

December 2006

Benefiting from Open Source Development Methodologies in Global Information Systems Organizations

Christopher Dondanville
MIS Department

Thomas Stafford
MIS Department

Follow this and additional works at: <http://aisel.aisnet.org/amcis2006>

Recommended Citation

Dondanville, Christopher and Stafford, Thomas, "Benefiting from Open Source Development Methodologies in Global Information Systems Organizations" (2006). *AMCIS 2006 Proceedings*. 106.
<http://aisel.aisnet.org/amcis2006/106>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2006 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Benefiting from Open Source Development Methodologies in Global Information Systems Organizations

Christopher C. Dondanville

MIS Department

Fogelman College of Business and Economics

The University of Memphis

Memphis, TN 38152

chris.dondanville@memphis.edu

Thomas F. Stafford

MIS Department

Fogelman College of Business and Economics

The University of Memphis

Memphis, TN 38152

tstaffor@memphis.edu

ABSTRACT

The current trend in outsourcing development to foreign countries where cheap programming labor and extensions of the workday can be had is creating its own set of problems. The traditional development model runs into various communication, collaboration and management problems when employed in a Global Virtual Team (GVT). The Open Source Software (OSS) development model has been created with a globally dispersed, Internet connected, development team from its inception and thus has addressed some of the problems that GVTs are encountering. This paper looks at the problems that are introduced in the traditional development model when it is used by a GVT and proposes that the adoption of selected OSS methods and tools will help to alleviate some of those problems and provide the cost, quality and time benefits that are sought in the implementation of GVTs.

Keywords

Open Source Software, IT Governance, Global Virtual Teams, Off-shoring

INTRODUCTION

The convergence of global communications on the Internet, the globalization of the economy and the globalization of the corporation are creating new problems and opportunities for the software development process (Booch 2001; Dempsey, Weiss, Jones and Greenberg 2002). As corporations look to offshoring and global virtual teams (GVTs) to gain the benefits of cheaper labor and accelerated time schedules, the issues surrounding the distribution of traditional software development models are impacting them. We suggest that these corporations look to the Open Source Software movement (OSS), which is a successful model designed from the ground up to support collaborative global networks of developers (e.g., Shen 2005).

The OSS movement is responsible for such successful software products such as the Linux operating system, the Apache web server, the Mozilla browser, the Internet Domain Name System (DNS and BIND) and the sendmail program that serves as the primary routing engine for all Internet mail (Weber and Bussell 2005). Since its inception, the OSS movement has been distributed, global, multinational and able to produce significant results in software development scenarios (Dempsey et al. 2002). With advances in computing that are “shaping our economy and our future” emerging from the OSS culture (O’Reilly, 1999), perhaps it is time for corporate IT departments to take a closer look at the OSS development methodology and apply some of what they find in their own distributed, global, multinational environments to help solve some of the problems that they are experiencing in employing global virtual teams under a traditional development model (e.g., Booch 2001). The rest of this paper will lead us through an examination of the paradigms and methods of Open Source software development practices and Global Virtual Team software development. We will then compare the similarities of the two methods, showing where there are opportunities for sharing of concepts. The paper will conclude with a discussion of possible methods to leverage the open source development concepts that are identified.

OPEN SOURCE SOFTWARE AND DEVELOPMENT

The evolution of Open Source Software began in the open environment of the early Internet and the academic and research institutions that created it. A communal culture had arisen among the programmers, students and researchers on the ARPAnet. Indeed, the very term “open source” was an artifact generated by the move of Netscape in the early Internet days to give away the code for its browser in an effort to speed diffusion (Schmidt 2004). Open source development often takes

the form of a “communal culture,” in which cooperative development is prized in an effort to achieve a common good, and is best exemplified by the culture of software and technology research at the Massachusetts Institute of Technology, where research Richard Stallman founded the GNU project that underlies much of open source development today (Dempsey et al. 2002). Although this is a very simplistic description of a long and complex evolution, it is important to note the difference between the paradigm of open source and the evolving Proprietary Source Software (PSS) paradigm.

