linux rather than Windows is likely to increase competition among vendors and thus enhance choice for customers. This dovetails with the message that Bob Young (1999) delivered when he identified *control over software* as the main benefit of open source software to the customer.

Some of the ways in which Microsoft has sought to combat Linux have actually helped, rather than harmed, the reputation of Linux. When Microsoft witnesses testified in court that Linux was a competitor for Windows, they enhanced the legitimacy and reputation of Linux. The performance benchmarking of Linux and other open source software against their Microsoft counterparts had a similar effect. The first published benchmark favored open source over Windows. The next favored Windows; but it was noted that the benchmark, although run by a firm called Mindcraft, had been sponsored by Microsoft. The result was not an outcome of cheating, as some initially suspected; it turned out that in some uses, Windows was actually faster. Nonetheless, Moody (2002, p. 283) described Microsoft's handling of the benchmarks as a blunder.

> By arranging for GNU/Linux, Apache, and Samba to be benchmarked against Windows NT, Microsoft said in the most emphatic manner possible that that these were rivals… This was a significant shift from Microsoft's previous stance that GNU/Linux was not up to enterprise-level tasks, and nobody was using it anyway.

In contrast, the relationship between Linux and many other major vendors is good, and good for the reputation of Linux. One of the best illustrations of this, and one of the biggest single boosts to the reputation of Linux in the commercial world, came in January 2000, when IBM announced that it would make all its server platforms "Linux-friendly." As Moody (2002: 290) remarked, "it would be hard to think of a better symbol of the rise of Linux… than its appearance on the mightiest and most prestigious of corporate mainframes, the IBM S/390."

This IBM announcement was one of the most salient of many signals to customers—and to other vendors—that Linux merited serious consideration as an operating system in enterprise computing, as well as in smaller systems. This was not the first time that IBM threw its weight behind a disruptive technology. IBM's investments of reputational and financial capital provided impetus for the personal computer to revolutionize the use of IT (Rindova and Fombrun, 1999). Twenty years later, it was in open source software that IBM made substantial investments of reputation and money.

A further similarity between the PC and open source software is that each made considerable inroads within organizations from the bottom up; neither was imposed from above by senior management. Within large corporations, departments were able to deploy PCs at a price low enough that they did not have to involve the corporation's top management or its IT department. In 1996, almost every large US corporation was using Linux on some scale, usually without top management being aware of it (Moody, 2002, p. 101). By this time, Apache was already the most widely web server (p. 129).

Hence the information senior exectives received about Linux and Apache was likely to lead to a positive impression of these projects. The information from outside the corporation was that IBM and other major vendors supported Linux and other open source software. If they sought information from inside the corporation, they were likely to find that open source software already supported some systems. In particular, they might well find that Apache supported the corporate web page. Indeed, to a large extent open source software supported the web itself, and still does so.

Reputations of Open Source

Although press coverage tends to put Linux at center stage, it also acknowledges that Linux is part of a more general phenomenon. The *Business Week* cover story (Kerstetter, 2003) provides an example; while the cover announces the Linux uprising, and shows Tux, the penguin mascot of Linux, the article also covers other open source projects, and open source in general. Due to the prominence of Linux, the *reputation of open source in the eyes of customers* is to a large extent based on the reputation of Linux. It is appropriate at this point to recall that the open source movement came into being in part due to concern for reputation in the eyes of customers, who might be deterred by the free software movement (Perens, 1999).

The *reputation of open source in the eyes of governments* may well be even more important than its reputation in the eyes of customers. Governments are customers, and among the largest ones, for IT. However, governments are a special enough case to be treated as a stakeholder group distinct from private sector customers. First, governments are particularly large customers; for example, in 2000, European governments (at levels ranging from local to federal) spent $7.8B on software (Festa, 2001). Second, government policy with respect to software may set the tone for the country, and hence influence the private sector. Third, it is as important as it is obvious to note that governments are political entities.

