THE RATIONALES BEHIND FREE AND PROPRIETARY SOFTWARE SELECTION IN ORGANISATIONS

Damien J. Sticklen and Theodora Issa
Curtin University - School of Management, Bentley, Western Australia

ABSTRACT

The aim of this paper is to critically examine the important assumptions behind the software-selection function in organisations. Software is incorporated in many situations within enterprises due to its unique ability to efficiently and effectively augment business functions and processes. Proprietary software with its inherent advantages and disadvantages remains dominant over "Free and Open-Source Software" (FOSS) in a large number of cases. However, the arrival of cloud-computing almost certainly mandates a heterogeneous software environment. Open standards, upon which most FOSS is based promotes the free exchange of information, a founding requirement of the systems embedded in organisations. Despite evidence to the contrary, the fact that FOSS is also available at low financial cost, combined with the benefits implicit in facilitating inter-process communication supports the view that it would be attractive to organisations. This paper approaches this paradoxical situation by examining the relevant literature in a broad number of disciplines. n important aspect examined is the roles that management, and in particular the executive, play in the software-selection function. It is on the basis of these findings that the rationales of use for both proprietary and FOSS are discussed in a multi-disciplinary context. Understanding the rationales behind the software-selection function may provide academics and practitioners with insight into what many would consider an ICT-centric problem. However, by abstracting to the management context, as opposed to the technical context, the organisational issues surrounding both proprietary software and FOSS adoption are counter-intuitively brought to the forefront.

KEYWORDS

Ubiquitous Computing; Protocols and Standards; Cyber-law and Intellectual Property; Social and Organizational Aspects; Freedom of Expression.

1. INTRODUCTION

Open standards are often discussed in academic literature when referring to the potential improvement of communications between organisations, business functions and processes. However, in management, less emphasis is placed on one of the key requirements of this integration – software. The benefits of hardware improvements over past decades have been readily measurable. However, the organisational benefit of adopting software has been largely intangible (Wiederhold 2009, p.9). Building on the foundations of open standards, but less frequently discussed, let alone adopted is "Free and Open-Source Software" which takes the form of both system and application programs. Its counterpart, proprietary software has enjoyed enterprise acceptance in both server and desktop environments for the past twenty to thirty years. While the literature tends to focus upon the advantages and disadvantages of proprietary and "Free and Open-Source Software" (FOSS), less attention has been paid to the software-selection processes within organisations, which has lead to the software-adoption landscape of today. The fact that significant research in this area has not been conducted as part of the management literature is anomalous, given that software is an important enabler of business processes (Guo & Zou 2008, p.333). Also difficult to reconcile are the following facts when viewed collectively rather than in isolation: 1. FOSS (GNU/Linux, for example) has some market penetration in server environments, but only moderate desktop use (Gray 2008, p.4); 2. FOSS is essentially free as a product (in both cost and user rights) while proprietary software-adoption typically invokes significant costs; 3. The quality of FOSS is beginning to, or in some cases as already met that achieved by proprietary equivalents; and 4. Most businesses are interested in reducing expenditure, while attempting to maintain the level of quality delivered by IT (what should be a key concern for enterprise executives, regardless of technical orientation). The situation described above would not be so significant if it were not for the paradigm-shift which is expected by some technologists in the near future – ubiquitous computing. Ubiquitous computing, enabled by cloud-based computing technologies, promises to enable access to data and processing power at almost any populated area of the world (Hooft & Swan 2006). Furthermore, cloud-computing, which may be of significant benefit to enterprises would likely make use of FOSS in addition to proprietary software. This is partly due to FOSS licenses which are seen as compatible with the requirements of cloud-based computing. Another contributory factor is that some well-established FOSS such as GNU/Linux has a proven security track record, mitigating some of the risks posed by cloud-computing. However the notion that FOSS is free of financial and litigate risk, which has been suggested by some of the on-line media, is based on false premises, as demonstrated by recent industry events related to its adoption (Foo 2009; Bray 2009). Cloud-computing implies at least some modicum of outsourcing, a practice which many managers and management academics associate with both benefit and risk to organisations. The immediate uncertainties surrounding FOSS, proprietary software, and cloud-based computing technologies supports the view that further research will be of benefit to both academics and practitioners in their understanding of these emerging events.

