# Challenges of open innovation: the paradox of firm investment in open-source software


## Joel West[1] and Scott Gallagher[2]

[1]College of Business, San José State University, One Washington Square, San José, CA 95192-0070, USA. Joel.West@sjsu.edu
[2]College of Business, James Madison University, Harrisonburg, VA 22807, USA. gallagsr@jmu.edu



**Open innovation is a powerful framework encompassing the generation, capture, and employment of intellectual property at the firm level. We identify three fundamental challenges for firms in applying the concept of open innovation: finding creative ways to exploit internal innovation, incorporating external innovation into internal development, and motivating outsiders to supply an ongoing stream of external innovations. This latter challenge involves a paradox, why would firms spend money on R&D efforts if the results of these efforts are available to rival firms? To explore these challenges, we examine the activity of firms in open-source software to support their innovation strategies. Firms involved in open-source software often make investments that will be shared with real and potential rivals. We identify four strategies firms employ – pooled R&D/product development, spinouts, selling complements and attracting donated complements – and discuss how they address the three key challenges of open innovation. We conclude with suggestions for how similar strategies may apply in other industries and offer some possible avenues for future research on open innovation.**


## 1. Introduction

What challenges does an open innovation approach present to managers? Interestingly, models of open innovation offer the promise that firms can achieve a greater return on their innovative activities and their intellectual property (IP) by loosening their control over both (Chesbrough, 2003a). Open innovation models stress the importance of using a broad range of knowledge 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 IP.

The open innovation paradigm is often contrasted to the traditional vertical integration or 'proprietary' model where internal R&D activities lead to products that are developed and distributed by the firm (Chandler, 1990). The way to manage this proprietary model was summed up by Harvard president James Bryant Conant as 'picking a man of genius, giving him money, and leaving him alone' (Conant, 2002). Of course, getting the ideas from the 'man of genius' was only half the challenge, the other half was to exploit those innovations. This exploitation is where the proprietary model frequently broke down. For while some IP that could not be internally commercialized was licensed to others, all too frequently it 'sat on a shelf' waiting either for internal development, its research proponents to leave the firm to develop it on their own, or

even more dangerously, for it to 'spillover' to other firms (Smith and Alexander, 1988; Chesbrough and Rosenbloom, 2002).

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 (cf. Cohen and Levinthal, 1990). Therefore, the open innovation paradigm goes beyond just utilizing external sources of innovation such as customers, rivals, and universities (e.g. von Hippel, 1988) and is as much a change in the use, management, and employment of IP as it is in the technical and research driven generation of IP.

As with other information goods, IP plays a crucial role in the software industry. A growing segment of the software industry is open-source software. Open innovation may be an especially applicable framework for examining how firms have been able to exploit the opportunities provided by open source. Over the last 20 years collaboration between firms, suppliers and customers has produced open-source products such as the Linux operating system, Firefox web browser, and the Apache web server. Here, we consider patterns of firm behavior towards open-source software as an exemplar for more general forms of open innovation.

Our research considers two questions. First, how do firms' use of open source correspond to theories of open innovation? Second, we consider the paradox: why would firms contribute resources, including IP, to projects that will benefit others, including their competitors? We are especially concerned with the strategies firms employ to help address three management challenges of open innovation – the maximization, incorporation, and motivation of IP – that we discuss in the next section.

## 2. The challenges of the open innovation paradigm

In contrast to earlier models and 'fully integrated innovators' like AT&T (now Lucent) Bell Labs and IBM which conduct 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, internal innovation is supplemented by systematic scanning for external knowledge (facilitated by firm investments in absorptive capacity) with firms maximizing the returns that accrue from both sources (Table 1). Such strategies require firms to realign innovation strategies to extend beyond the boundaries of the firm, while creating mechanisms for appropriating value from the combined innovation. Based on our definition of open innovation, in practice the integration of internal and external innovation entails three challenges (Figure 1):

- *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.

Table 1. Models of innovation and resulting managerial issues.

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

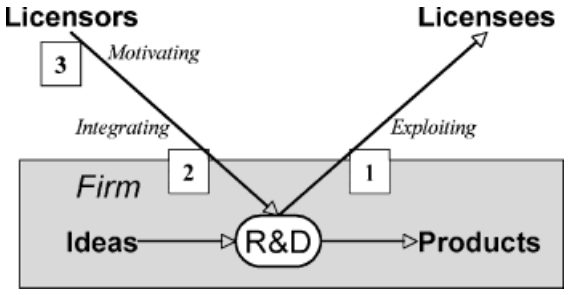


Figure 1. Motivating, integrating and exploiting innovation.

- *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. This requires scanning, absorptive capacity, and also the political willingness to incorporate external innovation.
- *Motivation*. Prior open innovation research assumes that external sources of innovation will arise. To date this has clearly been true. However, external sources of innovation are supplied by some person or entity. How can firms work to insure that this stream of external innovation is replenished? Why would firms contribute IP that was going to be made available to their rivals? This last issue is the heart of the 'paradox' of firm investments in open-source software.