Stallman’s creation, the Free Software Foundation, was intended to create a mechanism that could be used by developers to preserve the “free” status of their software and guarantee a certain number of rights to all current and future users of the software. The result of this model was the General Public License or GPL (von Krogh 2003). Software that is released under the General Public License gives those possessing the software a right to use it at no cost, the right to the source code for examination and modification (hence “open source”), and the right to redistribute the modified or unmodified code at no cost as long as all software and derivatives are released under the GPL as well.

A common misconception about the OSS movement is that it is essentially the move to develop and popularize the GNU-based Linux operating system, when, in fact, the innovator and leader of the Linux Kernel Project, Linus Torvalds, is noted as saying that the most exciting aspects of the emerging OSS movement are currently in terms of applications, rather than operating systems (Dempsey et al. 2002, p. 68). Indeed, the movement toward the adoption of the Linux operating system shows signs of slowing somewhat (Anon 2005), as new adopters begin to realize costs unrelated to the practically “free” nature of the Linux system, while application development for the platform proceeds apace, substantially enhanced by the modular and cooperative aspects of the OSS development approach (Weber and Bussell 2005). This application development initiative in the OSS movement has seen particular growth in the developing world, which is an area of the global economy that can substantially benefit from the virtual team methods and tools pioneered in OSS approaches to development (Booch 2001; Kshetri 2004; Shen 2005).

The OSS Development Model with its team-based open sharing and collaboration, and distributed, global, Internet-connected development processes has been likened to a bazaar, in contrast to the top-down hierarchical methods of design and software construction (Raymond 2000). The collaboration, cooperative work environment and configuration management tools are an intrinsic part of this model (Cubranic and Booth 1999; Wu and Lin 2001). It would be important to note here that this is somewhat of a simplistic view of the Open Source Model. Not all projects are created equal, nor are all teams. The open source model is a collection of concepts that are employed to a greater or lesser extent in all open source projects. Another important aspect of the open source development model is the level of centralized control and authority. Many of the most successful open source projects (Linux and Apache) have had a high level of centralized control (Zhao and Elbaum 2003). The most conspicuous employment of the OSS development model is at the Sourceforge site (<http://www.sourceforge.net>). This site is dedicated to the open source model and even sells versions of its project tracking and management software to enable enterprise teams with the OSS development model.

With the employment of this model, interested and diverse parties can meet virtually, and can share ideas, cooperate and facilitate development efforts in a collaborative and open setting, much as GVT proponents would like to see Proprietary Source Software (PSS) development efforts proceed (Goodbody 2005; Paul, Samarah, Seetharaman and Mykytyn 2005). This provides the basis to consider the OSS movement as a template for emerging global virtual team operation of development processes, since many of the factors that led to the successful launch and evolution of the open source paradigm share similar success characteristics with the virtual software development teams that are increasingly being used on a global basis.

GLOBAL VIRTUAL TEAMS AND SOFTWARE DEVELOPMENT

Evolving in parallel with the OSS movement, the trend toward geographically dispersed virtual work teams has grown to become a leading method for software development (Kayworth and Leidner 2002); the impact of corporate global strategies, and vast improvements in technology-supported communication and collaborative processes have given rise to this new work approach (Harvey, Novicevic and Garrison 2004; Maznevski and Chudoba 2000). Global organizations are increasingly engaging in global software development, simply as an aspect of their connection to and access to different markets and resource bases (Edwards and Sridhar 2005; Goodbody 2005). However, companies learned quickly that global virtual teams have their own unique challenges and issues, including managing a distributed work force, cultural differences, time zone separation, technology incompatibilities and language barriers (Paul et al. 2005).

Innovative communication and learning skills are required for global team development (Zakaria, Amelinckx and Wilemon

2004), and the procedures refined over the past two decades of evolution in the OSS movement appear to have promising capabilities for solving many of the problems arising from the challenges of coordinating software development across international borders and cultural barriers – as, indeed, has been the case with the OSS movement from the beginning. Where information and communications technologies (ICT) are critical to successful GVT operation (Sanders, Van Slyke and Vogel 2004), the OSS movement has perfected the process of tying dispersed and disparate communities of developers together in common cause to achieve a common goal (Dempsey et al. 2002; Weber and Bussell 2005). For this reason, we suggest the OSS movement as a template for successful evolution of the GVT concept in global software development.

