Chapter 5

Patterns of Open Innovation in Open Source Software

Joel West
Associate Professor, College of Business, San José State University
One Washington Square, San José, CA 95192-0070 USA
+1-408-924-7069; fax: +1-408-924-3555
Joel.West@sjsu.edu

Scott Gallagher
Assistant Professor, College of Business, James Madison University
Harrisonburg, VA 22807 USA
+1-540-568-8792; fax: +1-540-568-2754
gallagsr@jmu.edu

26 October 2005

Submitted for
Henry Chesbrough, Wim Vanhaverbeke and Joel West, eds.,
Open Innovation: Researching a New Paradigm, Oxford University Press (2006).

1. INTRODUCTION

Models of open innovation offer the promise that firms can achieve a greater return on their innovative activities and their resulting intellectual property (IP). Open innovation models stress the importance of using a broad range of sources for a firm's innovation and invention activities, including customers, rivals, academics, and firms in unrelated industries while simultaneously using creative methods to exploit a firm's resulting IP (Chesbrough, 2003a). While the use of external sources of innovation is nothing new, recently some of the most successful high-tech firms have been those that utilized open innovation rather than more traditional "vertically integrated" innovation approaches.

The open innovation paradigm is often contrasted to the traditional vertical integration model where internal R&D activities lead to internally developed products that are then distributed by the firm. Consistent with Chapter 1, we define open innovation as systematically encouraging and exploring a wide range of internal and external sources for innovation opportunities, consciously integrating that exploration with firm capabilities and resources, and broadly exploiting those opportunities through multiple channels. Therefore, the open innovation paradigm therefore goes beyond simply the externalization of research and development as identified by von Hippel (1988). Rather than just a shift in the technical production of intellectual property, open innovation reflects a transformation of how firms use and manage their IP.

Over the past 20 years, an increasingly popular example of open innovation has been open source software, exemplified by the Linux operating system. Open source software involves collaboration between firms, suppliers, customers or makers of related products to pool software R&D to produce a shared technology. At the same time, open source IP policies mean that this shared technology is available to potential buyers at little or no cost.

Together, the shared production and low cost of open source software has forced firms to reconsider the proprietary business models used by commercial software companies for the past 25 years. The essential issue for a firm's business model is, "How does the firm create value for the customer while simultaneously extracting some of that value for itself?" The rise of open source software has enabled a wave of experimentation in software business models that is ongoing even today.

In this chapter, we consider how open source addresses what we identify as three management challenges of open innovation: maximizing the use of internal innovation; incorporating external innovation into the firm; and motivating a supply of such external innovation to support the firm. We then classify the strategies taken by companies selling open source software based on their business models, and suggest how this fits into broader issues of open innovation.

2. CHALLENGES OF OPEN INNOVATION

The pace of technological advance has often been subdivided into two phases: invention (a scientific breakthrough) and innovation (commercialization of the invention) — a distinction Nelson and Winter (1982: 263) attribute to Schumpeter (1934). This split parallels a similar bifurcation between research and development, where inventions come from basic research and innovations from the development group. Many organizations, however, define additional phases between the two extremes, as with Intel's "advanced development" step (Tennenhouse, 2003). Others have attempted to subdivided innovation into "radical" and "incremental", where the former more closely resembles the "invention" concept (e.g. Leifer et al, 2000).

Like Nelson and Winter, we use "innovation" in its broadest sense to refer to the entire process by which technological change is deployed in commercial products. Such innovation

may incorporate formally protected intellectual property (such as patents or copyrights) that is difficult to imitate, or it may reflect codified knowledge that is readily imitated and at best provides a transient competitive advantage.

In contrast to earlier models and "fully integrated innovators" like AT&T (now Lucent) Bell Labs and IBM which do basic research through commercial products, open innovation celebrates success stories like Cisco, Intel and Microsoft, which succeed by leveraging the basic research of others (Chesbrough, 2003a). Under this paradigm, firms exploit both internal and external sources of innovation, while maximizing the returns that accrue from both sources (Table 5.1).

Tactics that embody an open innovation approach include exploiting knowledge spillovers and consulting with venture capitalists, while also using both inbound and outbound licensing of key technologies. Although earlier frameworks acknowledged the role of external knowledge and "accidental" internal discoveries, it is the systematic encouragement and integration of these issues coupled with creative exploitation of IP that distinguishes open innovation from earlier innovation models.

Firms practicing open innovation face three inherent managerial challenges:

- *maximization*. Firms need a wide range of approaches to maximize the returns to internal innovation — not just feeding the company's product pipeline, but also outbound licensing of IP, patent pooling and even giving away technology to stimulate demand for other products.
- *incorporation*. The existence of external knowledge provides no benefits to the firm if the firm cannot identify the relevant knowledge and incorporate it into its innovation activities faster than its rivals and new entrants. This requires scanning, recognition, absorption and also the political willingness to incorporate external innovation.

- *motivation*. Open innovation assumes an ongoing stream of external innovation, but this raises the question of who will continue to generate IP externally that can be used by the firm? We suggest that over the long term, firms must cultivate ways to assure continued supply of relevant external technologies and IP.

We discuss each of these challenges below.

2.1 Maximizing Returns to Internal Innovation

A central concern to open innovation is how to best use the internal R&D capabilities of the firm to maximum advantage. Those capabilities can be used for:

- generating innovations to be internally commercialized (the traditional model);
- building absorptive capacity and using that capacity to identify IP laying beyond the boundaries of the organization, i.e. external innovation;
- generating innovations that generate returns through external commercialization (e.g. licensable patent portfolios or spin-offs); and
- generating IP that does not produce direct economic benefit, but indirectly generates a return through spillovers or sale of related goods and products.

Successful approaches will often combine a variety of these approaches. For example, to identify promising technologies, Intel establishes research labs near top university research groups, with open flows of information in both directions. If an innovation proves promising, Intel recruits the top academic researchers to help commercialize the technology and see it through to production (Tennenhouse, 2003).

This approach can be used cooperatively as well, as with the GSM patent pool assembled by European telephone makers in the early 1990s. While the patents were often the result of basic research, contribution of a patent to the patent pool allowed firms to have favorable access to all

of the IPR of the GSM standard, creating a cost advantage for European pool participants over potential Asian rivals (Bekkers et al, 2002).

2.2 Incorporating External Innovations

While firms may generate considerable internal knowledge to support their innovation activities, von Hippel (1988) identified four external sources: 1) suppliers and customers; 2) university, government and private laboratories; 3) competitors; and 4) other nations. Various models have been developed to explain how firms can exploit external knowledge. Perhaps the simplest method is to imitate a competitor: such free riding on the product and market investments of rivals is a common way for firms to overcome a first mover strategy (Lieberman and Montgomery, 1998). Consulting with customers can provide firms ideas about discovering, developing, and refining innovations (von Hippel, 1988). Public sources are also an important source of knowledge, for example government R&D spending was identified almost 50 years ago as an important stimulus for private R&D (David, Hall, and Toole, 2000). University research is one key source of external innovation for some industries (Fabrizio, Chapter 7).

The managerial challenges of utilizing external knowledge then center around identifying useful external knowledge, and then integrating that knowledge with the firm. For example, for new products there are significant trade-offs between innovation speed, development costs, and competitive advantage for relying on external rather than internal learning (Kessler, Bierly, and Gopalakrishnan, 2000). Environmental scanning, competitive intelligence, sponsored research, and membership in relevant trade organizations is a way to uncover external knowledge opportunities. Developing absorptive capacity, via internal R&D investments appears to be an important prerequisite for converting external knowledge into internal innovation (Cohen and Levinthal, 1990).

Even if external innovations are identified, that does not mean they will be incorporated into the firm's product strategies. A firm that was once highly successful at the vertically integrated innovation model will tend to believe its innovations superior to any competing ideas from outsiders. For example, flush from its successful user interface innovations of the 1980s, engineers at Apple Computer rejected external ideas in areas such as handheld computers, adopting the phrase "not invented here" to describe such rejection (Kaplan 1996: 156).