## 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 proprietary model);
- building absorptive capacity and using that capacity to identify external innovations;
- generating innovations that generate returns through external commercialization (e.g. licensing patent portfolios); 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 firms may combine a variety of these approaches. For example, to identify promising technologies Intel establishes research labs near elite university research groups, with open flows of information in both directions. If an innovation proves promising, to internalize the innovation Intel recruits the top academic researchers to help commercialize it and facilitate its production (Tennenhouse, 2003). A cooperative example of multiple approaches is 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 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

To benefit from external innovations, organizations need to identify such innovations, maintain the absorptive capacity to understand them, and be able to combine such spillovers with firm-specific internal innovation to produce a product tailored to the firm's specific needs.

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 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, p. 156).

## 2.3. Motivating spillovers

With external innovation, there is often an unstated assumption that the sources of external innovations will continue to produce them. But what happens if everyone tries to be a 'free rider' by only absorbing external innovations? Will historic 'innovation benefactors' – such as government and non-profit research sponsors – continue to offer a plentiful supply (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.

Motivating individuals to generate and contribute their IP in the absence of financial returns is a management challenge for open innovation. One motivation model is expectancy theory, which posits

that individuals are motivated by a combination of valence (the intrinsic or extrinsic attractiveness of a reward) and instrumentality (the path to that reward) (Lawler, 1971). The proprietary innovation model solved this challenge though extrinsic compensation coupled with adherence to traditional scientific norms. The external model relies upon intrinsic factors or others, e.g. universities, to partially or wholly provide the motivation for creating IP.

The incentives for organizations to contribute spillovers fall into two categories. In the one case, the innovation benefits the innovator and nothing is reduced by sharing that benefit. 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 and Nalebuff, 1996, p. 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.

## 3. Research design

These three management challenges led to three related research questions:

- How do firms embrace open innovation approaches as part of their R&D efforts?
- Why would firms commit their IP and *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?

Because we are investigating relatively novel phenomena we focus on rich qualitative data to aid our theory building (Glaser and Strauss, 1967; Eisenhardt, 1989). Our research efforts included both primary and secondary sources. From 2002 to 2004, one author conducted 47 interviews with 41 informants representing 26 organizations, including 14 for-profit firms, spanning eight major open-source projects. These interviews focused on the strategies of these organizations for selling or participating in open-source software, and their motivations for doing so. Most interviews ranged from 45 to 90 min, and most were tape recorded for later consultation. This was supplemented by participation in five Silicon Valley industry conferences and seminars from August 2003 through November 2004 focused solely on open-source software. These primary data sources were complemented by a secondary data review of approximately 800 news articles from trade journals, business press and websites related to open-source topics.

## 4. Open source as open innovation

Open source and related collaborative development techniques in the software industry provide evidence of how the three key challenges of open innovation have been addressed by commercial firms. Open source can also 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. Open-source software is a great exemplar of open innovation because of the shared rights to use the resulting technology as well as the collaborative development of the technology.

### 4.1. Prior research on open-source software

'Open source' software includes source code that can be modified and redistributed to others, while acknowledging the original author's contribution (Perens, 1999; Raymond, 2004). The term encompasses a range of collaborative practices dating back to the 1970s, including university-based research on BSD Unix during the 1970s and the 'free software' movement launched by Richard Stallman in 1984 (McKusick, 1999; Stallman, 1999).[1]

Because 'open source' refers to a specific set of software licenses approved by the non-profit Open-source initiative, the term explicitly defines a particular subset of IP policies. However, it also often refers to a development methodology where geographically dispersed programmers collaborate to jointly produce software using virtual collaboration tools (West and O'Mahony, 2005). While the first programmers were hobbyists, they have been joined today by a number of professionals paid by employers that either intend to use the software internally, or to sell related products and services. Virtual development has been facilitated by widespread dissemination of tools and the availability of the Internet, which have also

enabled other forms of open innovation. One example is 'gated source,' in which open source processes are used within a firm's invitation-only group of software developers (Shah, 2003); another is PC game modifications (Scacchi, 2004).

What motivates individuals to contribute to open-source projects? Consistent with expectancy theory, empirical research (Hars and Ou, 2002; Lerner and Tirole, 2002; Hertel et al., 2003; Lakhani and von Hippel, 2003) found three classes of 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
- *signalling* one's capabilities to gain respect from one's peers or interest from prospective employers.

Meanwhile, firms have used hybrid strategies that combine the benefits of open-source software with some of the control of proprietary approaches (West, 2003).

Thus, open source as an open innovation strategy has two key components: shared rights to use the technology, and collaborative development of that technology using donated labor. From an analysis of major projects (Table 2), we identify four approaches for open innovation in open-source software – two driven by the structural relationship of contributor-participants, and two driven by the value proposition of a complex product.

