Opening Windows: Open Source Software and the Future of Business*

Aaron C.T. Smith
La Trobe University
Australia

Opening Windows: Open Source Software and the Alternative Future of Business

In a business world increasingly dependent upon integrated technology, rapid innovation and global competition, open–source software is shaping up as one of the most significant drivers of change entering the new millennium. Open-source software is characterised by the public availability of its underpinning source code – the units that software is written in. Thus, in stark contrast with the highly protected codes of commercial or proprietary software, open-source software can be accessed and used by anyone at no cost. Its power is not just to be found in its availability, however. What makes open-source software a future-maker are the virtual communities that embrace it, downloading source codes from the Internet, adapting, improving and uploading them again for others to utilise. The result of this process is innovation with a speed and utility wildly faster than commercial companies can match. (von Krogh 2003) From Linux, the operating system named after its chief developer, at the time a Finnish university student, to the Apache web-server software that is the backbone of the Internet, in some areas of application open-source software is more prevalent than its commercial equivalent. Not only has open-source code encouraged radically faster developmental times in software, more customised functionality and significantly decreased costs of business, it may bring about the end of Microsoft's monopoly control of software. Indeed, one of the most hotly debated issues in information technology is whether Microsoft will eventually surrender its source code. Open-source is responsible for some of the biggest advances in computing: "...advances that are significantly shaping our economy and our future," (O'Reilly 1999: 34) particularly as it has already captured around half of the market for Internet servers and promises much for converting home and office users as well. In this essay I present five arguments supporting the contention that open-source software is more than just a key trend, but rather will play a role in shaping future commerce. Each of these five points represents a switching variable. Building upon them, I conclude with five alternative scenarios and comment on their second order impact.

For many, open-source software is a relatively recent phenomenon that only gained momentum in 1998 when Netscape released its newest Internet browser Mozilla, as an open-source product, and when IBM adopted Apache as its chief web server software (O'Reilly 1999). However, in the 1960s and 1970s, when software development was chiefly the purview of academics and engineers in corporate labs, it was perfectly normal to make source codes available to anyone interested enough to ask. Codes were swapped, modified and built on as modules in new software. The forerunner of today's Internet, ARPAnet, was used to facilitate this process. In the 1980s, however, one of the key sites

* I am indebted to Mark Humphries for his technical advice and insight toward the development of this essay.

for the development of software, MIT's Artificial Intelligence Laboratory, clamped down on what they viewed as their proprietary intellectual property. With this software licensed to a commercial developer, its source code unavailable to the suite of programmers who had contributed to its evolution, one of the MIT programmers jumped ship. This programmer, Richard Stallman, reacted to the new regime by founding the Free Software Foundation. His intention was to provide a legal platform for the "free" sharing of software source code. Although the idea was not immediately popular, Stallman's efforts are generally seen as the re-birth of the open code movement. It is a movement that is changing the world, beginning with an immense potential to decrease information technology (IT) costs for business.

My first point is intuitive in the sense that any opportunity for decreasing costs in business is likely to be explored. "Free" software, however, in the case of open-source software means much more than cost. While open-source software can often be accessed at no cost, the "free" philosophy is more about opportunity than price. Stallman (1999) defined software as "free" if, for any given user: 1) they have the freedom to run the program for any purpose; 2) they have the freedom to modify the program to suit their needs, an opportunity that requires access to the source codes; 3) they have the freedom to re-distribute copies, either gratis or for a fee; 4) and, they have the freedom to distribute modified versions of the program. These conditions form the philosophical basis for the open-source software community, and do not preclude profit-making. To that end, there are three commonly noted opportunities for companies to save or make money with open-source software. (Ljungberg 2000; Von Krogh 2003) First, a company may choose to employ open-source software in their own operations. Secondly, a company can seek a profit collating and distributing open source software, or by adding value to it with additional proprietary products. Red Hat and SuSE, which distribute Linux are good examples of this approach. Finally, companies may sell hardware which uses or is packed with open source software, such as IBM. This unprecedented freedom will drive change.

The leveraging power of free software, as defined by Stallman (1999), holds some inherent advantages for businesses both now and in the future. Companies no longer have to pay for Windows or other proprietary software licenses, as most conventional software now has open-source equivalents with comparable or superior functionality. For large companies this constitutes a significant savings. Even with the requisite budgets, the prospect of managing licenses associated with proprietary software is onerous. This will no longer be a necessity. In addition, file formats are ubiquitous and not under the exclusive control of a single software provider, which might increase its licensing fees or go out of business altogether. Following on from this, when something does go wrong there are options for its resolution. Open-source software is sufficiently well-known that there are plenty of consultants who can offer support. The ongoing development of open-source software means that its anomalies are more readily discovered, remedied, and the resulting up-grade made available for anyone to benefit from.