2. ORGANISATIONS AND THEIR RELATIONSHIP WITH IT

Executives are mandated to evaluate, direct the development, and monitor the performance of organisational work-flow mechanisms including functions, processes, and tasks in order to achieve organisational goals (Standards Australia & Standards New Zealand 2010, p.7). However, invariably, whether they are for-profit, non-profit, or governmental, contemporary organisations are mandated to rely on software to facilitate this management. For the purposes of this paper, a business functions is:

A group of business-related processes that support the business. Functions can be decomposed into other sub-functions and eventually into processes that do specific tasks (Whitten & Bentley 2007, p.51).

The literature suggests that:

Business processes are the work, procedures, and rules required to complete the business tasks, independent of any information technology used to automate or support them (Whitten & Bentley 2007, p.21).

This definition implies that technologies are not embedded within business processes. In practice however, technology, and in particular software, are inexorably embedded within business processes, and therefore, also within business functions (about which, the first definition is silent) (Pautasso et al. 2007). This is exemplified by the fact that large organisations often rely heavily on such systems as enterprise resource planning (ERP) to accurately manage resources (Samson & Daft 2009, p.76) and web-enabled technologies to reach markets. Such is the ideal state of management and IT as disciplines. The responsibility for IT governance cannot reside with the technical implementers of systems, but with the executive. Software-selection for the purpose of augmenting business processes must be a consultative process between managers and IT experts (Standards Australia & Standards New Zealand 2010, p.9). Building upon the point made above, is that if cloud-computing becomes at first a source of global competitive advantage and then, in the long-term, a mainstream and normal requirement of doing business, prior academic research suggests that Australian organisations are not culturally and strategically prepared for the adoption of the software infrastructure necessary (i.e. incorporating FOSS) (Goode 2005, p.675). Given that IT is a fertile ground for frequent change, it can be seen that decision-making which is affected by or affects IT cannot be divorced from the executives' mandate, even if by "insourcing" to the IT department. This implies that if software-selection is to be effective, it should incorporate the heterogeneous skill-sets of IT and management amongst relevant and value-adding disciplines, in a synergistic approach. The following section briefly outlines software-selection practice for FOSS in reality versus best-practice software-selection for all software types.

3. SOFTWARE-SELECTION

Hauge et al. (2009, p.42) state that the adoption of FOSS is widespread amongst companies which have increasingly sought modular solutions to business problems. However, there is evidence that suggests that

GNU/Linux, the primary platform for FOSS delivery has an estimated market share of just over one percent of all computers globally (Net Applications 2010; StatCounter 2010), a fact that implies either that business computing is a small proportion of the overall computing market, or that proprietary operating systems are entrenched in both business and consumer environments. Interestingly, Hauge et al.'s (2009, p.43) research also suggests that the process of software-selection undertaken by developers in this context is not grounded by a generalised framework. Considering that the general purpose of software acquisition is to solve a problem related to business processes in a structured way, it would be reasonable to expect that the process would be at least as systematic as that which is suggested by Whitten and Bentley (2007, p.30), an approach which is similar to that outlined in the IT Infrastructure Library (ITIL). One reason why this might not be the case is that the adoption of FOSS is expected by some to be fiscally low-risk (Raab 2007). Research also suggests that for FOSS, rather than identifying software components on the merit of what will solve the problem in the best available way, the first software component to solve the problem is integrated instead (Hauge et al. 2009, p.45). Although the stated limits of the research conducted by Hauge et al. (2009, p.46) include that the companies interviewed were only small and FOSS components were not utilised for mission-critical roles, there is also an additional limitation of the study – it was geographically clustered. If these results are generalisable to corporations of larger sizes and in different locations, the software-selection process for FOSS undertaken by IT professionals would have to be considered problematic for two reasons: firstly, the adoption of FOSS is not universally accepted to be devoid of financial risk (Ruffin & Ebert 2004, p.85; Foo 2009); and secondly, the concept of using 'first fit' rather than 'best fit' does not appear to be congruent with a rigorous development, testing, and integration process as suggested by IT management frameworks such as ITIL.