It is not surprising, then, that politically and ethically charged terms such as *free software* and *software libre* seem to resonate with governments. Even if, as the founders of the OSI feared, such terms deter private sector customers, they seem to have the opposite effect on some governments. In particular, they seem to represent freedom from Microsoft and its domination of the software industry. Various proposed legislation in Europe, Latin America, and Asia seeks to avoid dependence on Microsoft, which has a reputation as an America monopolist. Open source software is also "software gratis," and so is of particular interest to poorer countries (Festa, 2001), further enhancing its reputation.

Finally, it is worth noting that open source software is of interest to stakeholders even larger than countries. For example, the European Commission's Working Group on Libre Software concluded:

> … that those countries and companies which adopt open source technologies in the short term will have a huge competitive advantage, and that society in general can benefit a lot from this early adoption. (Working Group, 2001: 28)

DISCUSSION

There is precedent for discussion of reputation in open source software. Raymond's (2001) treatment of hackers as reputational entities is already a classic of the open source literature. Young (1999), in remarking on the importance of brand, in effect treated corporations as reputational entities, and emphasized customers as stakeholders. The contribution of this article to the open source literature is to provide a more systematic account of reputation. It does so, first by extending the set of reputational entities to include open source projects and the open source movement as reputational entities. Second, as Figure 2 shows, the article identifies key stakeholders for each reputational entity. Third, it depicts reputation as a dynamic, multilevel phenomenon within open source software; for example, IBM’s announcements of support for Linux changed for the better the reputation of Linux.

The article also contributes to the literature on strategic management, in which reputation is of interest as a potential source of competitive advantage for firms. Charles Fombrum (in Davies, 2003, p. x) states that “managers build strategic advantage by generating favorable perceptions about the company in the eyes of key stakeholders.” The current case study includes evidence for the descriptive side of this statement, in that it includes accounts of the reputation-building activities of vendor firms. It does not include direct evidence that these activities lead to competitive advantage.

However, there are at least three arguments that the link between reputation and advantage is likely to be particularly strong in the open source arena. The first was stated in the introduction to this artice. The key input to software is the source itself. When this input is free, open, and hence perfectly imitable, firms must seek advantage in other resources, and intangible resources such as reputation are likely to provide particularly important.

The second argument arises from the observation that open source is comparatively new, rapidly changing, and thus particularly uncertain. In such environments, reputation is a mechanism for reducing uncertainly, and so is an asset likely to lead to advantage. This is the argument advanced by Kotha, Rajgopal, and Rindova (2001). They made this argument with respect to internet firms, but their logic is applicable also to open source firms. Their logic found support from the empirical portion of the article, in which reputation-building activities predicted competitive success in a sample of pure-play internet firms.

The third argument arises from consideration of pure-play open source firms, and the dramatic culling of their ranks amidst the dot.com crash. The open source world was linked to the dot.com world, and so the crash brought down the reputation of open source in the eyes of investors. Moody (2002, p. 324) notes that open source businesses "have seen their valuations slashed and additional funding evaporate, and several have shut down completely." But he remarks on "Red Hat's relative stability amidst the growing troubles of its fellow GNU/Linux distributors." It is Red Hat—a firm particularly assiduous in cultivating reputation in the eyes of customers and other stakeholders—that has proved most stable. In a turbulent period during which survival, let along stability, means outperforming similar firms, Red Hat has proved more than competitive.

This account of reputation in open source software has two main limitations. The first is that is far from comprehensive. It considered four types of reputational entity, and two stakeholder groups for each entity. Since some stakeholders are relevant to multiple entities, and some are themselves relevant as reputational entities, only six types of actor were considered and appear in Figure 2. It would be accurate to add further links, such as hacker-customer, to the figure. It would be possible to add further boxes, with appropriate links. For example, Young (1999) emphasizes the importance of the reputation of vendors in the eyes of investors; investors comprise a stakeholder group that could well have been added. The second limitation is that the case study offers no new data, drawing as it does from existing sources.