## 4.2. Structural approaches to open innovation

One way to classify open innovation is through the structural relationship of the R&D contributors. To the traditional integrated approach of software development within the firm, open source offers two alternatives: pooled R&D and spinouts (Figure 2). These two approaches can be combined with each other, or with one of the product-oriented strategies given in Section 4.3.

### *4.2.1. Pooled R&D or product development: Linux, Mozilla*

A noted instance of open innovation is that of pooled R&D or product development (Chesbrough, 2003a). 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 industries with strong vertical relationships (Sakakibara, 2001) and in areas that are highly risky or for industries most dependent on advanced science (Miotti and Sachwald, 2003). Two highly visible open source examples are the Open Source Development Labs (OSDL), and the Mozilla project. In both, firms donate IP to the open-source project while exploiting the common benefits of all contributors to facilitate the sale of related products.

A great example of pooled R&D is the Mozilla web browser project that was created by Netscape in 1998, in response to competitive pressures from

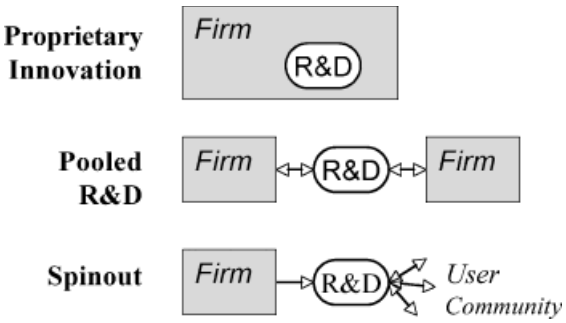


Figure 2. Knowledge flows in three software R&D models.

Table 2. Open source projects with commercial applications leading to open innovation.

| Project | Founded* | Founders | Product Category | Commercialization |
|---|---|---|---|---|
| Apache | 1995 | Eight webmasters | Web server | Shared R&D |
| Darwin | 1999 | Apple | Operating system | Selling complements |
| Eclipse | 2001 | IBM | Programming environment | Spinout |
| Jikes | 1998 | IBM | Java compiler | Spinout |
| Konqueror | 2000 | KDE project | Web browser | Selling complements |
| Linux | 1991 | L. Torvalds | Operating system | Shared R&D |
| Mozilla | 1998 | Netscape | Web browser | Spinout, shared R&D |
| MySQL | 1995 | M. Widenius and D. Axmark | Database | Selling complements |
| OpenOffice | 2000 | Sun | Business productivity | Selling complements |
| Sendmail | 1983 | UC Berkeley | Mail router | Selling complements |

*Date of open source project founding or source code release, whichever is earlier.

Microsoft's Internet Explorer. In July 2003, Netscape ended its sponsorship, deferring responsibility to the open-source community. Vendors such as IBM, HP, and Sun needed a Unix-based browser to help sell Internet-connected workstations, so each assigned software engineers to work with the Mozilla project, both to help keep the project moving forward and to assure that new releases were compatible with their respective systems (Dotzler, 2004). Today, this descendant of Netscape Navigator is available for a wide range of Unix systems, among others.

A more complex and structured example is Linux-related development at the OSDL. While Linux began in 1991, the OSDL was founded in 2000, attracting a wide range of Linux-related providers of hardware, software and services. In its first five years, the consortium worked on three projects: data center Linux, carrier grade Linux and desktop Linux.

How do these projects address the three open innovation challenges? For individuals and firms participating in Mozilla, the *quid pro quo* is straightforward: systems vendors maximize the returns of their innovation by concentrating on their own needs (e.g. platform-specific customization), and then distribute the shared browser technology with their integrated systems. For the OSDL, firms contribute specialized knowledge to build a common platform. OSDL resembles other self-supporting industrial research consortia, where firms pool interests towards a common goal, cooperating in supporting that goal and competing in selling their respective products. It is similar to other IT consortia in the wide range of motivations represented by the members (Table 3).

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

- *Spillovers are not controllable*. Many consortia attract members by limiting direct access to the consortium's research output to member-participants, reducing 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*. Technical contributions to these 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.

These two differences highlight how 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.

#### *4.2.2. Spinouts: Jikes, Eclipse, Beehive*

Open innovation can release the potential of IP within the firm that is not creating value. In some cases, the IP is no longer strategic, as when AOL Time Warner spun off Mozilla into a stand-alone open-source project (Hansen, 2003). But in addition to spin-off and abandonment, firms may also release more value from their technologies by situating them outside the firm, while at the same time maintaining an ongoing corporate involvement. Here, we use 'spinout' to refer to

Table 3. Members of the open source development labs.

| Category | Companies | Motivation |
|---|---|---|
| Computer systems vendor | Dell, Fujitsu, **Hitachi, HP, IBM, NEC,** Sun | Replacing proprietary Unix in computers with shared Linux |
| Telecommunications vendor | Alcatel, Cisco, Ericsson, NEC, Nokia, NTT, Toshiba | Replacing proprietary Unix with Linux in telecom equipment |
| Microprocessor producer | AMD, **Intel,** Transmeta | Enter Unix market using Linux |
| Linux distributor (server and desktop) | Miracle Linux, NEC Soft, Novell, Red Hat, SuSE, Turbolinux | Sell Linux distributions and services |
| Embedded Linux distributor | LynuxWorks, MontaVista, TimeSys, Wind River | Design Linux into custom products for customers |
| Linux support company | VA Software, Linuxcare, LynuxWorks | Sell Linux services |
| Software developers | Computer Associates, Trolltech | Adapt proprietary applications to Linux |

Founding member in **bold.** *Source*: 'OSDL Members,' OSDL and company websites (as of mid-2004).

all cases where firms transform internal development projects to externally visible open-source projects.