The Information Technology Infrastructure Library (ITIL) is a well-known standard for IT management throughout the world (Addy 2007, p.1). Version 3 of this framework provides a richer set of considerations for software management than practised in the above case. This software development and selection process is described as 'Applications Management' and includes: *Requirements; Design; Build; Deploy; Operate; and Optimise (Cannon & Wheeldon 2007, p.131)*. In light of the set of considerations offered by ITIL, it appears in the above study that the parameters used by these integrators of FOSS technologies are limited to requirements and deployment. Arguably, software components, regardless of their origin need to be assessed by their individual strengths and weaknesses against a formalised framework, even if that framework is developed internally.

4. OLD AND NEW ARCHITECTURES FOR SERVICE DELIVERY

In the general business context, a considerable amount of literature has been written with regards to the social and environmental results of organisations, in addition to the obligatory *financial* results. The rationale behind encouraging or pressuring organisations to go beyond financial results largely stems from perceived market failures and their resultant negative impacts on both private citizens and the ecology of the planet (Melville & Ross 2010, p.1).

Organisations which produce and employ large amounts of ICT infrastructure are on notice concerning environmental issues such as electricity consumption and electronic waste (Jeurissen 2000, p.229). Increasingly, companies are becoming responsive to requests and demands for them to become environmentally conscientious due to the financial benefits associated with compliance and the threat of litigation in its absence (Melville & Ross 2010, p.1; Jeurissen 2000, p.229). Previous architectures for mass computing required one operating system per server, which implied that the organisation's purchase of servers was not driven entirely by computational power requirements as much as it was for the need to access multiple operating systems and applications (Baschab & Piot 2007, p.202). Recognising the inherent inefficiency of this arrangement, organisations have sought consolidation of servers and associated equipment. One way in which this consolidation has been achieved is through the introduction of virtualisation, which is a probable cornerstone of cloud-computing (Tata Consultancy Services 2010, p.6).

Virtualisation is a means by which multiple operating systems and associated software may run on common hardware. The purported advantages of virtualisation include a decreased requirement to purchase additional hardware to increase scale and potential reductions in overall electricity consumption (Baschab & Piot 2007, p.202). This reduction in electricity consumption assumes that efficiency gains in floor space are not used for further ICT installations. If one takes an optimistic view, it can be seen that virtualisation potentially offers lower financial and environmental costs than traditional architectures. The above discussion has focused on the environmental benefits of recent software developments but has been silent on the perceived social benefits and costs of software development methods and use in organisations. Furthermore, by and large, there is no debate presented in the above discussion because the facts presented are not considered to be contentious. However, the ideologies behind proprietary and FOSS must be considered polar opposites (Schmidt 2004, p.679).

5. INTRODUCTION OF SOFTWARE TYPES

5.1 Proprietary Software: The Traditional Model

In 1976, an open letter concerning software licensing was written and presented by Bill Gates. In this letter, he outlined the efforts by Paul Allen and himself to develop some software for the hobbyist community, only to have this software copied and 'stolen' by users who were unwilling to pay for its use (Gates 1976, p.2). It was his and others' perception that the efforts of software developers should be remunerated financially that led to the concept of proprietary software, as it is now known. In 1976, through US federal copyright legislation, Gate's opinion on this issue was authoritatively confirmed. Software programs which implement mathematical algorithms began to be officially protected by US copyright law in 1978, when the 1976 legislation came into effect (Franz 1985, p.147). Previously, it had been legislated that only material which could be viewed with the human eye would be afforded protection by earlier copyright legislation (Franz 1985, p.148). This was insufficient for those who sought to profit from the emerging personal computing market due to the large amount of time, labour, and finance required to successfully commercialise software (Franz et al. 1981, p.56).