2.3 Motivating Spillovers

With external innovation, there is often an unstated assumption that the supply will continue. But what happens if everyone tries to use others' basic research? Will "innovation benefactors" — such as government and nonprofit research sponsors — continue to provide as fertile a field (Chesbrough, 2003b)? If commercial firms do not realize a return on their innovative activities, they will tend to under-invest in innovative activities that are either highly risky (e.g. basic research) or that are easily imitated by free-riding competitors. Therefore, we consider the incentives for generating the knowledge spillovers at two levels — the individual and the organizational. This also has important societal implications, such as the funding of basic research and development by national governments. At the simplest level, such incentives can reflect direct financial payment to the innovators, but innovation recipients can and do exploit a wide range of alternatives.

Motivating individuals to generate and contribute their IP in the absence of financial returns is a significant management challenge for an open innovation approach. One of the simplest models of motivation is expectancy theory that posits that individuals are motivated when both valence, the attractiveness of a reward, which can be either intrinsic (e.g. happiness) or extrinsic (e.g. fame or fortune) and instrumentality, the path to that reward, are present (Lawler, 1971).

The integrated innovation model solved this challenge though extrinsic compensation from the firm coupled with adherence to traditional professional scientific norms, e.g. scientific freedom, support for publishing. The external model doesn't formally address individuals but appears to rely upon others, e.g. universities, to partially or wholly provide motivation.

The incentives for organizations to contribute spillovers fall into two categories. In the one case, the innovation benefits the innovator and there is no loss from sharing that benefit with others. For example, customers often share their innovations with their vendors if it means improved products in the future (von Hippel, 1988). And of course suppliers invest in innovations to sell more products, as when Intel increases the performance of microprocessors that it sells to Dell.

Spillovers to a direct competitor are more problematic, but still are economically rational under conditions of "co-opetition". Firms in the same industry complement each other in creating markets but compete in dividing up markets (Brandenburger & Nalebuff, 1996: 34). So if a firm stands to benefit from an innovation that grows the market, it will accept spillovers if the return from its share of market growth is attractive enough. For example, Intel's venture capital arm makes investments to grow the ecosystem around its microprocessors, even though a small portion of that investment accrues to AMD, the second-place maker of microprocessors (Chesbrough, 2003a: 125-131).

3. RESEARCH DESIGN

Our three challenges led to three related research questions:

- What circumstances motivate firms to embrace open innovation approaches as part of their R&D efforts?

- Why would for-profit firms commit their intellectual property as well as *ongoing* human resources to an effort that they know will benefit others, including competitors?
- Why do individuals contribute their IP to a project that benefits firms without receiving financial remuneration?

We chose to study the use of external innovation in the software industry, in particular the "open source" movement. Open source and other collaborative development techniques in the software industry offer examples of how the three key challenges of open innovation can and have been addressed by commercial firms. Open source also offers an approach to address what West (2003) refers to as an "essential tension" in information technology innovation: appropriating the returns from an innovation versus winning adoption of that innovation.

3.1 Traditional Software Production Models

Modern software engineering techniques are based on an abstraction of the software design to minimize interdependencies between individual components of a complex system. This enables both specialization in the software development effort and reuse of existing technology to facilitate cumulative innovation (Krueger, 1992). As such, software development efforts are highly modular within or between firms (Sanchez and Mahoney, 1996). This specialization in software mirrors the specialization of other forms of industrial production.

The output of this software production effort is an information good, marketed to individual consumers or businesses. Even if software is distributed on a physical medium such as a magnetic tape, floppy, or optical CD-ROM, the value of the software is tied to the information on those tapes or discs, not the value of the physical medium or the cost of reproduction. As with

any other good, firms price their software to appropriate some but not all of the utility derived by consumers.

While the marginal cost of each copy of a software application is thus very low, initial development costs are quite high. Commercial firms spend millions to hundreds of millions of dollars to make a commercial software release, and in most cases are forced to re-invest comparable amounts every one to three years to continue their ongoing revenue stream. This gives software firms huge supply side economies of scale: the first copy of Windows is very expensive, but additional copies cost almost nothing.

Interestingly, many software firms also benefit from not only supply side economies of scale but demand side economies as well. Most software users would face significant switching costs in using some other software package, due to some combination of retraining user skills and converting data stored in proprietary file formats. As Arthur (1996) observes, software thus has tremendous positive returns to scale, generally allowing only one (or a small number) of winners to emerge. These winners are tempted to extract rents from their customers by increasing prices and creating additional switching costs to protect those rents (Shapiro and Varian, 1999). From these production economics, commercial software firms seek to build complete systems to meet a broad range of needs, in hopes of forestalling potential competitors and protecting high gross profit margins.

These supply and demand side economies of scale have fueled ongoing consolidation in the software industry. Microsoft is the most obvious beneficiary of these scale economies of having exploited both supply and demand side economies of scale, having captured not only the desktop operating system market but the most common office applications as well. However,

consolidation has also occurred in other segments, such as in enterprise software as with Oracle's 2005 acquisition of PeopleSoft.

3.2 Open Source Software

Open source software emerged as a reaction to the proprietary software model, with two direct antecedents. One was the open systems movement — centered around Unix and its variants — reflecting an attempt by customers to reduce their dependence on proprietary software vendors. Another antecedent was the creation of university research software during the 1980s, including the BSD variant of Unix from UC Berkeley. However, the open source (and related free software) movements differed from the open systems and university initiatives by focusing on user rights, especially establishing a series of principles to enable shared development and perpetual use rights. The rapid rise of open source in the late 1990s was enabled by the dissemination of software tools and the Internet to facilitate the shared distribution production of software (West and Dedrick, 2001, 2005).

Open source software differs from proprietary software in two ways: in its intellectual property philosophy and how it is produced (West and O'Mahony, 2005). Researchers have considered each of these in turn:

IP Philosophy. Differences between the open source movement and proprietary software are most dramatic over the treatment of the source code of software. The definition of "open source" requires free redistribution of software in source code form and the right to modify the software. An allied but distinct group, the "free software" movement, also requires that software remain perpetually "free" by compelling users to return all modifications, enhancements and extensions (West and Dedrick, 2005). These conditions are enforced by a wide range of software licenses

approved by one or both factions (Rosen, 2004). In comparison, proprietary firms aggressviely protect their software source code.

Production via Collaboration. Another important difference between open source and proprietary efforts is the collaborative open source production process. In examining two projects, Mockus et al (2002) concluded that the development was controlled by a small group, but received occasional error correction from a much larger group of developer-users. Considering a broader group of projects, Healy and Schussman (2003) showed that participation was highly skewed according to a power-law (log-log) distribution: a handful of projects attract most of the attention, and participation in these projects is heavily skewed towards a small number of contributors.

What motivates individuals to contribute to open source projects? Consistent with expectancy theory, empirical research on more successful projects (Hars and Ou, 2002; Hertel et al, 2003; Lakhani & von Hippel, 2003) found three general categories of contributor motivations: *direct utility,* either to the individual or to one's employer; *intrinsic benefit* from the work, such as learning a skill or personal fulfillment; and *signaling* one's capabilities to gain respect from one's peers or interest from prospective employers.

Firm Participation. Despite these key differences between open source and proprietary software, for-profit IT producers have gotten involved in open source software. Why? West (2003) showed that in a positive returns environment, such firms made limited use of openness to win adoption of their technologies, either by opening portions of the their technology or providing partly open access to key technologies. O'Mahony (2003) demonstrated that while open source projects employ intellectual property licenses that grant use rights to a broad class of users, they also used a combination of legal and rhetorical tactics to aggressively safeguard the

independence and permanence of their development efforts, particularly when negotiating with firm seeking to advance potentially conflicting proprietary interests.