If a firm is essentially giving 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 (see Section 4.3.1). Two examples of this come from IBM and its efforts to promote the Java programming language developed by Sun Microsystems to compete with Microsoft. In a Java-centric world, IBM would still generate revenue from sales of hardware and supporting services.

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 released Jikes in open source form to allow external programmers to extend and improve it. IBM continues to host the project website, but since 2000 development has been led by non-IBM engineers (Gonsalves and Coffee, 1998; Shields, 2004).

A second IBM spinout came with Java development tools. IBM created such tools for its WebSphere application server product, and then released much of this technology in open source form when it founded the Eclipse project in 2001. Other software companies involved in web application development as well as rival hardware makers joined Eclipse. In 2004, the project became an independent non-profit corporation, 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).

However, Java rivals BEA and Sun chose not to join IBM's coalition, instead promoting the competing Java Tools Community. 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. By 2006, Eclipse had become the most vibrant open-source community controlled by its vendor members; the IBM spinout benefits both from IBM's initial contribution of technology and also the pooled R&D investments of the current vendor-members.

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, in order to fuel adoption of related products. As with other organizations that sponsor open-source projects, the benefits included:

- helping establish their technology as *de facto* standards, which reduces the likelihood of having to re-implement other products 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 (e.g. WebSphere and WebLogic); and
- generating mindshare and goodwill with the same audience that includes the potential customers for these related products.

These motivations for open-source spinouts are contrary to those of the oft-cited example of Xerox PARC, which spun out technologies that no longer aligned with Xerox's strategy (Chesbrough and Rosenbloom, 2002). Here, firms relinquished control of key technologies – precisely because they were strategically aligned, and giving up control was an effective strategy to win adoption.

## *4.3. Product-centric approaches*

Many innovations require a combination of goods and service to provide a 'whole product solution' to buyers (Moore 1991). In computers and electronics, they often fall into what Katz and Shapiro (1985) term the 'hardware–software paradigm.' As Teece (1986) notes, the base innovation ('hardware') requires an investment into producing complementary goods ('software') specialized for that innovation, in order to make the entire system useful.

In some cases, a system architecture will consist of various components – with mature components highly commoditized, while other pieces are more rapidly changing or otherwise difficult to imitate and thus offer opportunities for capturing economic value (West, 2003). In other cases, the complementary products are more valuable than the core innovation – as when videogame console producers deliberately lose money on the hard-

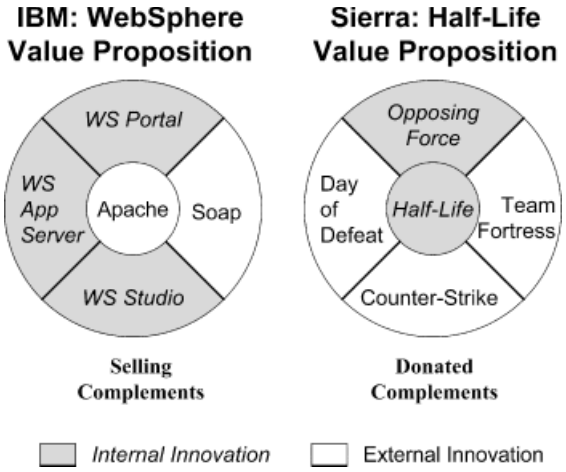


Figure 3. Open innovation to complete whole product solution.

ware so that they can make money from software royalties (Gallagher and Park, 2002).

Both approaches – selling complements or providing incentives to attract them – provide examples of how firms use open innovation to complete their whole product solution (Figure 3).

*4.3.1. Selling complements: Apache, KDE, Darwin*

In Teece's (1986) conception, essential complements to a product include not only other products, but services to buyers and (often invisible) activities within the producer's value chain. Linux distributors (such as Red Hat) that take freely available software and providing installation, training and support services would be selling complements to a free core product, as would the many other firms that sell services for 'free' software. Two open source examples of selling complements are IBM's WebSphere and Apple's Safari browser.

Customers access IBM's 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, rather than waste resources trying to catch up to the better quality and larger market share enjoyed by Apache (West, 2003). Today, IBM engineers participate in the ongoing Apache innovation, both for the httpd server and also related projects hosted by the Apache Software Foundation.