Proprietary software is usually provided without the source-code (Simon 2005, p.231). Because of the desire to protect trade-secrets and profits, software is given protections to promote innovation throughout parts of the world (Freedman 2005, p.631), including Australia by enforcement of patent, copyright, and trademark legislation (Latimer 2009, p.139). Furthermore, the ownership of software is not passed onto the end-user as happens with physical property. Rather software is usually licensed under the terms of an End-User License Agreement (EULA). This situation artificially makes software somewhat similar *and* different to physical property. Because EULAs regulate activities which might occur in conjunction with software, under certain circumstances, managers need to understand the importance of remaining in compliance due to the potential for litigation and court-imposed sanctions. For example, Psystar Corporation, in 2008 began selling hardware bundled with OS X (Feintzeig 2009), a UNIX-based operating system produced by Apple, but including some key FOSS components. Despite the presence of FOSS components (Apple Inc. 2007, p.2), the Apple EULA acts as a wrapper over the use of the software compilation as a whole, making it in effect, a proprietary software compilation. The Apple EULA explicitly prohibits the running of software covered by this license on hardware, other than that sanctioned by Apple Corporation (Apple Inc. 2007, p.1). Psystar, like others, suspected that this was allowing Apple Inc. an obfuscated market dominance. Psystar however was found to be in breach of copyright (Bawaba 2009) and the OS X EULA, faced litigation, then subsequently received court-imposed sanctions including fines exceeding $2 million (Burrows 2009).

As stated above, the protections afforded to software and in particular, proprietary software are designed by governments to promote innovation (McTaggart et al. 2007b, p.252) which in turn are expected to provide benefits the consuming public through superior products and services (McTaggart et al. 2007b, p.282). It can also be deduced that this introduction of 'legal monopolies' through property rights is to intended create a market around the commercial development of software, which would be less likely to occur if these rights did not exist (McTaggart et al. 2007a, p.48). A problem with enforcing ‘intellectual property’ laws is that rather than preventing outright theft, these laws it could be argued, serve to protect against the *opportunity costs* created by those who obtain bootlegged copies of the original software i.e. lost sales, not stolen inventory, for the software vendor. Without broad compliance with these laws, it would not be inaccurate to suggest that software is inherently non-rival in the economic sense i.e. one's use of a bootlegged copy does not significantly impede the sale of a legitimate copy to another. Unlike most companies which manufacture physical products, software vendors have significant research and development costs (Chiang 2010, p.104) which must be reimbursed through comparatively lower costs of production and distribution as offered by optical media and the Internet. Copyrights, patents, and trademarks support and protect this approach to innovation, development, and profit generation. However, if the software vendor enjoys market domination, such as Microsoft does with their desktop operating system, MS Windows (Net Applications 2010; StatCounter 2010), these laws which were once designed to afford protection to their software may become a source of *durable* competitive advantage. This would not serve to benefit the general public, including corporate customers of the market dominant software vendor, but rather would further the interests of shareholders in that company (Federspiel & Brincker 2010, p.40). Through the above discussion, the manager may now understand why software volume licensing costs can be considerable. The increasing trend toward monopolies on technologies is not lost on the courts throughout Europe and the US. For example, Microsoft has recently faced pressure from the European Union (EU) not to abuse their market dominance of their operating system and web browser (Forelle & Wingfield 2009; Palmer & Tait 2009). Furthermore, in the “In re Bilski” case, the applicability of business-process related and software patents was tightened (Hulse et al. 2010, p.12; McFarlane & Litts 2010, p.71). This precedent is not binding in Australian law, but does present a strong persuasive argument regarding the decreasing scope to which software patents may protect software-based inventions in the future. Therefore the manager needs to ask; if the traditional business model surrounding proprietary software were to be slowly eroding, what other models might emerge?