But in addition to these independent, organically grown open source projects, firms have also sponsored their own open source projects. Such sponsored projects differ from the organic ones in both production and governance, as the bulk of the resources are provided by the sponsor to achieve its goals (O'Mahony and West, 2005). Firms have also experimented with sponsoring collaborative software production that is similar to open source in most but not all dimensions, such as "gated" communities in which collaboration is available to some but not all (Shah, 2004).

Here we consider the strategic motivations and business models of firm involvement in open source software projects, both those projects that begun autonomously and those created by sponsors to directly advance firm goals.

3.3 Research Methods

Given the comparative newness of the open innovation paradigm, we chose to use a theory-building approach grounded in the context of rich data. This draws on established procedures for generating theory from qualitative data (Glaser and Strauss, 1967), as well as management studies that employ the inductive method to draw theory from a set of case studies (Harris and Sutton, 1986; Eisenhardt, 1989).

Our research efforts included both primary and secondary sources. From 2002 through mid-2005, one author conducted 56 interviews with 46 informants representing 30 organizations. Of the 30 organizations, 18 were for-profit companies in the I.T. industry. The interviews also included 7 major open source projects, as well as other professionals indirectly involved in the industry. Interviews typically ranged from 45-90 minutes, and most were tape recorded for later

consultation. This was supplemented by observation of (and, in some cases, participation in) six Silicon Valley industry conferences and seminars from August 2003 through March 2005 that focused solely on open source software. This primary data were complemented by a secondary data sources, we also incorporated secondary data. During the observation period, one of the authors reviewed approximately 800 news articles from trade journals, business press and websites related to Linux and other open source topics. We also reviewed prior research from technology management, sociology and computer science on open source collaboration.

From our data, we sought to identify regular patterns of open innovation among IT firms actively participating in open source projects, and (where possible) to identify the goals and motivations for such innovation strategies. Once we identified key patterns, we shared preliminary conclusions with a subset of informants and used their feedback to refine our conclusions.

4. OPEN SOURCE AS A MANIFESTATION OF OPEN INNOVATION

Open source as an open innovation strategy has two key elements: shared rights to use the technology, and collaborative development of that technology. Unlike many individual participants, firms must also consider a third issue: capturing an economic return to justify their investment.

Here we consider four approaches for external innovation in software, where firms have both invested in open innovations and benefited from them (Table 5.2).

4.1 Pooled R&D: Linux, Mozilla

A familiar model of open innovation is that of pooled R&D. While cooperative research often occurs to save costs, prior research also suggests that firms cooperate in cases where they cannot appropriate spillovers from their research (Ouchi and Bolton, 1988), in areas that are

highly risky or for industries most dependent on advanced science (Miotti and Sachwald, 2003). They also tend to collaborate in industries with strong vertical relationships (Sakakibara, 2001), with firms that share overlapping technological capabilities and are in the same industry or sector (Mowery, Oxley and Silverman, 1998).

Two highly visible open source examples are support for the Linux operating system through the Open Source Development Lab, and the Mozilla web browser. For both, firms donate R&D to the open source project while exploiting the pooled R&D of all contributors to facilitate the sale of related products.

A simple example is the Mozilla open source project, a descendant of the Netscape Navigator browser offered for a wide range of systems — Windows, Macintosh, Linux and at least 7 Unix variants. This browser was among the first commercial browser products ("Netscape Navigator", 2004). Navigator held more than two-thirds of the browser market until late 1997 — surpassing Microsoft's Internet Explorer — but only two years later the shares were reversed due to IE's bundling with Microsoft Windows (Bresnahan and Yin, 2004).

Netscape created the Mozilla open source project in 1998 and terminated all internal development of it in July 2003— deferring further work to the open source community. At this point, Unix system vendors such as IBM, HP and Sun were left without a supported browser, which they needed to sell Internet-connected workstations. Thus, each of them assigned software engineers to work with the Mozilla project, both to help keep the project moving forward and to assure that new releases are compatible with their respective systems[i].

The R&D cooperation in the Open Source Development Labs (OSDL) for Linux is more complex. Founded in 2000, the OSDL takes as its mission "To be the recognized center-of-gravity for the Linux industry; a central body dedicated to accelerating the use of Linux for

enterprise computing" ("Corporate Overview," 2004). In its first five years, the consortium began work on three projects: data center Linux, carrier grade Linux and desktop Linux.

The founders, sponsors and other members of OSDL and their motivations for supporting OSDL could be grouped into four broad categories: vendors of computer and telecommunications systems, producers of microprocessors, Linux distributors and support organizations, and developers of complementary software products (Table 5.3).

How do such projects address the three open innovation challenges? For firms participating in Mozilla, the *quid pro quo* is straightforward: systems vendors *maximize* the returns of their innovation by concentrating on their own needs (such as platform-specific customization), and then *incorporate* the shared browser technology into their integrated systems. However, the *motivation* challenge is not completely solved, in that the systems vendors assume a pool of individual open source contributors that sustain innovation in the core product.

For the OSDL, firms contributed their specialized knowledge (e.g. in telecommunications operations or microprocessor architecture) to build a common platform. OSDL resembles other self -supporting industrial research consortia, where firms pool interests towards a common goal, and assume they can both cooperate in supporting that goal and compete in selling their respective products.

However, both Mozilla and OSDL differ from typical consortia in two ways:

- *Spillovers are not controllable*. Many consortia reward members by limiting direct access to the consortium's research output to member-participants, limiting access to indirect spillovers. Open source licenses typically make it impossible to limit even direct access, allowing non-members to accrue many of the same benefits as members.

- *Contributions from non-participants*. The engineering contributions to these open source projects extend beyond the sponsoring companies to include user organizations, academics, individual hobbyists and other interested parties. Unless the corporate contributions eventually dwarf the individual ones, the projects must continue to motivate such contributions to survive.

Given these factors, an open source innovation model is inherently more "open" than a typical R&D consortium, both in terms of exploiting information from outside the consortium, and sharing that information back out to non-member organizations and individuals.

4.2 Spinouts: Jikes, Eclipse, Beehive

Open innovation can release the potential of technologies within the firm that are not creating value. In some cases, the technologies are no longer strategic, as with AOL Time Warner's decision to spinoff Mozilla into a stand-alone open source project after firing its Netscape development team (Hansen, 2003).

But in addition to spinoff (and, frequently, abandonment), firms also have opportunities to release more value from their technologies by situating them outside the firm, but at the same time maintaining an ongoing corporate involvement. Here we use the term "spinout" to refer to all cases where firms transform internal development projects to externally visible open source projects.

If a firm gives away its IP, how can such spinouts create value? One way is that the donated IP generates demand for other products and services that the donor continues to sell. Two examples of this come from IBM and its efforts to promote the Java programming language developed by Sun Microsystems, that was widely embraced by firms competing with Microsoft in web-based technologies. As Java become more widely adopted, IBM would generate

increased revenue from sales of its own hardware and supporting services, especially its consulting services which have become an increasingly important component of IBM's overall revenues.

IBM came to realize that it made sense to take other software projects into the open domain, as part of its overall strategy. IBM's first open source spinout came from a pre-production R&D project. In response to IBM's growing interest in Java, in early 1996 two IBM researchers began work on an experimental Java compiler, which they named "Jikes". They quickly developed a prototype that was more efficient than Sun's industry standard compiler. After customer requests for a better Java compiler, in December 1998 IBM announced the release of Jikes in open source form to allow external programmers to extend and improve the compiler. Since 2000 development has been led by non-IBM engineers.[ii] Jikes has been widely adopted, and is now bundled with several distributions of open source operating systems.