Dozens of Internet sites (eg. GNU.org, Php.com, MySQL.com) offer the opportunity to access open-source software and participate in the discussion and comparison of modifications that occur. If a visitor to the site is so inclined, they can download a suite of software for their personal use. Equally, the General Public Licenses associated with open-source software (sometimes referred to as a copyleft as a play on the word copyright) allow anyone including commercial businesses to use and distribute copies as well. This has had serious implications for businesses. For example, the MySQL database is the most popular in the world, with up to ten million installations estimated to have been undertaken from the site. For businesses previously dependent upon expensive commercial software licenses and support services, the opportunity to access free alternatives represents significant savings. Thousands of large companies have decided to, at least in part, venture down this path, including Google, NASA and Suzuki. In addition, some pivotal players in the information technology market such as Sun Microsystems, IBM and Oracle, are explicitly supporting open-source software.

The biggest conundrum associated with the extraordinary success of the open source code movement is associated with programmers' motivations for making contributions. As von Krogh (2003) pointed out, individuals and firms innovate because there are private incentives to do so. But where is the economic incentive for highly skilled programmers to contribute? The likelihood is that programmers involved in the open-source movement have much to gain, including experience and feedback in developing complex software, the cultivation of personal reputation, fun and a sense of belonging to a virtual community. (von Hippel & von Krogh 2003) It may also be that many of these programmers have no great love for Microsoft, and are genuinely pleased to be involved in offering an alternative that diminishes its power. This leads to my second point: open-source software will encourage changes in the power structure of the IT sector in the future. In turn, this will necessitate changes to the structure and operations of IT divisions, departments and support services in companies. For example, less time will be spent liasing with suppliers, installing upgrades and patches, and managing licenses. More time will be required to customize free software to the specific needs of the company and its divisions. IT support services will have to be capable of making modifications rather than seeking to get around the in-built functionality of commercial software.

With companies forced to make their own adjustments to software, the fragmentation of software versions seems inevitable. Hence, my third argument: It is difficult to imagine that commercial businesses imbued with a competitive instinct will surrender to feelings of community and sharing. As Glass (2004) counselled, open-source software fails most frequently because of the limited economic motivations for programmers, and political splintering. I would add to this the disincentive for companies to contribute their insights to public forums even if they remain comfortable with taking the contributions of others. The philosophical platform of the open-source movement seems inconsistent with that of the corporate environment. While this has been overcome with vigour over the last decade and a half, I suspect this might have something to do with a common adversary in Microsoft. At present the equation is simple in that programmers tend to either support or undermine Microsoft. If indeed Microsoft's effective monopoly is compromised, support for the range of commercial alternatives might be divided. Nevertheless, the power of the Internet for facilitating open-source communities has proven robust. At present it does not require the cooperation or commitment of the corporate sector in order to flourish, and this does not appear to be changing as the disseminative clout of the Internet continues to grow.

My fourth reason for suggesting that the open-source software movement requires more than just a passing interest from businesses concerned with foresight, is associated with the way in which open-source virtual communities work. The process does not end with the downloading and use of the open-source software. Anyone using the software can make contributions to the evolution of the source code, sometimes by providing "patches" to fix "bugs" or to add new modules and functionality. Users may make some modifications in order to add some customisation for their needs, and subsequently provide these source changes to the general community. Programmers, seeking to hone their skills, or simply expose their abilities, may also make contributions. Alternatively, interested parties may make suggestions, offer feedback, seek advice or suggest new projects through Internet site discussion forums. The great power of the open-source philosophy is manifested with the exponential growth of participants. In a short time, an immense amount of development can be achieved.