5.2 FOSS and Cloud Computing: The Emergent Models

As early as 1984, computer scientists, such as Richard Stallman, considered the proliferation of restrictions within the licenses of proprietary software to be contra-preferential to rights of software users (Wolf et al. 2009, p.279). As a computer scientist, Stallman expected to have access to the source code of programs in order that efficient modification and subsequent distribution of improvements could be achieved. In proprietary software, this code is unlikely to be available for end-users and its unauthorised use is usually prevented by copyrights, patents, or a combination of both. Furthermore, free software advocates like Stallman believe that software patents, proprietary formats and standards can threaten the rights of the end users in ways which are neither ethically or socially responsible (Brown 2010). Although not opposed to the commercialisation of software (Wolf et al. 2009, p.280), Stallman has stated through the Free Software Foundation, which he established, that four fundamental rights or freedoms should exist for all users of software:

“The freedom to run the program, for any purpose; the freedom to study how the program works, and change it to make it do what you wish. Access to the source code is a precondition for this; the freedom to redistribute copies so you can help your neighbour; and the freedom to distribute copies of your modified versions to others” (Free Software Foundation, Inc. 2010b; Baschab & Piot 2007, p.171).

It is apparent, all things remaining the same, that software licensed under such terms will have a diminishing marginal cost (not benefit) at each instance of distribution. Also obvious is that the protections provided by the law are less applicable to free software, and therefore the methods of generating profits that apply to proprietary software are generally unavailable (Freedman 2005, p.631). The four freedoms are captured within the various versions of the GNU General Public License (GNU GPL), which like the terms of an EULA, are legally binding (Kumar 2006, p.35; Wacha 2005, p.492). A potential problem for commercial developers of software is that if part of their software is copied from or linked to GNU GPL licensed source code, the entire body of their code becomes subject to the GNU GPL upon distribution (Norris & Kamp 2004, p.44; Ruffin & Ebert 2004, p.85). This restriction has led some to criticise the GNU GPL and others to liken it to a 'cancer' which binds itself to commercial inventions (Scacchi 2007, p.6; Kuehnel 2008, p.115).

Regardless of the vehement claims from both sides of the debate, FOSS is generally improving in functionality over time, which with the advent of cloud-computing, should increase its relevance. While FOSS might be seen as an important complement to proprietary software, 'cloud-computing' is however seen by some to be a paradigm-shift away from both of these software distribution-models. Large organisations have traditionally made use of data-centres to provide massive data processing and storage. It is expected in future, with the adoption of cloud-computing, organisations will make increasing use of software-as-a-service (SaaS) and infrastructure-as-a-service (IaaS), with virtualisation to provide software and data processing capabilities to the organisation via a utility model (Sotomayor et al. 2009, p.14; Schneider 2009, p.19). These capabilities may offer financial benefits through reduced fixed costs. Additionally, cloud-computing may mitigate some of the burden placed upon the environment by reducing electricity use and decreasing electronic waste (Kavanagh 2009). Surprisingly, despite the potential sustainability advantages potentially available through cloud-computing technologies, the Free Software Foundation (2010a) has indicated that the term is confusing, even going so far as to quote Oracle Corporation's CEO, Larry Ellison, creating some doubt over the term's delimitation (Farber 2008). This aversion to the term is likely due to the potential threat that these parties perceive cloud-computing creates for their ideals and business models. Cloud-computing appears to challenge industry knowledge such that in order to better understand the business issues surrounding each development model, identifying what managers need to know is required.