A second IBM spinout came with Java development tools. In 1996, IBM purchased a Canadian software company that created such tools for its WebSphere application server product. IBM released much of this technology in open source form when it founded the Eclipse project in 2001 and these efforts were further boosted by its acquisition of Rational (Brody, 2001). Other software companies involved in web application development, including Borland, Red Hat, SAP and SuSE, and well as hardware makers HP, Fujitsu and Intel joined the Java development tools effort. In 2004 the project became an independent non-profit corporation ("Eclipse Forms Independent Organization," 2004), although IBM engineers retained technical leadership of key projects. As an IBM executive later explained, "It is not that we are looking to make more money off the platform. It is just that we are looking to accelerate the adoption of Java and the building up of it for all of us" (Southwick, 2004). Very recently, IBM even donated

500 software patents to the open source community, while it continues to license other parts of its patent portfolio for significant revenues.

But despite this openness, BEA and Sun — IBM's two major rivals in Java applications servers — chose not to join IBM's coalition, instead promoting the rival Java Tools Community (Taft, 2003). During 2004 BEA also created a "Beehive" open source project to release key application libraries from its WebLogic product for use with other development systems; it also helped a third party development of a "Pollinate" library to link Beehive with Eclipse. Finally, in March 2005 BEA officially joined the Eclipse project.

These spinouts also differ in the ongoing participation of their respective firm sponsors. IBM continues to provide hundreds of programmer-years of software development for Eclipse but limited support for Jikes, while Netscape has cut all sponsorship ties to Mozilla. The conditions under which a "foundling" project can become immediately self-sufficient are likely to be much narrower than one which enjoys ongoing support from its original sponsor.

The spinout thus makes sense for technologies that either are not yet commercialized (as with Jikes), or that will eventually become commoditized and thus of limited commercial value (as many predicted for Java development tools). Both IBM and BEA donated internal innovations to create open source projects, which were intended to fuel adoption of the innovations. As with other organizations that sponsor open source projects, the benefits included:

- helping establish their technology as *de facto* standards, which, at a minimum, reduces the likelihood of having to re-implement their technology to conform to competing standards;
- attracting improvements and complements that make the technology more attractive;

- together, the innovation and complements enable the sale of related products (such as other components of WebSphere and WebLogic);
- generating mindshare and goodwill with the same audience that includes the potential customers for these related products; and
- Lowering or eliminating the ongoing costs of supporting projects, while providing customers of the project some possible source of ongoing support.

These motivations for open source spinouts go beyond those identified for one of the most-often cited firm sponsor of spinouts. Xerox Parc spun out technologies that no longer aligned with the company's strategy (Chesbrough and Rosenbloom, 2002), which is consistent with Netscape's exit through creation of the Mozilla project. However, for the Jikes, Eclipse and Beehive projects, the sponsoring firms spun out open source projects that were closely aligned with the firm's ongoing strategies. As West (2003) observed for other open source projects, relinquishing some level of control was essential to win adoption.

4.3 Selling Complements: Apache, KDE, Darwin

Many goods in computers and electronics fall into what Katz & Shapiro (1985) term the "hardware-software paradigm". As Teece (1986) notes, the base innovation ("hardware") requires an investment in producing complementary goods ("software") specialized for that innovation, in order to make the entire system useful. In many cases, these complements are more valuable than the core innovation. For example, makers of videogame consoles deliberately lose money or break even on the hardware so that they can make money from software royalties (Gallagher and Park, 2002).

In other cases, a system architecture will consist of various components. Some mature (or highly competitive) components may be highly commoditized, while other pieces are more

rapidly changing or otherwise difficult to imitate and thus offer opportunities for capturing economic value. Two open source examples are the IBM's WebSphere and Apple's Safari browser.

Customers access the WebSphere e-commerce software using standard web browsers, so IBM originally developed a proprietary httpd (web page) server. IBM later abandoned its server for the Apache httpd server, recognizing that it would be wasting resources trying to catch up to the better quality and larger market share enjoyed by Apache (West, 2003). Today, IBM engineers are involved in the ongoing Apache innovation, both for the httpd server and also related projects hosted by the Apache Software Foundation (Apache.org website).

Similarly, in 2002 Apple Computer decided to build a new web browser called Safari, to guarantee one would be available for buyers of its computer systems. The browser built upon libraries from the Konqueror open source web browser, which in turn were developed to support the KDE desktop interface for Linux users (Searls, 2003). The move paralleled Apple's earlier use of BSD Unix as a foundation for its OS X operating system, in which it created a new open source project (Darwin) to share all modifications of the BSD code (West, 2003). For both Safari and OS X, Apple used open source and contributed back its changes, but the company did not release the remainder of the proprietary code for its browser and OS, respectively (Brockmeier, 2003).

In the case of the Apache, Konqueror and Darwin open source projects, the firms adopting open source components had four common characteristics:

- there was pre-existing open source code being developed without the intervention of the focal firms;

- the "buy vs. build" decision to use external innovation was made easier because the code was "free"[iii];
- the firms were willing to contribute back to the existing projects on an ongoing basis, both to assure that the technology continued to meet their respective needs and to maintain absorptive capacity;
- the firms could continue to yield returns for internal innovation by combining the internal and external technologies to make a product offering that was not directly available through open source.

Another model for selling complements is the "dual license" strategy, where a firm develops code and releases it both as an open source project and a commercial product. Buyers who want free software get no support and restrictions on source code distribution in exchange for development feedback. Less price sensitive buyers (e.g., corporations) pay the sponsoring firm a license fee to receive full features and support (Välimäki, 2003; West & O'Mahony, 2005). However, the ongoing proprietary control of such sponsored projects mean that they have trouble attracting external innovation, and open source thus becomes a marketing technique to attract adopters and build network effects through a large community of adopters.

4.4 Donated Complements: Avalanche, PC Game "Mods"

In other cases, firms make their money off of the core innovation but seek donated labor for valuable complements.

For decades, I.T. companies have encouraged their users to collaborate and share user-developed software that filled in the gaps for their proprietary offerings. This has been particularly relevant for medium and large buyer organizations (companies, universities and

government) with large internal I.T. organizations. In the 1960s, IBM sponsored its SHARE user group, while in the 1970s Digital Equipment Corporation had its DECUS.

More recently, firms have indirectly or directly supported user collaboration that is coordinated using open source techniques. One example is the Avalanche Technology Cooperative, a Minneapolis-based nonprofit founded in 2001 to pool IT customizations developed by local enterprise IT users. This would allow companies to integrate disparate packages such as PeopleSoft and SAP that do not provide their own integration modules.[iv]

Another example of donated complements is the use of "mods" for PC video games. The PC gaming industry competes with lower priced dedicated gaming platforms such as Sony's PlayStation2 or Microsoft's Xbox. The commercial publishers of PC games thus have decided to exploit the one key advantage they have versus the consoles: the ability for PC users to update and modify their games. To do this, publishers release editing tools for their games to encourage user mods that create different environments, scenarios, or even total rebuilds of the game; the users then freely distribute these mods on the Internet. A few of the mods (such as Battle Grounds) are developed as open source, but most are developed as closed source.

While mods do not directly generate publisher revenues, the novelty of the mods extends the otherwise relatively short demand period for most computer games. Meanwhile, the mods keep the name of the game in front of consumers for additional months, while the publishers need years to prepare follow-on products. This external innovation keeps the product current without tying up internal innovation resources. In rare cases, the publisher serves the need identified by the mod by creating its own game or even buys the mod outright.

As with open source, a key issue for mods is motivating the contributors. The motivations parallel those for open source: direct utility, intrinsic reward or external signaling. Individuals (or

virtual teams) contribute mods because of their creative nature, love of either the computer game they modified or the milieu they recreated via their mod (Todd, 2004). Students are also frequent contributors, increasing their enjoyment of a favorite game (direct utility) as well as signaling their value to potential employers.