The open-source community is founded on virtual networking via the Internet, the outcome of which is software that in many ways outperforms commercial alternatives. Open-source groups are loosely-coupled communities that are completely distributed, self-selecting, delegated and emergent. (Ljungberg 2004) Ljungberg claimed that the open source movement is one key to the understanding of future forms of organizations, information work and business. He argued that these communities are a strange mixture of different organizational forms; there is hierarchy, but at the same time there is a rampant marketplace at work. Similarly, Awazu and Desouza (2004) argued that the future of knowledge work can be sighted through an understanding of how open-source communities function. Their goal was to entice thinking about how what they called "closed knowledge management agendas" that presently preside in organizations might be transformed into structures resembling open-source community environments. It is my view that the bottom-up properties of emergence that characterize the evolution of open-source software is its most vital ingredient. Irrespective of their size and resources, companies like Microsoft that offer proprietary software, cannot match this emergent development in terms of speed to market or responsiveness to the needs of users. One study of a mid-sized software developer, for example, indicated that only thirty people had the authority to make changes to the official version of the software (von Krogh, Spaeth & Lakhani 2003), whereas in open-source software communities, activity is directed by interest and self-selection, the product of which is available for anyone to test and evaluate. The limitation is that the program grows in its own way, without a specific direction, goal or leadership.

Holtgrewe (2004) found that open-source communities epitomize a complex acceleration of technological development, facilitated by the information exchange capabilities of the Internet and the open-source philosophy of shared intellectual property. For them, the accessibility of technology and knowledge are preconditions for future market creativity. According to von Hippel (2001), open-source software projects have led to innovation, consumption and development communities run completely by and for users. My fifth point is therefore that virtual open-source software communities are relevant to future business as an exemplar of organization and management of innovation, where bottom-up contributions are at least as important as the top-down decision-making of proprietary software companies.

Scenarios

To repeat, I have made five points in forming an argument that the future of open-source software will have serious implications for business. They are:

1. Open-source software offers the potential for substantial cost savings;
2. Open-source software has already begun to change the power structure of the software sector, particularly in diminishing Microsoft's position;
3. Open-source software challenges the philosophy of commercial business;
4. Open-source software is supported by an immense volume of contributors making its development and improvement rapid but unpredictable;
5. The emergent, bottom-up nature of open-source software's innovation is, in itself, suggestive of future models of product development.

Five scenarios emerge from the interplay and implications of the previous points. The timeline is the short-medium future; no more than five years.

Scenario 1: Changing of the Guard

Open-source software takes over as the principle supplier and developer of software to business. This scenario will be driven chiefly by cost savings and the diminishment of Microsoft's monopolistic power. Proprietary software will severely diminish as generic open-source alternatives are readily available with only switching costs as a barrier. Mass market open-source software will be supplemented by companies seeking to leverage the customisation market and other support services.

Scenario 2: The Triumph of Capitalism

History repeats itself as software engineers and businesses protect the innovations they develop for open-source software, and the proprietary software market returns to prominence. This scenario is driven mainly by the philosophical challenge to commercial business that open-source software stimulates. Business-employed software engineers and independent software developers recognise the value of their innovations and begin to protect and commercialise them. In-company developers will either be forced to become secretive about their customisations, or alternatively, they will be on-sold for profit. The most important developments and modifications will become proprietary, encouraged by a few radical leaps in innovation that its owners are reluctant to part with without compensation. Business can soon only download a very basic version of software and need to purchase the right "add-ons" in order to reach an acceptable level of functionality. Commercialism is sovereign.

Scenario 3: Software as Differentiation

The exponential increase in open-source software communities leads to a proliferation and fragmentation of software versions. Business is forced to find help in selecting the right version in the hopes that it can lead to a form of competitive advantage and differentiation. Presently, software is principally used in business for infrastructure purposes; it fills essentially the same functions for most businesses. But if open-source software is driven by the communities that develop it, it is more likely that the versions of software will fragment. This will be driven by a huge range of additional Internet sites where communities can form. Businesses will be faced with a dilemma. Proprietary software will diminish but remain expensive. Alternatively, business can turn to open-source software. But instead of one version, there will be thousands, each offering a unique product. Differentiation will be the rule. Companies will look for software that can offer more than infrastructure; a competitive advantage that suits their particular needs. As most software development will be undertaken by Internet communities, the commercial opportunities will be exploited by businesses that can understand this massively fragmented marketplace, and provide the right recommendations.

Scenario 4: Open-Source Out-Muscled

Further consolidation of large proprietary software vendors leads to a handful of immense and well-resourced conglomerates that successfully squeeze open-source software out of the market through government lobbying, over-zealous litigation and brokering deals. Like scenario two, here open-source software suffers at the hands of commercial imperatives. However, in this scenario it is because of the actions of large proprietary vendors. This may come from insistent lobbying to the government for legislation and exclusivity opportunities in the name of economic activity, and ruthless legal action. Gradually, the open-source communities left will only be made up of hardcore engineers who are prepared to run the risk of legal action. In addition, the key players such as Microsoft will increase their recent trend of offering incentives including cash to companies for converting back to Microsoft software. Before the trend becomes a policy, the large software vendors will have crushed the open-source competition and they will regain most of the software development and servicing opportunities. Open-source software becomes an underground, niche activity.