Similarly, in 2002 Apple Computer decided to build its own web browser to guarantee one would be available for its customers. The Safari browser built upon libraries from the Konqueror web browser developed for the KDE open-source desktop (Searls, 2003). Apple's move paralleled its OS X strategy, where it created a new open-source project (Darwin) to share its modifications of the BSD Unix code (West, 2003; Hawkins, 2004).[2] 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 versus build' decision to use external innovation was made easier because the code was 'free'[3];
- the firms were willing to contribute back to the existing projects on an ongoing basis, to assure that the technology continued to meet their respective needs, to maintain absorptive capacity, and to avoid discouraging external innovators; and
- 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.

A more sophisticated version of selling complements is the 'dual license' model, where firms such as MySQL or Trolltech create and sponsor an open-source project with their own software, and continue to provide development resources to develop and improve that software.[4] These firms use a price discrimination (or 'versioning') strategy, consistent with Shapiro and Varian (1999): 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 and O'Mahony, 2005). In response to the most successful dual license software, MySQL, from 2004 to 2006, Microsoft, Oracle, and IBM announced free entry-level versions of their database software for entry level and evaluation purposes.

*4.3.2. 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. This is nothing new, as such sharing dates to at least the 1950s, when IBM

encouraged the establishment of users' groups, with the hope that pooling software development would alleviate both a programmer shortage and high custom software costs. The SHARE user group was established in 1955, and one year later it estimated it saved its 60 members $1.5 million in in-house programming costs from one program alone (Campbell-Kelly, 2003, pp. 31–34).[5] A contemporary version of this is the Avalanche Technology Cooperative, founded in 2001 to pool IT customizations developed by enterprise IT users that would allow companies to integrate disparate packages such as PeopleSoft and SAP (Gomes, 2004; Lien and Black, 2004).

However, open-source complements appear less common than open-source core innovations. In interviews, vendors talk about the 'customer facing' portion of a system providing the best opportunity for visible differentiation, so such innovation is more likely to be proprietary.

Today, the PC game industry has a proven model, in which the game developer provides the core technology and some 'customer facing' complements, while encouraging users to develop their own complements, known as game modifications (aka 'mods'). To allow users to update and modify their games, publishers release editing tools for their games to encourage user mods; the users then freely distribute the mods on the Internet. Frequently, this is followed up with the release of the core game itself under and open-source license, as with Id Software's Quake.

A few of the mods are developed as open source, but most are not. While mods do not directly generate publisher revenues, the novelty of the mods extends the 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 publishers prepare follow-on products, keeping the product current without tying up internal innovation resources. Perhaps the most successful example comes from Sierra, which released the Half-life multi-player game as an 'engine,' and Opposing Force as an alternate scenario using that engine. It attracted several donated 'mods,' including Day of Defeat (which it later purchased), and 'Counter-Strike,' which sold an estimated $40 million worth of games (Keighley, 2002).

A hybrid model of game modifications and open source comes with the popular multiplayer online game Quake. Id Software released successive GPL-licensed versions of the core Quake simulation engine from 2001 to 2005, with two goals in mind. First, to enable the creation of modifications and other enhancements by the gamer community. Second, Id generates licensing revenue from commercial game developers who (under the dual license strategy) pay for the rights to distribute their commercial Quake-derived game.

As with open source, a key issue for mods is motivating contributors. The motivations parallel those for open source: direct utility, intrinsic reward or external signalling. 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 signalling their value to potential employers. While not pure open-source software, the computer game makers have adopted some specific practices to motivate IP contributions:

- *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, 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 is a site to highlights the mods. Modern technologies make the cost of such infrastructure quite low and accessible to global contributors around the world.
- *Recognition for contributors,* including added visibility for the most popular creators.

However, mods also address a problem very different from those of business-oriented open-source projects. As with other entertainment products, novelty-seeking consumers eventually grow bored with a PC game so by combining the core game engine with new externally generated game scenarios, the external innovation extends the life of the core (internally developed) innovation.

## 5. Discussion

Open-source software highlights many ways firms can enhance their competitive advantage by using the ideas of open innovation. However, the case of open source seems particularly paradoxical, in that a key part of the customer offering is inherently 'free.' Here, we have shown how firms

are able to create value (and revenue) from their IP over and above that which was given away 'free.'

## 5.1. Open innovation in IT industries

Firms have long faced concerns on how to obtain returns to innovation, particularly when they lack the resources to fully exploit or appropriate the returns (Teece, 1986). The open innovation framework helps explain how firms have used the rise of open-source software to develop new forms of innovation strategies. The use of open source by firms typically begins in ways that does not disrupt their fundamental business model (e.g. selling complements), or comes at a time when their existing business model is so threatened that they are forced them to make drastic changes. We identified four open innovation strategies software firms used in order to exploit internal and external innovation. Each of these four strategies addresses the three open innovation challenges (Table 4).