The computer game industry highlights three key ideas for attracting external innovation that similar to those for open source:

- *minimizing technical obstacles*. Contributors develop mods because they can build upon the publisher's proprietary innovation to make a compelling game experience. As with other software development platforms (such as operating systems or databases), third party developers are attracted by platform capabilities and the prompt availability of development tools.
- *creating an infrastructure* that encourages participation and collaboration. For open source, this is a project website and e-mail lists, but for mods this would be a distribution site that highlights the mods. Modern technologies make the cost of such infrastructure quite low.
- *recognition for contributors,* including added visibility for the most popular creators. For example, since 2002 Apple has given annual awards for the best use of open source related to its OS X operating system.

However, the mods also help address one problem that's very different from those of business-oriented source projects. As with other entertainment products, novelty-seeking consumers eventually grow bored with a PC game; by combining the core game engine with new externally generated game scenarios, the external innovation extends the life of the core (internally developed) innovation. Of course, this novelty comes at the potential expense of loss

of some control over the software product. For example, recently "modders" unlocked some features in the PC version of Grand Theft Auto: San Andreas (GTA:SA) that enabled a pornographic "mini-game." As a result of this activity, the rating on all versions of GTA: SA (not just the PC version) were changed to "Adults Only" thereby severely curbing the availability of the title.

5. DISCUSSION

5.1 Open Source as Open Innovation

How did firms in open source software efforts effectively exploit open innovation? We identified four strategies of open innovation in software that addressed the unique combination and exploitation of innovation from multiple sources. Table 5.4 cross references these four strategies with the key open innovation challenges we identified.

Despite the wide popularity of open source collaboration, software firms appear to embrace open innovation only when there is no alternative — specifically a broad dispersal of both production knowledge and market share that forces vertically integrated producers to admit that they no longer can "do it all." The use of open source by firms typically begins in ways that do not disrupt their fundamental business strategy (e.g. selling complements), or comes at a time when their existing strategy is so threatened that they are forced then to make drastic changes.

At the same time, firms faced (and often acknowledged) the risks of collaborating, risks that have not yet been fully realized. For example, encouraging users to develop complements could reduce the availability of vendors to achieve proprietary lock-in, an explicit long-term goal of the Avalanche project.

Especially significant is the role of open source in enabling pooled R&D, which has been seen in other industries but rarely in software. This scarcity is hardly surprising, given that up front R&D forms one of the major barriers to entry in the software industry (Arthur, 1996; Cusumano, 2004). Even for products for which software is only part of the value creation — such as computer systems — the shared R&D available through open source software such as Linux can significantly reduce the barrier to new entrants (West and Dedrick, 2001).

Interviews suggested that firms were willing to share software R&D — and thus eliminate potential differentiation for the component — in areas that were necessary prerequisites for selling a complete system (e.g. "infrastructure layer") but offered few opportunities for differentiation. Removing duplicative investment reduced costs for this type of commodity component, consistent with Miotti and Sachwald's (2003) finding that R&D collaboration with more direct competitors was associated with reducing costs.

Is such pooled R&D an example of open innovation? As with other aspects of open innovation, it depends on the ability of firms to capture value for their investment (Chesbrough, Chapter 1). So if the firms use open source to provide a common base, but have a way to sell complements, then investing in an open source project to provide pooled R&D is consistent with open innovation.

To what degree are the lessons of pooling open source R&D applicable to other industries? If cost savings are a major goal of the collaborative R&D, then we would presume that the transaction costs of creating and maintaining such collaboration would have to be limited to make collaboration more attractive than a go-it-alone strategy. Two factors tend to reduce these costs, at least for open source projects. First, such virtually dispersed, decentralized production takes advantage of software and communication technologies that facilitate joint software

development (West and Dedrick, 2001; Lerner and Tirole, 2002). Secondly, considerable effort has gone into developing mechanisms for coordinating and governing such efforts (O'Mahony, 2003; Shah, 2004; O'Mahony and West, 2005). While we would not suggest that such collaboration and governance advances are necessary for other forms of pooled open innovation, we predict that adapting such practices to other industries would make such pooled R&D more likely.

5.2 Where Open Source and Open Innovation Part Ways

For the firms and projects in our sample, we concluded that most firm involvement in open source fits the Chesbrough (2003a) definition of Open Innovation, in which firms both use a broad range of external sources for innovation and seek a broad range of commercialization alternatives for internal innovation. However, we would not mean to suggest that all open source software is an example of open innovation — or for that matter, that all open innovation in the IT industry relates to open source software.

In fact, some form of open innovation has become the norm in the computer industry for decades. For the reasons identified by Teece (1986), Shapiro and Varian (1999) and others, computer vendors have long relied upon third-party suppliers of complementary software products; this has been true of even the most proprietary of systems, such as Apple's Macintosh (West, 2005). However, what we now refer to as open innovation was extended by IBM's 1980 decision to source its PC CPU and operating system from Intel and Microsoft instead of its traditional vertical integration.

Examples outside the overlap of open source and open innovation include the following (Figure 5.1):

- *Open Source but not Open Innovation*. Open Source is only Open Innovation if it has a business model. There are tens of thousands of Open Source projects created and run for non-pecuniary motivations — such as the work done within Project GNU, which is motivated by a strong ideology (West & Dedrick, 2005). Also this category could be AOL's exit strategy with Mozilla, which (like many of the Xerox PARC spinoffs) reflected the failure of the sponsor to create a viable business model, leaving the foundling innovation abandoned to whoever is willing to nurture it.
- *Open Innovation but not Open Source*. The "Wintel" PC using Windows and Intel components represents a powerful embrace of what we now term Open Innovation. On the one hand, it lowered barriers to entry, enabling the rise of numerous makers of PC "clones" (Moschella, 1997). On the other hand, IBM's failure to appropriate adequate returns from its PC systems integration activities led to its 2004 exit from the PC industry.
- *Neither Open Source nor Open Innovation*. As noted above, the use of independent software vendors (ISVs) for external innovations is the norm in the computer industry. However, some firms are heading in the other direction, becoming increasingly integrated. For example, Microsoft has integrated downstream from operating systems into applications such as Windows, Money and SQL Server – decreasing the relative importance of third party application providers. Intuit is adding additional services (such as loans) to extend its Quicken financial management software. In video games, console makers supply both hardware and software, with Nintendo particularly dependent on its internally generated game franchises like Pokemon and Mario Brothers rather than ISV-supplied titles.

In a self-interested prediction, Grove (1996) argued that the IT industry had been irrevocably transformed by component-based systems integration model, in which component suppliers using horizontal specialization achieved insurmountable economies of scale over vertically integrated innovators. We believe that open source is also having a profound impact on IT value creation and capture, but it is too soon to say what effect open source open innovation will have upon proprietary alternatives.

5.3 Open Source Collaboration Within a Value Network

The chapters in Section III how open innovation both enables and builds upon inter-organizational collaboration. Such collaboration has been variously referred to as a network form of organization (Powell, 1990), value network (Christensen and Rosenbloom, 1995) or an ecosystem (Iansiti and Levien, 2004a). Open innovation in the IT industry — and particularly the commercial support of open source software — certainly would fit such a classification.

Due in large part to IBM's aforementioned shift from vertical integration to externally sourced components, the IT industry has moved to a separation of technical activities based on a modular subdivision of labor between firms that parallels the technical modularity of the overall systems architecture (Grove, 1996; Langlois, 2003b). This form of open innovation enables firms that provide a component for a complex system to use various appropriability mechanisms to capture a return from their portion of the system's value creation (West, Chapter 6). When open source software is used as part of a complex system, a firm still faces the fundamental issues of coordinating the systemic innovation, assuring overall value creation and capturing the firm's portion of that value. These issues are common to any open innovation value network, as noted by Maula, Keil and Salmenkaita (Chapter 12) and Vanhaverbeke and Cloodt, (Chapter 13). But the use of open source software changes this ecosystem management in at least two ways.

The task is made easier where open source software reflects the commoditization of some portion of the overall complex system. Because open source is (by definition) free,[v] the use of open source makes value capture less important for that component: pooled R&D to develop a shared component reflects an acknowledgement of the commoditization of that component. Value capture is still a factor if the firm merely shifts its value capture from selling software licenses to selling support services for that software.