This case study of reputation in open source software suggests several directions for future research. One direction—the addition of additional entities—is implicit in the description of the first limitation. Other directions lie in quantitive research into issues discussed qualitatively above, such as the relationship between reputation and performance.

Yet another direction arises from open source as a model of “private-collective” innovation. Von Hippel and von Krogh (2003) remark that open source software development is collective action to produce a public good, but that the costs are privately borne. They are born either by hackers expending their own time and other resources, or by firms paying employees to contribute to open source projects. Hackers contribute because they derive private benefits from doing so; for example, they may enhance their programming skills or reputations. In contrast to other cases of collective action, free riders may not be a problem. People who use the code without contributing to it contribute to its market market share and reputation. However, a problem may arise when we switch the level of analysis from hackers to vendors. Red Hat and IBM contribute to the public good that is Linux. Other firms can and do free ride on their efforts, by competing in the Linux market without making any contribution to Linux development. Contributing firms, like contributing individuals, reap private benefits, including learning and reputation. What is the optimum balance between contribution and benefit?

In conclusion, reputation is an important factor in open source software, and open source is a very suitable field in which to conduct research into reputation and competitive advantage. Moreover, although open source is a very important type of software, it is more than that. Steven Weber (2004, p. 224) points out that it is “a way of organizing production, of making things jointly,” and hence may be applied in fields ofther than computer software. Whatever the field in which it is applied, reputation is likely to be one of the motives to contribute to the production, and one of the factors influencing the adoption of the product.

REFERENCES

All URLs are current as of November 4, 2004.

Ante, S. E. 2001. Big Blue’s big bet on free software. *Business Week,* December 10: 78-79.

Barney, J. 1991. Firm resources and sustained competitive advantage. *Journal of Management,* 17: 99-120.

Bromley, D. B. 1993. *Reputation, image and impression management.* New York: Wiley.

Davies, G. 2003. *Corporate reputation and competitiveness.* London: Routledge.

Eisenhardt, K. M. 1989. Building theories from case study research. *Academy of Management Review,* 14: 532-550.

Feller, J., & Fitzgerald, B. 2002. *Understanding open source software development.* London: Addison-Wesley.

Festa, P. 2001. Governments push open source software. *CNET news.* http://news.com.com/2100-1001-272299.html.

Ghosh, R. A., Glott, R., Krieger, B., & Robles, G. 2002. *Survey of developers.* http://floss.infonomics.nl/report/FLOSS_Final4.pdf.

Kerstetter, J. 2003. The Linux uprising. *Business Week*, March 3: 78-84.

Kerstetter, J. 2004. The most hated company in tech. *Business Week,* Feb 22: 78-80.

Kollock, P. 1999. The economies of online cooperation: Gifts and public goods in cyberspace. In M. A. Smith & P. Kollock (Eds.), *Communites in cyberspace:* 220-239. London: Routledge.

Kotha, S., Rajgopal, S., & Rindova, V. 2001. Reputation building and performance: An empirical analysis of the top-50 pure internet firms. *European Management Journal*, 19: 571-586.

Lakhani, K.R., Wolf, B., Bates, J., & DiBona, C. 2002. *Hacker survey.* http://www.osdn.com/bcg/.

Lerner, J., & Tirole, J. 2002. The simple economics of open source. *Journal of Industrial Economics,* L: 197-234.

Moody, G. 2001. Rebel code: *The inside story of Linux and the open source revolution.* Cambridge, MA: Perseus.

Open Source Initiative. Open source definition. http://www.opensource.org/docs/definition.php.

Perens, B. 1999. The open source definition. In C. DiBona, S. Ockman, & M. Stone (Eds.), Open Sources: Voices from the Open Source Revolution: 171-196. Sebastopol, CA: O'Reilly.

Raymond, E. S. 2001. *The cathedral and the bazaar: Musings on Linux and open source by an accidental revolutionary.* Sebastopol, CA: O'Reilly.

Rindova, V. P, & Fombrun, C. J. 1999. Constructing competitive advantage: The role of firm-constituent interactions. *Strategic Management Journal*, 20: 691-710.