How did firms maximize the returns to their internal IP? Consistent with absorptive capacity arguments, in each case the firms were required to posses some level of IP either to enable them to target a high value portion of the market (selling complements) or to enable them to trade with other firms (pooled R&D) or lay the foundation for others to contribute to the project (Spinouts & Donated Complements). Effective open innovation does not eliminate the need for an internal stock of IP, but instead integrates it throughout the firm so additional opportunities can be identified and exploited.

How did firms integrate the external IP into their organization? Sometimes integration is explicitly planned for as it is under the pooled R&D strategy. However, firms face risks from collaborating, such as when the pooled R&D supporting a common Linux platform commoditized the existing Unix systems market and reduced barriers to entry (West and Dedrick, 2001).

What motivates external sources of IP? Occasionally firms can simply rely upon the dedication of novices or well-meaning individuals to make IP contributions (pooled R&D, donated complements). For example, by pooling changes from its Darwin project with other open-source projects – such as FreeBSD, NetBSD, and OpenBSD – Apple both contributes and receives IP as part of a pooled R&D strategy. In other cases, commercialization often requires either a significant up front contribution (spinouts) or a more ongoing level of support and coordination of the efforts (selling complements) by firms.

Consistent with the Almeida et al. (2003) finding that larger firms are more likely to build on external knowledge, in our study the large IT firms with a broad scope of products became involved in open source because they could not ignore any significant source of external innovation available to rivals. Among smaller firms, some aligned their innovation strategies with open source, while others sought niches unaffected by open-source competition.

## 5.2. Implications for other industries

The four patterns of combining internal and external innovation in open source could be applied to more general forms of open innovation:

- *Pooled R&D*. As with other consortia, firms leveraging open source often need to change corporate culture to realize the benefits of shared R&D. An open culture is essential to accept external innovations, overcome 'not

Table 4. Open source strategies as solutions to open innovation challenges.

| Open source strategy | Example | Maximizing returns of internal innovation | Role of external innovation | Motivating external innovation |
|---|---|---|---|---|
| Pooled R&D/ product development | Linux | Participants jointly contribute to shared effort | Pooled contributions available to all | Ongoing institutions establish legitimacy and continuity |
| Spinouts | Eclipse | Seed non-commercial technology to support other goals | Supplants internal innovation as basis of ongoing innovation | Free access to valuable technology |
| Selling complements | Apache | Target highest value part of whole product solution | External components provide basis for internal development | Firms coordinate ongoing supply of components |
| Donated complements | Half-Life | Provide an extensible platform for external contributors | Adding variety and novelty to established products | Recognition and other non-monetary rewards |

invented here' biases and build trust between firms (Nakamura et al., 1997; Santoro and Chakrabarti, 2001; Chesbrough, 2003a). For example, Novell acquired Ximian, an open-source startup, to transform its internal culture to become more outwardly focused to work better with external open-source projects (Freedman, 2004).

- *Spinouts*. As spinouts are valuable for technologies locked in the laboratory, they are most relevant to the largest firms, which both have the largest innovation budgets and also the largest bureaucracies to defeat commercialization. While Xerox PARC has exemplified such obstacles, in some cases Xerox spun out the technology and participated financially in its commercialization (Chesbrough and Rosenbloom, 2002).
- *Selling complements*. For industry segments where firms previously succeeded through innovation-based product differentiation, firms face a choice of accepting commoditization or (consistent with Henderson and Clark, 1990) developing architectural innovation capabilities to develop differentiated products using commodity components.
- *Donated complements*. These match what von Hippel and Katz (2002) call 'user toolkits,' where general purpose technologies are sold to users capable of generating their own modifications and improvements. Such strategies are most feasible when selling to technically proficient buyers, whether corporate engineers or hobbyist programmers.

A key problem for open innovation is that firms integrating internal and external innovations can face higher coordination costs and risks than if all activities were internalized; the firms in our sample relying on open-source external innovations faced both these costs and risks. Of the two types of open-source projects identified by West and O'Mahony (2005), the firm-sponsored projects forced the sponsoring firm to bear the preponderance of coordination costs; for community-led projects, firms faced lower coordination costs but higher potential risks.

Open-source software as part of corporate open innovation strategies is still a comparatively recent phenomenon, and there are some unresolved issues. Open source built on a confluence of ideology, professional norms and enthusiasm, which may or may not be sustained as it becomes more commercialized. 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.

A final challenge for managers that may arise in other industries that open source has yet to resolve are IP issues of accepting donations from a wide community of unknown contributors. What happens if a user inadvertently contributes proprietary IP to an open-source project? Such a case is exemplified by SCO's lawsuit accusing Linux contributors of stealing copyrighted source code from SCO's proprietary Unix implementation. Others have suggested that proprietary 'stealth' IP could be deliberately donated to open-source projects to sabotage those projects (Cargill and Bolin, 2004).

## 5.3. Further research

This analysis of open-source software provides further evidence that open innovation provides opportunities for firms to concentrate their R&D efforts on a small fraction of the 'whole product' solution. At the same time, it raises other questions about defining what is common to the practice of open innovation.