However, economic coordination of the systemic innovation is made more difficult if participants are motivated not by pecuniary goals such as value capture, but instead some combination of intrinsic and extrinsic factors. This chapter has identified some of the ways that firms can motivate the supply of external innovations from individuals, while the work of O'Mahony (2003; O'Mahony and West, 2005) highlights the motivations of open source communities in their collaboration with firms. But such coordination poses difficulties, both in practice and also for our theories of open innovation.

One thing that is clear from studying open source innovation — both through direct observation and from prior accounts of the process (e.g. DiBona et al, 1999; O'Mahony, 2003; Shah, 2004) — is the distinct set of attitudes and norms that set open source software production apart from commercial IT innovation. Shared organizational culture is long-identified form of governance within organizations, particularly for creative, individualistic workers and when direct monitoring of performance is difficult (cf. Kunda, 1992). But little is known about how such culture is created and maintained across a network organization, particularly if that culture is used to facilitate the process of open innovation.

5.4 Limitations

There are key limitations to the generalizability of our findings. While we were informed by broader secondary data on the open source movement, the framework was constructed using inductive theory-building from a small number of cases, anchored to specific firms in U.S. IT industry. As such, they may not generalize to other firms, let alone other industries.

More seriously, there are fundamental questions about drawing conclusions regarding an emergent phenomenon: open source software as part of corporate open innovation strategies is still a comparatively recent phenomenon, and there are many key questions regarding the sustainability of this model.

The open source movement built on a confluence of ideology, professional norms and enthusiasm; some question the long-term sustainability of such motivations. Also, many projects have been created as challenges to an entrenched incumbent (e.g. Microsoft), and if such challenges are largely unsuccessful, vendor interest in sponsoring future open source efforts could wane. In addition, as measured by traditional profitability standards, many open source projects have had problems. For example, currently JBoss and MySQL appear to be unprofitable and Linux vendor Red Hat is reportedly operating at breakeven or slight profit levels (Lyons, 2005).

Also, open source has yet to fully resolve the IP issues of accepting donations from a wide community of unknown contributors, as reflected by SCO's legal attacks against Linux. While such potential infringement has been attributed to ignorance, others have suggested that infringing "stealth" IP could be deliberately donated to projects to sabotage their success (Cargill and Bolin, 2004).

Finally, many have accused open source software as being more about re-implementing and commoditizing prior technologies than creating new innovations. The adoption of Linux as a low-cost Unix is the best known example of such commoditization, but other examples include MySQL and OpenOffice. To rebut such criticisms, open source supporters point to the mainly Internet innovations that were first implemented using open source projects (such as X, BSD, sendmail or Apache) and to ongoing university-led research projects such as the Globus Alliance. Even without this, prior researchers have shown how radical innovations disrupt existing market structure if they merely deliver similar capabilities at significantly lower cost (Christensen, 1997; Leifer et al, 2000; O'Connor, Chapter 4); open source software certainly would qualify as a radical (or "disruptive") innovation under such definitions.

5.5 Future Research

There are many factors that enable open innovation in open source. Future research could consider whether these characteristics of open source are prerequisites for other forms of open innovation:

- *feasibility of virtual teams* as a way to organize innovation, enabling pooled R&D and other collaboration between organizations;
- *a culture of open innovation* throughout such teams that spans organizations, vanquishing both a "not invented here" attitude towards external innovation and a "crown jewels" attitude of controlling internal innovations;
- *modularization* of technologies and products, to allow the external production of components or complements;
- *formal IP mechanisms* that encourage collaboration;
- *economic prerequisites* for effective open source collaboration; and

- *abandonment* of open source projects: how and when do they terminate?

Our attempts to define open innovation uncovered questions beyond those specific to software. Two relate to the availability of external spillovers:

- *commercialization of public research*. Universities have gotten increasingly sophisticated about profiting from their research spillovers, a trend encouraged in the U.S. by the Bayh-Dole Act (Colyvas et al, 2002; Fabrizio, Chapter 7). Will this restrict the flow of external innovations or provide an ongoing incentive for greater supply?
- *increasing conflict over patents*. The increasing scope and commercial value of patents has spawned various concerns that patents will inhibit traditional closed innovation (e.g. Jaffe & Lerner, 2004); the threat to external spillovers is likely greater.

Other questions relate to potential patterns for leveraging external knowledge:

- *boundaries of the firm*. While firms are making increasing use of virtual teams, collaborative R&D consortia and other shared fora, the root cause is far from clear. Is this evidence that R&D is no longer necessary to internalize in firms? Or are these or merely examples of specific innovations that cannot be appropriated by firms, symptomatic of industry segments that have become commoditized and thus where R&D produces little competitive advantage?
- *role of process innovations*. Companies like Dell combine external product innovations with internal process innovations. Research on open innovation has focused on innovation to produce products, so would the process of open innovation be fundamentally different when it incorporates process innovations?

- *low R&D intensity firms*. Many firms have low R&D intensity, either due to size (e.g. small businesses) or industry characteristics (low tech). Are they pursuing "external innovation," "open innovation," or (as commonly assumed) "no innovation" strategies?

5.6 Conclusions

Open source software offers a significant example of how open innovation can transform an industry. While producers of complements and add-ons have occurred since the IBM 360, the rise of open source has increased attentions on alternate forms of organizing to exploit firms' IP. The use of external innovation is not a wholly new idea, however the activities of firms surrounding open source software highlights ways firms can reap returns by "giving away" their IP and related firm resources. The transformation appears hastened by the nature of the good, the available tools, and previous trends in the sector away from vertical integration.

Open source software provides a powerful example of how firms can manage a complex ecosystem to combine external and internal innovations, creating an architecture for the whole product solution that both creates and captures value. Some of the sociological and legal characteristics may seem particularistic to open source, particularly with the culture of "free software" (West and Dedrick, 2005). However, the conflicts around open source — between those who want to share value and others who want to capture value from shared innovation — anticipate parallel concerns in other industries, as will be discussed in the next three chapters (West, Chapter 6; Fabrizio, Chapter 7; Simcoe, Chapter 8).

6. REFERENCES

"Corporate Overview," Open Source Development Labs, April 1, 2004, URL: http://www.osdl.org/about_osdl/OSDL_overview_website.pdf

"Eclipse Forms Independent Organization," press release, February 2, 2004, URL: http://www.eclipse.org/org/index.html

"Netscape Navigator", Wikipedia, April 16, 2004, URL: http://en.wikipedia.org/wiki/Netscape_Navigator, accessed May 22, 2004.

Arthur, W. Brian, 1996. "Increasing Returns and the New World of Business," *Harvard Business Review,* 74/4: 100-109.

Bekkers, Rudi, Geert Duysters and Bart Verspagen, (2002) "Intellectual property rights, strategic technology agreements and market structure: the case of GSM." *Research Policy,* 31/7: 1141-1161.

Brandenburger, Adam M. and Nalebuff, Barry J. (1996) *Co-opetition,* New York: Doubleday.

Bresnahan, Tim and Yin, Pai-Ling (2004) Standard setting in browsers: technology and (real world) incomplete information. Standards and Public Policy conference, Federal Reserve Bank of Chicago, May 13 - 14.

Brockmeier, Joe. (2003) "Is Open Source Apple's Salvation?," NewsFactor Network, April 21, 2003, URL: http://www.newsfactor.com/perl/story/21318.html

Brody, Steve, "Interview: The Eclipse code donation," IBM developerWorks, 1 November 2001 URL: http://www.ibm.com/developerworks/linux/library/l-erick.html

Cargill, Carl & Bolin, Sherrie (2004) "The Changing Nature of Standards Organizations: Walden Pond has been Drained." Standards and Public Policy conference, Federal Reserve Bank of Chicago, May 13 - 14.