The type of outsourcing of IT development to other countries known as offshoring has been a side effect of the immediacy of Internet communication and data transportation (Edwards and Sridhar 2005; Harvey et al. 2005). The primary driver for this practice are the rising cost of software development combined with the need to maximize scarce development resources and a paucity of time for project completion (Maznevski and Chudoba 2000; Montoya-Weiss, Massey and Song 2001; Saunders et al. 2004). Such team structures allow for time compression of the work cycle by leveraging a virtual 24-hour work day, combined with the advantage of sharing a global network of knowledge and expertise. While these benefits are lucrative and are driving the move towards offshoring, there are a new set of problems related to managerial style and cultural differences across team members arising from the implementation of global virtual teams and the distributed software development paradigm (cf., Panteli and Davison 2005; Paul et al. 2005). GVTs are plagued with human and technology issues such as cultural and language barriers, as well as differences in IT proficiency, performance and expectations (cf., Goodbody 2005; Harvey et al. 2005; Paul et al. 2004). Technology differences, incompatibilities and infrastructures can also be seen to provide their share of opportunities for improvement (Dubé 2001; Saunders et al. 2004).

As regards the people side of the development equation, culture and language differences are key hurdles to overcome in GVT operation. As noted in Table 1, the issues of language and culture that inevitably arise in diverse multinational teams directly implicate technology issues (e.g., Sanders et al. 2004). While language barriers can be exacerbated by ICT systems incompatibilities, reliable flows of information can do much toward integrating diverse views of culturally disparate virtual team members toward a common goal (Kerber and Buono 2004). Technological proficiency among team members facilitates not only the operation of systems with which development will proceed, but also greatly facilitates the ICT-mediated information that supports group cohesion and cooperation towards development tasks. Indeed, some of the issues related to cross-border virtual team operation imply information flow restrictions that may exist between different host nations of team members, which can be quite problematic in view of the critical nature of such information flows in support of group efforts (Kerber and Buono 2004; Zakaria et al. 2004). Finally, there will often be culturally-bound differences between team subgroups that can have implications for successful operation (Dubé and Paré 2001; Panteli and Davison 2005; Zakaria et al. 2004). These issues can impact not only team performance and cohesion, but also matters as mundane as vendor preference and choice.

People and Organization Issues	Technology and Infrastructure Issues
Cultural Differences including the expected levels of performance, work behaviors, and conflict resolution.	Reliable channels of communications must be provided to allow for testing of interfaces and integration of modules
Language Barriers	Incompatibility of systems
Computer Proficiency including not only the technology employed in the systems development effort but also the proficiency with technologies that will be used in collaboration such as video conferencing, file sharing, and virtual meetings.	Telecommunications oriented issues and can be a real challenge in countries where even narrowband ISDN is not developed or prohibitively expensive.
National Strategies and Legalities including the restrictions on the flow of information and the distribution of test data or the prohibition of technology export to certain regions.	Reliable lines and means of communication are necessary to allow for the avoidance of redundant work and cost.
Standards development execution and enforcement across the cultural boundaries can be a major issue.	Vendor presence can be an issue if there are restrictions on the technology or the vendor is absent by choice in a particular region.

TABLE 1 - Issues in Global Development¹

¹ Adapted from Dubé and Paré (2001)

It should be noted that although there are many similarities, as pointed out in Table 1, there are also many differences between OSS and GVT development. Limitations in the corporate culture that do not exist in the open source development model should be taken into account, including hard deadlines, cost considerations and staffing restrictions. Not all propositions discussed here are practical in every potential business situation; managerial judgment and situational context will often play formative roles in team development.

A COMPARISON OF OSS AND GVT PARADIGMS: SIMILARITIES

The OSS development model has been described as a massively parallel development and debugging methodology (Feller 2000). The differences between this development approach and that of GVT are in terms of venue and scope: mainstream OSS takes place among a self-governed worldwide community of practice, while GVT is inherently proprietary and corporation-specific. The key difference of substance is that OSS participants are able to meet as putative equals and govern themselves within the bounds of the OSS licensing structure and a community-based social hierarchy, while GVT members are inherently laden with hierarchical structure imposed by corporate governance. A comparison of key features of each approach, presented in Table 2, demonstrates the similarity in form and function between the two approaches.