There is a huge gap between free-riding on basic research (e.g. Chesbrough, 2003b) and this study of the deliberate partitioning of software development between firms and open-source projects. What do these two open innovation extremes have in common? Examples might include virtual teams, cultural openness, technological modularization and public/private collaboration. At the same time, what other options are there between relying on free spillovers and coordinating a complex production ecosystem?

We obviously believe that there are useful lessons for all industries from the four open innovation strategies we've suggested here. However, there remain important considerations for attempting to take open innovation to other industries. For example, many firms still profitably engage in internal R&D and use the proprietary R&D model. Does this happen only in cases where (as Teece, 1986 predicts) firms can appropriate the returns from their innovations? There also remain importance considerations around process innovations. Is open innovation as powerful a framework for process as it is product innovative activity?

Other concerns center around our question of motivating innovations. University research spawned key open-source projects such as BSD, Python and sendmail. Recently, universities have increasingly sought to profit from their research

spillovers, a trend encouraged in the United States by the Bayh-Dole Act (Colyvas et al., 2002). This might restrict the flow of external innovations; conversely, it could increase the incentive for an ongoing supply of them, albeit at a higher cost for open innovators. Simultaneously, increasing conflict over patents may be an issue for firms as patent litigation severely affects those without defensive patents (e.g. Jaffe and Lerner, 2004). This raises an interesting issue. Do firms require a portfolio of legally protected IP, i.e. patents, in addition to absorptive capacity in order to exploit (or continue to exploit) open innovation?

Finally, the software industry reflects an industry with a high percentage of revenues spent on development (if not research). However, many firms in other industries have a lower R&D intensity and lower rate of internal innovation, either due to firm characteristics (lack of scale economies) or industry characteristics (low technology industries). Are such firms pursuing 'external innovation,' 'open innovation,' or (as commonly assumed) 'no innovation' strategies?

## References


Almeida, P., Dokko, G. and Rosenkopf, L. (2003) Startup size and the mechanisms of external learning: increasing opportunity and decreasing ability? *Research Policy*, **32**, 2, 301–315.

Bekkers, R., Duysters, G. and Verspagen, B. (2002) Intellectual property rights, strategic technology agreements and market structure: the case of GSM. *Research Policy*, **31**, 7, 1141–1161.

Brandenburger, A.M. and Nalebuff, B.J. (1996) *Co-opetition*. New York, NY: Doubleday.

Brockmeier, J. (2003) Is open source apple's salvation? NewsFactor Network, 21 April, http://www.newsfactor.com/perl/story/21318.html.

Campbell-Kelly, M. (2003) *From Airline Reservations to Sonic the Hedgehog: a History of the Software Industry*. Cambridge, MA: MIT Press.

Cargill, C. and Bolin, S. (2004) Standardization: a failing paradigm. Standards and Public Policy Conference, Chicago, IL, http://www.chicagofed.org/news_and_conferences/conferences_and_events/files/cargill.pdf

Chandler, A.D. (1990) *Scale and Scope*. Cambridge, MA: Belknap.

Chesbrough, H. and Rosenbloom, R.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, H.W. (2003a) *Open Innovation*. Boston, MA: Harvard University Press.

Chesbrough, H.W. (2003b) The Era of Open Innovation. *Sloan Management Review*, **44**, 3, 35–41.

Cohen, W.M. and Levinthal, D.A. (1990) Absorptive capacity: a new perspective on learning and innovation. *Administrative Science Quarterly*, **35**, 1, 128–152.

Colyvas, J., Crow, M., Gelijns, A., Mazzoleni, R., Nelson, R.R., Rosenberg, N. and Sampat, B.N. (2002) How do university inventions get into practice? *Management Science*, **48**, 1, 61–72.

Conant, J. (2002) *Tuxedo Park*. New York, NY: Simon & Schuster.

Dotzler, A. (2004) Interview, Mozilla.org, March 8.

Eisenhardt, K.M. (1989) Building theories from case study research. *Academy of Management Review*, **14** , 4, 532–550.

Freedman, D.H. (2004) Sellout or savior? *Technology Review*, **107**, 7, 44–49.

Gallagher, S. and Park, Seung H. (2002) Innovation and competition in standard-based industries: a historical analysis of the US home video game market. *IEEE Transactions on Engineering Management*, **49**, 1, 67–82.

Glaser, B. and Strauss, A. (1967) *The Discovery of Grounded Theory: Strategies of Qualitative Research*. London: Wiedenfeld and Nicholson.

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

Gonsalves, A. and Coffee, P. (1998) Jikes! More open source code. *PC Week*, **15**, 49, 6.

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

Hars, A. and Ou, S. (2002) Working for free? Motivations for participating in open-source projects. *International Journal of Electronic Commerce*, **6**, 3, 25–39.

Hawkins, R.E. (2004) The economics of the open source software for a competitive firm. Why give it away for free? *Netnomics*, **6**, 2, 103–117.