Chesbrough, Henry and Rosenbloom, Richard S. (2002) The role of the business model in capturing value form innovation: evidence from Xerox corporation's technology spin-off companies. *Industrial and Corporate Change,* 11/3: 529-555.

Chesbrough, Henry W. (2003a) *Open Innovation: The New Imperative for Creating and Profiting from Technology.* Boston, MA: Harvard Business School Press.

Chesbrough, Henry W. (2003b) The Era of Open Innovation. *Sloan Management Review*, 44/3: 35-41.

Christensen, Clayton M. 1997. *The Innovator's Dilemma: When new technologies cause great firms to fail.* Boston: Harvard Business School Press.

Christensen, Clayton M.; and Rosenbloom, Richard S. 1995. "Explaining the attacker's advantage: technological paradigms, organizational dynamics, and the value network," *Research Policy* 24/2: 233-257.

Cohen, Wesley M. & Levinthal, Daniel A. (1990) "Absorptive capacity: a new perspective on learning and innovation." *Administrative Science Quarterly*, 35/1: 128-152.

Colyvas, Jeannette, Michael Crow, Annetine Gelijns, Roberto Mazzoleni Richard R.Nelson, Nathan Rosenberg, Bhaven N.Sampat (2002) "How do university inventions get into practice?" *Management Science,* 48/1: 61–72.

Cusumano, Michael A. (2004). *The Business of Software.* New York: Free Press.

David, Paul A., Hall, Bronwyn H., and Toole, Andrew A. (2000) "Is public R&D a complement or substitute for private R&D? A review of the econometric evidence." *Research Policy*, 29/4-5: 497-529.

DiBona, Chris, Sam Ockman and Mark Stone, eds. 1999. *Open Sources: Voices from the Open Source Revolution,* Sebastopol, Calif.: O'Reilly

Eisenhardt, Kathleen M. 1989., "Building Theories from Case Study Research," *Academy of Management Review* 14/4: 532-550.

Gallagher, Scott. and Park, Seung Ho (2002) "Innovation and competition in standard-based industries: A Historical Analysis of the U.S. Home Video Game Market." *IEEE Transactions on Engineering Management*, 49/1: 67-82.

Glaser, Barney and Anselm Strauss, 1967, *The Discovery of Grounded Theory: Strategies of Qualitative Research*. London: Wiedenfeld and Nicholson.

Gomes, Lee. (2004) "Avalanche Project is Clearing the Path for Tech Cooperation." *Wall Street Journal*, April 12, p. B1.

Gonsalves, Antone and Coffee, Peter. (1998) "Jikes! More Open Source Code," *PC Week,* December 7, p. 6.

Grove, Andrew S. (1996). *Only the Paranoid Survive: How to Exploit the Crisis Points that Challenge Every Company and Career,* New York: Doubleday.

Hansen, Evan, "AOL lays off Netscape developers," CNET News.com, July 15, 2003, URL: http://news.com.com/2100-1032_3-1026078.html

Harris, Stanley G., Sutton, Robert I. 1986. "Functions of Parting Ceremonies in Dying Organizations," *Academy of Management Journal*. 29/1: 5-30.

Hars, Alexander and Ou, Shaosong (2002) "Working for free? Motivations for participating in open-source projects." *International Journal of Electronic Commerce,* 6/3: 25-39.

Healy, Kieran and Alan Schussman 2003. "The Ecology of Open-Source Software Development". *Working paper, Department of Sociology, University of Arizona,* January 29, http://opensource.mit.edu/papers/healyschussman.pdf.

Hertel, Guido, Niedner, Sven and Herrmann, Stefanie (2003) "Motivation of software developers in open source projects: an Internet-based survey of contributors to the Linux kernel." *Research Policy,* 32/7: 1159-1177.

Iansiti, Marco and Levien, Roy (2004a); *The keystone advantage: What the new dynamics of business ecosystems mean for strategy, innovation and sustainability*, Boston: Harvard Business School Press.

Jaffe, Adam B. and Lerner, Josh. (2004) *Innovation and Its Discontents: How Our Broken Patent System is Endangering Innovation and Progress, and What to Do About It.* Princeton, NJ: Princeton University Press.

Kaplan, Jerry (1996) *Startup: A Silicon Valley Adventure,* Paperback ed, New York: Penguin.

Katz, Michael L. and Shapiro, Carl. (1985) "Network externalities, competition, and compatibility." *American Economic Review,* 75/3: 424-440.

Kessler, Eric H., Bierly, Paul E., Gopalakrishnan, S Shanthi. (2000) "Internal vs. external learning in new product development: effects of speed, costs and competitive advantage." *R&D Management*, 30/3: 213-224.

Krueger, Charles W. (1992). "Software reuse," *ACM Computing Surveys,* 24/2: 132-183.

Kunda, Gideon. 1992. *Engineering culture: control and commitment in a high-tech corporation*. Philadelphia: Temple University Press.

Lakhani, Karim R. and von Hippel, Eric. (2003) How open source software works: "free" user-to-user assistance. *Research Policy,* 32/6: 923-943.

Langlois, Richard N. 2003b. "Modularity in technology and organization," *Journal of Economic Behavior and Organization,* 49/1: 19-37.

Lawler, Edward E. (1971) *Pay and Organizational Effectiveness: A Psychological View*, New York: McGraw-Hill.

Leifer, Richard, Christopher M. McDermott, Gina Colarelli O'Connor, Lois S. Peters, Mark P. Rice, Robert W. Veryzer, Mark Rice, (2000) *Radical innovation: how mature companies can outsmart upstarts,* Boston: Harvard Business School Press.

Lerner, Josh and Jean Tirole 2002. "Some Simple Economics of Open Source," *Journal of Industrial Economics* 50/2: 197-234.

Lieberman, Marvin B. and Montgomery, David B. (1998) "First mover (dis)advantages: retrospective and link with the resource-based view." *Strategic Management Journal*, 19/12: 1111-1125.

Lyons, Daniel. (2005) "Open Source Smackdown." *Forbes.com*. (June 15). http://www.forbes.com/2005/06/15/jboss-ibm-linux_cz_dl_0615jboss.html

Miotti, Luis and Frédérique Sachwald (2003) "Co-operative R&D: why and with whom?: An integrated framework of analysis" *Research Policy,* 32/8: 1481-1499.

Mockus, Audris, Fielding, Roy T. and James D. Herbsleb. 2002. "Two case studies of open source software development: Apache and Mozilla." *ACM Transactions on Software Engineering and Methodology*, 11/3: 309-346.

Moschella, David C., *Waves of power: dynamics of global technology leadership, 1964-2010,* New York: AMACOM, 1997.

Mowery, David C., Oxley, Joanne E., Silverman, Brian S. (1998). "Technological overlap and interfirm cooperation: implications for the resource-based view of the firm," *Research Policy,* 27/5: 507-523.

Nelson, Richard R. and Winter, Sidney G. (1982) *An Evolutionary Model of Economic Change.* Cambridge, Mass.: Harvard University Press.

O'Mahony, Siobhán. (2003) "Guarding the commons: how community managed software projects protect their work." *Research Policy*, 32/7: 1179-1198.

O'Mahony, Siobhán and West, Joel (2005). "Building a Participation Architecture in Open Source Communities," Harvard Business School working paper, October.

Ouchi , William G. and Michele Kremen Bolton (1988) "The Logic of Joint Research and Development" *California Management Review* 30/3: 9-34.

Powell, Walter W. (1990). "Neither Market Nor Hierarchy: Network Forms of Organization." In Barry M. Staw and Larry L. Cummings, eds., *Research in Organizational Behavior* 12. Greenwich, CT: JAI Press.

Rosen, Lawrence (2004). *Open Source Licensing: Software Freedom and Intellectual Property Law,* Upper Saddle River, NJ.

Sakakibara, Mariko (2001) "Cooperative research and development: who participates and in which industries do projects take place?" *Research Policy*, 30/7: 993-1018.