Scenario 5: Uneasy Co-existence

Open-source and proprietary software find an uneasy balance, each dominating different parts of the market, and each offering products and services that bolster the other. Business has no alternative but to utilise both. If the power of the emergent model of innovation in open-source software development is sufficient, but a lack of direction and unpredictability remain, the market will divide. In this scenario, open-source software will be slow to solve specific problems, because while it can develop quickly, it does so in its own organic way. Companies will still need to solve specific problems, such as those associated with integration and coordination of logistics, human resources and finance.

However, the general software market will be dominated by easily available, high quality, generic open-source software. This will mean that proprietary software vendors will be forced to consolidate. Proprietary products will dominate the customised markets and will be first to the market as providers of leading, new software. On the other hand, open-source software will always catch up and overtake commercially available products, leaving business with the question of whether they are prepared to pay for something new that potentially offers a competitive advantage. Although a different market, it is not unlike the present one. Proprietary software companies will find ways to add value to open-source versions, thus finding the market opportunities. Software bundling relationships – where software like Microsoft Office is default on hardware provided by manufacturers like Toshiba – will be devolved. Instead, software will be provided on the basis of customer needs and requests. Open-source software and the work of proprietary software companies will become inextricably entangled. Both have a role to play in the market. The main implication, however, is that in such a market there is no room for monopolistic power.

Final Comments

Over the next few years, more and more organizations will convert to open-source software. The software has developed sufficient robustness and security over a range of essential platforms so that organizations can solve most or all of their operational needs, including databases, web servers and business intelligence through to conventional word processing or "Office" software. It is not the inexorable future for software, but it seems a safe bet to conclude that it will play a key role in the future of business if for no other reason than its present market share, cost, reliability, performance, functionality, customisability, security and freedom to manipulate. At the risk of falling into the oldest forecaster's trap of making a prediction, I tend to favour scenario five. Open-source software is well beyond a trend. It is here to stay. However, the expectation that it will overhaul the market entirely is dangerous because it assumes that proprietary software companies will die quietly. I am convinced that commercial opportunity will prevail. In other words, commercial software vendors will find a way of staying in the market even if it means providing support products for open-source software or providing integrative software to make it compatible with their own. I believe this will also happen in reverse. But this marketplace I envisage is not one in which a market leader the size of Microsoft can maintain its dominance. Windows will be opened one way or another.

Correspondence

Aaron Smith
Associate Professor, La Trobe University
School of Sport, Tourism & Hospitality Management
Faculty of Law & Management
Melbourne, Australia
aaron.smith@latrobe.edu.au

References

Awazu,Yukika and Kevin Desouza. 2004. "Open Knowledge Management: Lessons from the Open Source Revolution." *Journal of the American Society for Information Science and Technology* 55(11): 1016-1020.

Glass, Robert. 2004. "A Look at the Economics of Open Source Association for Computing Machinery." *Communications of the ACM* 47(2): 25.

Holtgrewe, Ursula. 2004. "Articulating the Speed(s) of the Internet: The Case of Open Source/free Software." *Time & Society* 13(1): 129.

Ljungberg, Jan. 2000. "Open Source Movements as a Model for Organizing." *European Journal of Information Systems* (9): 208-216.

O'Reilly, Tim. 1999. "Lessons from Open-Source Software Development." Association for Computing Machinery. *Communications of the ACM* 42(4): 32-37.

Stallman, Richard. 1999. "The GNU Operating System and Free Software Development."

In Open Source: Voices from the Open Source Revolution. Sebastopol: O'Reilly & Associates.

von Hippel, Eric. 2001. "Innovation by User Communities: Learning from Open-Source Software." *MIT Sloan Management Review* 42(4): 82-86.

von Hippel, Eric and Georg von Krogh. 2003. "Open Source Software and the 'Private-Collective' Innovation Model: Issues for Organization Science." *Organization Science* 14(2): 209-226.

von Krogh, Georg. 2003. "Open Source Software Development." *MIT Sloan Management Review* 44(3): 13-18.

von Krogh, Georg., Sebastian Spaeth and Karem Lakhani. 2003. "Community, Joining, and Speciali zation in Open-Source Software Innovation: A Case Study." *Research Policy* 32(7): 1217.