As can be seen, scalability is similar between OSS and GVT, distribution is global, communication modalities are similar, and motivations are comparable in regard to cost reduction and acceleration of development. That OSS focuses on user problems and GVT focuses on business problems is easily reconcilable; a more user-centric orientation to business problem solving would be beneficial in development. The primary difference related to governance is an issue for consideration: leadership being centralized is not quite the same thing as reporting relationships being enforced in hierarchical forms in the corporation.

Factor	Open Source Model	Global Virtual Team
Project Sizes	Miniscule to large scale	Medium to large scale
Reason for Initiation	Solve a particular user problem	Solve a particular business problem
Team distribution	Globally	Globally
Team communication	Phone, Internet CMC, video and audio conferencing	Phone, Internet CMC, video and audio conferencing
Team coordination	Versioning software and Internet Support Sites; email	Versioning software, collaborative communications applications and group decision support applications
Motivation	Shared cost and risk in development of robust code on an accelerated schedule.	Reduced cost in development of robust code on an accelerated schedule.
Actors	Professional programmers, highly motivated hobbyists and enthusiasts.	Professional programmers
Leadership	Centralized	Centralized
Management	Self-governance, participative	Hierarchical, reporting relationships

TABLE 2 - Similarities between OSS and GVT

While the similarities of the OSS and GVT factors are striking, the differences in the approach to development are what can separate the two. The main issue in comparison is that OSS community members meet as peers, while GVT members often meet as superiors and subordinates. The secondary issue is that the corporate atmosphere lends itself to the structured development approach that has been traditionally practiced by businesses, while the OSS approach is free form and collaborative by both function and necessity (Herbsleb, et. al. 2001). The thought presented here is that a methodology that approaches the communication, coordination and management of development in a manner that is totally different from traditional software development would alleviate some of the problems of trying to use traditional development methodologies in conjunction with global virtual teams. The Open Source Software development model has had a different approach to the management of software development and the communication and coordination that are required to complete a development effort, from its inception.

LEVERAGING OSS METHODS IN A GLOBAL ORGANIZATION

Virtual team operation by companies is considered to be taxing on management oversight (Harvey et al. 2005), and structured relationships such as sub-groups that tend to arise in corporate settings can have the potential to impact the effectiveness of GVT operation (Panteli and Davison 2005). A flat hierarchy is considered desirable in the governance of virtual teams (Maznevski and Chudoba 2000), which is comparable to the OSS governance approach of a community of peers. The participating governance style of open source software development, with widespread code availability and the communal structure of the development team, results in reduced cycle time for development and debugging in the OSS model, while company-enforced managerial roles and rules often act to restrict efficient operation of virtual teams in the manner envisioned here. A “peer” approach to GVT operation has been suggested as a way to enhance necessary levels of trust and collaboration in team operation (Edwards and Sridhar 2005), and managerial effort in virtual teams ought to be directed at managing conflict rather than attempting to manage people, on the presumption that collaborative and communal structures are inherently more motivating than traditional hierarchical reporting relationships (Paul et al. 2004).

In OSS, participants are self-managed, motivated by the good of the community. Operating as peers, OSS developers manage the development processes collaboratively and participatively, and this echoes suggestions found in the management literature for the improvement of GVT operation (e.g., Goodbody 2005). The challenge of GVT lies in its inherently cross-cultural nature (Paul et al. 2004), and cultures and management style can sometimes clash in structuring and guiding teams (Dubé and Paré 2001). GVT members collaborate better when corporate managers facilitate joint planning and task definition with the group, (Kerber and Buono 2004); this gives rise to levels of trust and reciprocity very like what are achieved in OSS development (e.g., Zakaria et al. 2004). This can lead to an “employee-centric” approach to team governance which can alleviate many of the motivational problems associated with agency-theoretic relationships in business (Harvey et al. 2004). Governance forms that may work well in GVT operation include normative influence styles, which involve unwritten rules that are respected by the community (Saunders et al. 2004), very much like the self-governance style practiced in OSS.