6. SOFTWARE-ADOPTION AND THE KNOWLEDGE GAP

Baschab and Piot (2007, p.172) argue that IT managers, including executives, have faced corporate resistance to the adoption of FOSS in the past due in part to the differing values, beliefs, and assumptions between FOSS organisations and commercial enterprises. Another challenge has been that proprietary software has arguably had a superior appearance to that offered by FOSS (Baschab & Piot 2007, p.172). Given that earlier discussion has supported the view that not all decisions concerning software are made on the basis of technical merit, this result is not surprising. Adding to these impediments to adoption is the fact that managers may not be cognisant of the existence of FOSS (Baschab & Piot 2007, p.173). Furthermore, research conducted by Goode (2005, p.675) quoted in Table 1 suggested that there may even be a conscious decision to reject FOSS by managers, particularly in Australia due to the perceptions listed in Table 1.

Table 1. Reasons why FOSS use is not a priority in top Australian organisations (Goode 2005, p.675).

Rationale for Omission of FOSS in Organisation	**Percentage**
Lack of relevance	36.00%
Lack of support	20.00%
Minimal or No Requirement	16.00%
Insufficient Resources	8.00%
Committed to Microsoft	8.00%
Not Commercial	8.00%
No time	4.00%

7. CONCLUSION

This paper has provided the reader with a critical examination of information that might appear outside the realm of traditional management. However, in the twenty-first century, inculcating knowledge silos does not benefit organisations, it impedes them. Furthermore, with a dependence on automated systems, management more than ever requires a heterogeneous skill-set which borrows from other discrete disciplines. It is expected that the reader would now understand the fundamentals and implications of FOSS, proprietary software, and cloud-computing as they relate to organisations. There are significant business opportunities and challenges that arise when an organisation chooses to adopt any of these software development and distribution models. Although the use of proprietary software is widespread throughout the world and Australia, relatively scant empirical information regarding FOSS software and cloud-computing selection processes in Australian organisations has been identified. This creates a potentially imbalanced understanding of software-selection at the academic level, and a basis for bias in the practice of software-selection. Therefore, given that there is potential for significant disruptive effects in social and environmental terms, future research to obtain this information ought to be of particular utility to IT and management academics as well as the organisations on which this research would be focused.