Stallman, R. 2002. *Free software, free society*. Boston, MA: GNU.

Von Hippel, E, & von Krogh, G. 2003. Open source software and the 'Private-Collective' innovation model: Issues for organization science. *Organization Science*, 14: 209-223.

Wayner, P. 2000. Free for all: *How Linux and the free software movement undercut the high-tech titans.* New York: HarperCollins.

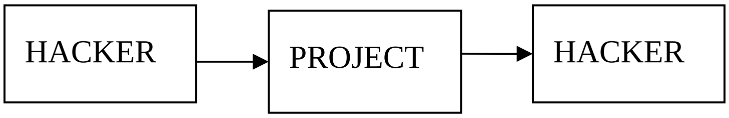
Weber, S. 2004. *The success of open source.* Cambridge, MA: Harvard University Press.

Working Group on Libre Software. 2000. *Free software/open source: Information society opportunities for Europe?* http://eu.conecta.it/.

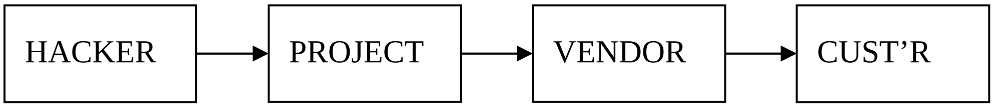
Young, R. 1999. The GNU operating system and the free software movement. In C. DiBona, S. Ockman, & M. Stone (Eds.), Open Sources: Voices from the Open Source Revolution: 113-125. Sebastopol, CA: O'Reilly.

FIGURE 1

Open Source Supply Chains



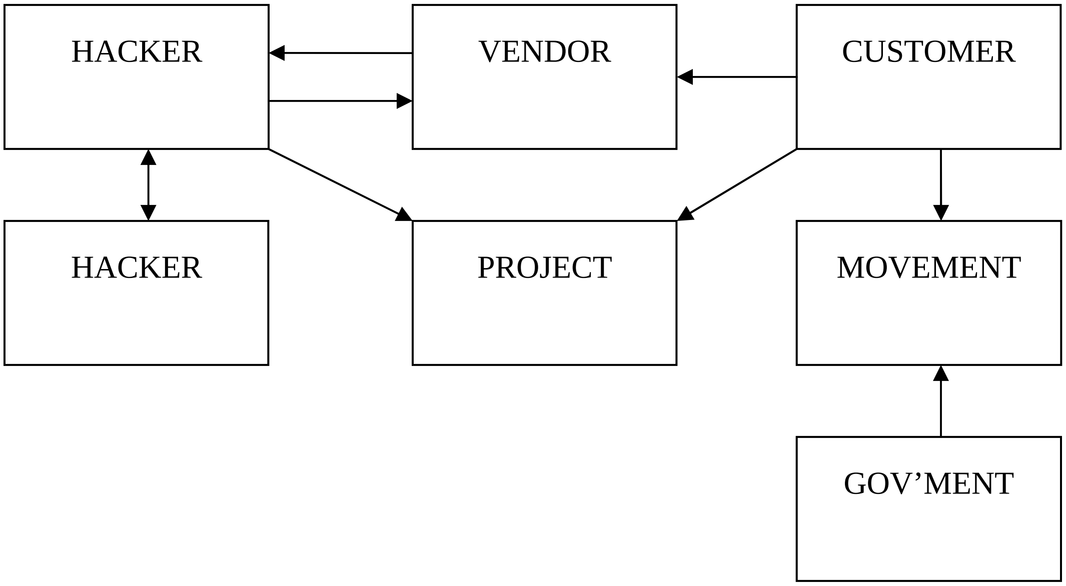
1.1 Chain in first era



1.2 Chain new in second era

FIGURE 2

Reputational Entities and Stakeholders



An arrow from entity A to entity B denotes the reputation of B in the eyes of A. Hence B is the reputational entity, and A the stakeholder. For example, the lower right arrow denotes the reputation of the open source movement in the eyes of governments.