Henderson, R.M. and Clark, K.B. (1990) Architectural innovation: the reconfiguration of existing product technologies and the failure of established firms. *Administrative Science Quarterly*, **35**, 1, 9–30.

Hertel, G., Niedner, S. and Herrmann, S. (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.

Jaffe, A.B. and Lerner, J. (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, J. (1996) *Startup: A Silicon Valley Adventure*. Paperback edn. New York, NY: Penguin.

Katz, M.L. and Shapiro, C. (1985) Network externalities, competition, and compatibility. *American Economic Review*, **75**, 3, 424–440.

Keighley, G. (2002) Game Development à la Mod, Business 2.0, October, 66.

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

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

Lerner, J. and Tirole, J. (2002) Some Simple Economics of Open Source. *Journal of Industrial Economics*, **50**, 2, 197–234.

Lien, S. and Black, A. (2004) Personal Interview, Avalanche Technology Cooperative, 16 March.

McKusick, K. (1999) Twenty years of Berkeley Unix. In DiBona, C., Ockman, S. and Stone, M. (eds), *Open Sources: Voices from the Open Source Revolution*. Sebastopol, CA: O'Reilly, pp. 31–46.

Miotti, L. and Frédérique, S. (2003) Co-operative R&D: why and with whom?: an integrated framework of analysis. *Research Policy*, **32**, 8, 1481–1499.

Moore, G.A. (1991) *Crossing the Chasm*. New York, NY: Harper Business.

Nakamura, M., Vertinsky, I. and Zietsma, C. (1997) Does culture matter in Inter-Firm cooperation? research consortia in Japan and the USA. *Managerial and Decision Economics*, **18**, 2, 153–175.

Ouchi, W.G. and Bolton, M.K. (1988) The logic of joint research and development. *California Management Review*, **30**, 3, 9–34.

Perens, B. (1999) The open source definition. In DiBona, C., Ockman, S. and Stone, M. (eds), *Open Sources: Voices from the Open Source Revolution*. Sebastopol, CA: O'Reilly, pp. 171–188.

Prabhakar, E. (2005) Personal Interview, Apple Computer, 21 July.

Raymond, E. (2004) The Open Source Definition, Version 1.9, http://www.opensource.org/docs/definition.php

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

Santoro, M.D. and Chakrabarti, A.K. (2001) Corporate strategic objectives for establishing relationships with university research centers. *IEEE Transactions on Engineering Management*, **48**, 2, 157–163.

Scacchi, W. (2004) Free and open source development practices in the game community. *IEEE Software*, **21**, 1, 59–66.

Searls, D. (2003) Surprise: Apple's new browser is a sister to Konqueror. LinuxJournal.com, 11 January, http://www.linuxjournal.com/article.php?sid = 6565

Shah, S. (2003) Understanding the Nature of Participation and Coordination in Open and Gated Source Software Development Communities. MIT Free/Open Source working papers, April, http://opensource.mit.edu/papers/shah3.pdf

Shapiro, C. and Varian, H.R. (1999) *Information Rules: Strategic Guide to the Network Economy*. Boston, MA: Harvard Business School Press.

Shields, D. (2004) Personal Interview, IBM Corporation, 24 May.

Smith, D.K. and Alexander, R.C. (1988) *Fumbling the Future*. New York, NY: William Morrow.

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

Stallman, R. (1999) *The GNU operating system and the free software movement*. In DiBona, C., Ockman, S. and Stone, M. (eds), *Open Sources: Voices from the Open Source Revolution*. Sebastopol, CA: O'Reilly, pp. 53–70.

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

Tennenhouse, D. (2003) Innovation breeds success at Intel. *IEE Engineering Management*, **13**, 6, 44–47.

Todd, B. (2004) The Whys of Modding. Computer Games, 163, June, pp. 38–40.

Välimäki, M. (2003) Dual Licensing in open source software industry. *Systemes d'Information et Management*, **8**, 1, 63–75.

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

von Hippel, E. and Katz, R. (2002) Shifting innovation to users via toolkits. *Management Science*, **48**, 7, 821–834.

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

West, J. and Dedrick, J. (2001) Open source standardization: the rise of Linux in the network era, knowledge. *Technology and Policy*, **14**, 2, 88–112.

West, J. and O'Mahony, S. (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.

## Notes

1. However, the 'free' software contains IP restrictions intended to force sharing of any derivative works, while other forms of 'open' software (such as the Apache license) allow private commercialization of related innovations (West 2003).
2. After creating the Darwin open source project, Apple found the administrative overhead of allowing direct contributions and bug reporting was too great. However, the company monitors changes in the external OpenDarwin.org project, as well as in independent BSD projects such as FreeBSD (Prabhakar, 2005).
3. 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.
4. The third dual-license example cited by Välimäki (2003), Sleepycat Software, was purchased in February 2006 by Oracle, MySQL's largest proprietary competitor.
5. Distributing software as 'public domain' (no copyright asserted) explicitly qualifies as an open source-compatible license under rules of the Open Source Initiative.