REFERENCES

1. Addy, R., 2007. *Effective IT Service Management: To ITIL and Beyond!*, New York: Springer Berlin Heidelberg.
2. Apple Inc., 2007. Software License Agreement for MAC OS X. Available at: http://images.apple.com/legal/sla/docs/macosx105.pdf [Accessed May 8, 2010].
3. Baschab, J. & Piot, J., 2007. *The Executive's Guide to Information Technology* 2nd ed., New Jersey: John Wiley and Sons.
4. Bawaba, A., 2009. Apple Moves to Kill Second Psystar Lawsuit. *Network World Middle East*. Available at: http://proquest.umi.com.dbgw.lis.curtin.edu.au [Accessed August 5, 2010].
5. Bray, H., 2009. Microsoft Files Suit Against GPS Device Maker. *Boston Globe*, B.7.
6. Brown, P.T., 2010. We Must Make Freedom our Goal. Available at: http://www.fsf.org/appeal/2009/freedom-is-the-goal [Accessed May 17, 2010].
7. Burrows, P., 2009. Lessons from the Apple-Psystar Battle. *Business Week (Online)*. Available at: http://proquest.umi.com.dbgw.lis.curtin.edu.au [Accessed August 5, 2010].
8. Cannon, D. & Wheeldon, D., 2007. *Service Operation*, London: The Stationery Office.
9. Chiang, C., 2010. Product Diversification in Competitive R&D-Intensive Firms: An Empirical Study of the Computer Software Industry. *Journal of Applied Business Research*, 26(1), 99-108.
10. Farber, D., 2008. Oracle's Ellison Nails Cloud Computing. Available at: http://news.cnet.com/8301-13953_3-10052188-80.html [Accessed May 17, 2010].
11. Federspiel, S. & Brincker, B., 2010. Software as Risk: Introduction of Open Standards in the Danish Public Sector. *Information Society*, 26(1), 38-47.
12. Feintzeig, R., 2009. Computer Maker sued by Apple files for Chapter 11 Protection. Available at: http://proquest.umi.com.dbgw.lis.curtin.edu.au [Accessed August 5, 2010].
13. Foo, F., 2009. Kennards Hire offloads Linux Machines after costly Experiment. *The Australian*. Available at: http://www.theaustralian.com.au/australian-it/kennards-hire-offloads-linux-machines-after-costly-experiment/story-e6frgakx-1225810376238 [Accessed May 14, 2010].
14. Forelle, C. & Wingfield, N., 2009. Corporate News: EU Hits Microsoft With New Antitrust Charges --- Reviving Old Fight, Regulators Accuse Software Giant of Harming Competitors by 'Tying' Web Browser to Windows. *Wall Street Journal*. Available at: http://proquest.umi.com.dbgw.lis.curtin.edu.au [Accessed September 5, 2010].
15. Franz, C.R., 1985. Re-examining the Proprietary Software Protection Issue in the U.S.A. for the 1980s. *Information & Management*, 8(3), 147-153.
16. Franz, C.R., Wilkins, S.J. & Bower, J.C., 1981. A Critical Review of Proprietary Software Protection. *Information & Management*, 4(2), 55-69.
17. Free Software Foundation, Inc., 2010a. Some Confusing or Loaded Words and Phrases that are Worth Avoiding. Available at: http://www.fsf.org/licensing/education/essays/index_html/words-to-avoid.html [Accessed May 17, 2010].
18. Free Software Foundation, Inc., 2010b. The Free Software Definition. Available at: http://www.gnu.org/philosophy/free-sw.html [Accessed May 8, 2010].
19. Freedman, W., 2005. Virtual Speech: At the Constitutional Crossroads. *Santa Clara Computer and High - Technology Law Journal*, 21(4), 629-643.
20. Gates, B., 1976. An Open Letter to Hobbyists. *Homebrew Computer Club Newsletter*, 2(1), 2.
21. Goode, S., 2005. Something for Nothing: Management Rejection of Open Source Software in Australia's Top Firms. *Information & Management*, 42(5), 669-681.
22. Gray, J., 2008. Linux and the Enterprise Desktop: Where are we Today? *Linux Journal*, 2008(171), 7.
23. Guo, J. & Zou, Y., 2008. A Business Process Explorer: Recovering Business Processes from Business Applications. In *Reverse Engineering, Working Conference on*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 333-334.
24. Hauge, O., Osterlie, T., Sorensen, C. F. & Gerea, M. 2009. An Empirical Study on Selection of Open Source Software-Preliminary Results. In *Proceedings of the 2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development*. pp. 42–47.
25. Hooft, M. & Swan, K., 2006. *Ubiquitous Computing in Education: Invisible Technology, Visible Impact*, Routledge.
26. Hulse, R., Sachs, R. & Patel, R., 2010. Patent Practice and In re Bilski. *Computer and Internet Lawyer*, 27(2), 7-12.