Sanchez, Ron and Mahoney, Joseph T. (1996) "Modularity, Flexibility and Knowledge Management in Organizational Design. "*Strategic Management Journal.*

Schumpeter, Joseph A. (1934), *The Theory of Economic Development,* Cambridge, Mass., Harvard University Press.

Searls, Doc "Surprise: Apple's New Browser Is a Sister to Konqueror," LinuxJournal.com, January 11, 2003, URL: http://www.linuxjournal.com/article.php?sid=6565

Shah, Sonali 2004. "Understanding the nature of participation and coordination in open and gated source software development communities". Proceedings of the Academy of Management Conference, New Orleans, August 9-11, B1-6.

Shapiro, Carl & Hal R. Varian (1999). *Information rules: a strategic guide to the network economy*. Boston, Mass.: Harvard Business School Press.

Southwick, Karen "Big Blue's Mr. Web services," March 17, 2004, CNET News.com, URL: http://news.com.com/2008-7345_3-5173667.html

Taft, Darryl K., "Sun Mulls Joining Java Eclipse Effort," *eWeek,* September 1, 2003, URL: http://www.eweek.com/article2/0,1759,1236123,00.asp

Teece, David (1986) "Profiting from technological innovation: implications for integration, collaboration, licensing and public policy." *Research Policy,* 15/6, 285-305.

Tennenhouse, David. (2003) "Innovation breeds success at Intel." *IEE Engineering Management,* 13/6: 44-47.

Todd, Brett. (2004) The whys of modding. *Computer Games*, No. 163, June, 38-40.

Välimäki, Mikko, (2003) "Dual Licensing in Open Source Software Industry," *Systemes d'Information et Management*, January, URL: http://opensource.mit.edu/papers/valimaki.pdf.

von Hippel, Eric (1988) *The Sources of Innovation*. New York: Oxford University Press.

West, Joel (2003) "How open is open enough? Melding proprietary and open source platform strategies." *Research Policy,* 32/7, 1259-1285.

West, Joel, 2005. "The fall of a Silicon Valley icon: Was Apple really Betamax redux?" in Richard A. Bettis, ed., *Strategy in Transition,* Oxford: Blackwell, pp. 274-301.

West, Joel and Jason Dedrick. (2001), "Open Source Standardization: The Rise of Linux in the Network Era," *Knowledge, Technology and Policy* 14/2: 88-112.

West, Joel and Jason Dedrick. (2005) "The Effect of Computerization Movements Upon Organizational Adoption of Open Source," Social Informatics Workshop: Extending the Contributions of Professor Rob Kling to the Analysis of Computerization Movements, Irvine, Calif., March 11, 2005. URL: http://www.crito.uci.edu/2/si/resources/westDedrick.pdf

West, Joel, and O'Mahony, Siobhan. (2005) "Contrasting Community Building in Sponsored and Community Founded Open Source Projects." *Proceedings of the 38th Annual Hawai'i International Conference on System Sciences,* Waikoloa, Hawaii.

7. FIGURES AND TABLES

Innovation Model	Management Challenges	Resulting Management Techniques
Proprietary (or internal or "closed")	1. Attracting "best & brightest" 2. Moving research results to development	1. Provide excellent compensation, resources, and freedom. 2. Provide dedicated development functions to exploit research and link it to market knowledge.
External	1. Exploring a wide range of sources for innovation. 2. Integrate external knowledge with firm resources & capabilities	1. Careful environmental scanning 2. Developing absorptive capacity, and/or using alliances, networks, and related consortia.
Open	1. **Motivating** the generation & contribution of external knowledge 2. **Incorporating** external sources with firm resources & capabilities 3. **Maximizing** the exploitation of diverse IP resources	1. Provide intrinsic rewards (e.g. recognition) and structure (instrumentality) for contributions. 2. As above. 3. Share or give away IP to maximize returns from entire innovation portfolio.

Table 5.1: Models of Innovation and Resulting Managerial Issues

Project	**Product Category**	**Approach**
Apache	web server	shared R&D
Darwin	operating system	selling complements
Berkeley DB	database	spin-in, then dual license
Eclipse	programming environment	spinout, then shared R&D
Jikes	Java compiler	spinout
Linux	operating system	shared R&D
Mozilla	web browser	spinout, then shared R&D
MySQL	database	dual license
OpenOffice	business productivity	selling complements
Sendmail	mail router	spin-in, then dual license

Table 5.2: Open source projects as examples of open innovation

Category	Companies	Motivation
Computer systems vendor	Dell Fujitsu **Hitachi** **HP** **IBM** **NEC** Sun	These vendors spent the late 1980s and 1990s fighting the "Unix wars" with mutually incompatible Unix implementations for their workstations and servers. In the late 1990s, they began shifting resources from their proprietary Unix implementations towards adapting and extended a shared implementation of Linux.
Telecommunications vendor	Alcatel Cisco Ericsson NEC Nokia NTT Toshiba	These vendors used Unix to run their switching systems but began shifting to Linux. As with systems vendors, interested in assuring that Linux evolved to work with their respective hardware and customers.
Microprocessor producer	AMD **Intel** Transmeta	Makers of Intel-compatible processors wanted to speed the shift of enterprise applications from proprietary RISC processors to their commodity processors.
Linux distributor (server and desktop)	Miracle Linux NEC Soft Novell Red Hat SuSE Turbolinux	Distributors have a clear interest both in free riding off the work of others in developing Linux, and making sure that the software met the specific needs of their customers.
Embedded Linux distributor	LynuxWorks MontaVista TimeSys Wind River	Similar to motivations of desktop and server Linux distributors, but need to support more heterogeneous customer needs for use with custom system configurations.
Linux support company	VA Software Linuxcare LynuxWorks	Without development capabilities, the firms both want to leverage the work of others and understand how it met customer needs.
Software developers	Computer Associates Trolltech	Want to make the operating system more reliable for running their specific applications and libraries.

Founding member in **bold**
Source: OSDL and company websites (as of May 2004)

Table 5.3: Members of the Open Source Development Labs

Open Source Strategy	***Maximizing Returns of Internal Innovation***	***Role of External Innovation***	***Motivating External Innovation***	***Challenges***
Pooled R&D	Participants jointly contribute to shared effort	Pooled contributions available to all	Ongoing institutions establish legitimacy and continuity	Coordinating and aligning shared interests
Spinouts	Seed non-commercial technology to support other goals	Supplants internal innovation as basis of ongoing innovation	Free access to valuable technology	Sustaining third party interest
Selling Complements	Target highest value part of whole product solution	External components provide basis for internal development	Firms coordinate ongoing supply of components	Maintaining differentiation as shared components add capabilities
Donated Complements	Provide an extensible platform for external contributors	Adding variety and novelty to established products	Recognition and other non-monetary rewards	Third parties can control the user experience

Table 5.4: Open source strategies as solutions to open innovation challenges

	Open Innovation	*not Open Innovation*
Open Source	Apple: Darwin BEA: Beehive IBM: Apache, Eclipse, Jikes OSDL	Project GNU
Not Open Source	PC makers: CPU, Windows Game Mods	Microsoft: applications† Intuit: Quicken†

† Increasing use of vertical integration, with some reliance on external innovations

Figure 5.1: Overlap of Open Source and Open Innovation

8. END NOTES

[i] Interview, Asa Dotzler, Mozilla.org, March 8, 2004.

[ii] Interview with David Shields, IBM Corporation, May 24 and June 10, 2004, as well as news coverage including Gonsalves and Coffee (1998).

[iii] Both Apache and BSD packages were open without restriction in the typology of West (2003), while KDE contained the compulsory sharing restrictions of the GPL.

[iv] Interview with Scott Lien and Andrew Black, Avalanche Technology Cooperative, March 16, 2004; see also Gomes (2004).

[v] The ongoing debate over open source as "free speech" versus "free beer" is beyond our scope; see, for example, West and Dedrick (2005).