27. Jeurissen, R., 2000. Cannibals with Forks: The Triple Bottom Line of 21st Century Business. *Journal of Business Ethics*, 23(2), 229-231.
28. Kavanagh, R., 2009. IT Virtualization Helps to Go Green. *Information Management*, 19(2), 20.
29. Kuehnel, A., 2008. Microsoft, Open Source and the Software Ecosystem: of Predators and Prey - the Leopard can Change its Spots. *Information & Communications Technology Law*, 17(2), 107-124.
30. Kumar, S., 2006. Enforcing the GNU GPL. *Journal of Law, Technology & Policy,* 1-36.
31. Latimer, P., 2009. *2009 Australian business law*, North Ryde: CCH Australia Limited.
32. McFarlane, R. & Litts, R., 2010. Business Methods and Patentable Subject Matter Following In re Bilski: is "Anything under the Sun Made by Man" Really Patentable? *Santa Clara Computer and High - Technology Law Journal*, 26(1), 35-83.
33. McTaggart, D.F., Parkin, M. & Findlay, C.C., 2007a. *Macroeconomics* 5th ed., Frenchs Forest: Pearson Education Australia.
34. McTaggart, D.F., Parkin, M. & Findlay, C.C., 2007b. *Microeconomics* 5th ed., Frenchs Forest: Pearson Education Australia.
35. Melville, N. & Ross, S.M., 2010. Information Systems Innovation for Environmental Sustainability. *MIS Quarterly*, 34(1), 1-21.
36. Net Applications, 2010. Operating System Market Share. Available at: http://marketshare.hitslink.com/operating-system-market-share.aspx?qprid=10&qpcal=1&qpcal=1&qptimeframe=M&qpsp=135 [Accessed May 8, 2010].
37. Norris, J. & Kamp, P., 2004. Mission-Critical Development with Open Source Software: Lessons Learned. *IEEE Software*, 21(1), 42-49.
38. Palmer, M. & Tait, N., 2009. Brussels Accepts Microsoft's Browser Offer. *Financial Times*, 24.
39. Pautasso, C., Heinis, T. & Alonso, G., 2007. Autonomic Resource Provisioning for Software Business Processes. *Information and Software Technology*, 49(1), 65-80.
40. Raab, D.M., 2007. Software Selection Methods, Part 2. *DM Review*, 17(5), 34.
41. Ruffin, M. & Ebert, C., 2004. Using Open Source Software in Product Development: a Primer. *IEEE Software*, 21(1), 82–86.
42. Samson, D. & Daft, R., 2009. *Management* 3rd ed., South Melbourne, Vic: Cengage Learning Australia.
43. Scacchi, W., 2007. Free/Open Source Software Development: Recent Research Results and Emerging Opportunities. In *The 6th Joint Meeting on European software engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering: companion papers*. p. 468. Available at: http://www.ics.uci.edu/~wscacchi/Presentations/ESEC-FSE07/ESEC-FSE07-Scacchi-Paper.pdf [Accessed May 18, 2010].
44. Schmidt, D.P., 2004. Intellectual Property Battles in a Technological Global Economy: A Just War Analysis. *Business Ethics Quarterly*, 14(4), 679-693.
45. Schneider, M., 2009. How to Keep Your Head in the Clouds. *Information Management (1521-2912)*, 19(2), 18-20.
46. Simon, K.D., 2005. The Value of Open Standards and Open-Source Software in Government Environments. *IBM Systems Journal*, 44(2), 227-238.
47. Sotomayor, B., Montero, R., Llorente, I., & Foster, I., 2009. Virtual Infrastructure Management in Private and Hybrid Clouds. *IEEE Internet Computing*, 13(5), 14-22.
48. Standards Australia & Standards New Zealand, 2010. Corporate Governance of Information Technology. Available at: http://www.saiglobal.com.dbgw.lis.curtin.edu.au [Accessed June 5, 2010].
49. StatCounter, 2010. Top 5 Operating Systems on Apr 10. Available at: http://gs.statcounter.com/#os-ww-monthly-201004-201004-bar [Accessed May 8, 2010].
50. Tata Consultancy Services, 2010. TCS and Cloud Computing. Available at: http://www.tcs.com/resources/white_papers/Pages/TCS_Cloud_Computing.aspx [Accessed March 16, 2010].
51. Wacha, J.B., 2005. Taking the Case: is the GPL Enforceable? *Santa Clara Computer and High - Technology Law Journal*, 21(2), 451-492.
52. Whitten, J.L. & Bentley, L.D., 2007. *Systems Analysis and Design for the System Enterprise* 7th ed., New York: McGraw-Hill/Irwin.
53. Wiederhold, G., 2009. Tutorial: How to Value Software in a Business, and Where Might the Value Go? In *Advanced Information Systems Engineering*. pp. 9-10. Available at: http://dx.doi.org.dbgw.lis.curtin.edu.au/10.1007/978-3-642-02144-2_5 [Accessed July 19, 2010].
54. Wolf, M., Miller, K. & Grodzinsky, F., 2009. On the Meaning of Free Software. *Ethics and Information Technology*, 11(4), 279-286.