The particular characteristics of OSS methods and the benefits that they would provide in alleviating some of the issues with the global distribution of a traditional software methodology can be seen in Table 3, which presents a taxonomy of OSS development methods adopted for corporate GVT deployment. As we can see from this taxonomy of development methods, the adoption of the OSS model provides many mechanisms that help alleviate a large percentage of the issues identified in the implementation of global virtual teams.

Open Source Characteristic	GVT Issues Addressed
Online Tool Sets for Coordination	Standards enforcement and execution
Online Communication Tools	Reliability of communication channels and elimination of language barriers (through standardization of minimums for implementation of tool set) Language barriers
Self Selection of Resources	Proficiency levels of resources
Parallel development and debugging	Time compression of traditional schedules.
Communal Organization of Developer	Cultural differences
Community Documentation and Agreement	Cultural differences in expectation behavior and resolution of conflict. Standards enforcement and execution
Release Early and Often	Standards enforcement and execution as well as product quality (through peer review and reputation)

TABLE 3 - Taxonomy of Open Source Solutions for GVT Operation

It is important to note that online toolsets are specified as part of the taxonomy. OSS developers who need to solve a specific coding problem will often access the SourceForge site in the OSS community (<http://www.sourceforge.net>). This OSS portal currently has over 1.1 million registered users and 110,000 registered projects, and visitors can use site-supplied coding tools, source control mechanisms and communication tools to assign labor resources and communicate progress, issues and needs over the life of the problem-solving project. As noted in the taxonomy, online communication can solve many of the cross-

cultural problems that corporate GVTs encounter, by providing a single source of access and communication across a development group. The process of allowing team members to meet at an online collaborative site, and self-selected project contributions can ameliorate skill and resources disparities that may trouble GVT members across global distances, while the single point of community contact and documentation enhances both conflict resolution and standard enforcement in development projects that are performed in team settings.

It can be seen that there could be distinct benefits from adopting the OSS methodology for development in environments where global virtual teams are being employed. This potential benefit has to do with modifying organizational culture to match the collaborative development culture practiced by the OSS community. As part of the shift, team performance will be less characterized by “Brooks Law, ” which specifies that complexity and errors will increase as the number of people involved in the project increase, and more characterized by the OSS-style “Linus’ Law,” which posits that quality is the result of massive collaboration (Neus and Scherf 2005). This sort of organizational cultural and philosophical change is at the heart of the shift to OSS methods and models.

Early reports from industry are positive and suggest that OSS methods can help with distributed global development teams. Some of the OSS-style innovations that can improve traditional GVT governance styles include switching to specialized Internet-based tool sets that are shared across a development group, and which promote the online, communal, parallel nature of the OSS methodology. Other changes have more to do with corporate governance than with development methodology, and the governance ideas that we can borrow from OSS will help to ease some of the difficulties faced in traditionally-managed GVT operation. These governance innovations can be summarized into three basic areas:

Parallel, self-selected, rapid development – By using parallel development in a self-selected developer environment, we can increase the efficiency of the application of resources to a problem. We need to be sure to maintain the appropriate level and balance of resources in the pool of participants, but self-selection will insure that there is adequate expertise applied to development tasks. It has been shown that community reputation and participation help to prevent the misapplication of resources. Since OSS-style development is rapid (release early and often) and performed in parallel, we can achieve the desired results in the desired time frame with a possible increase in quality arising from increased self-governance.

The “Bazaar” Attitude - It is important to note the necessity for cultural change as well as structural change. The communal nature of the OSS model can be seen to be at odds with the hierarchical structure of the traditional software development life cycle and traditional corporate management styles. This “cathedral and the bazaar” difference (e.g., Raymond 2000) is a cultural and philosophical change that development organizations must adopt and embrace to make this proposed change in development paradigms work. Collaboration and cooperation are essential to the OSS model, and a sense of community and participation is required to achieve many of the beneficial results accruing to the OSS style of development.

New Tools that support distributed development – Though not typically a governance issue, management controls the provision of tools in industry. Online, Internet-based tools like those employed by the Open Software Initiative and the Free Software Foundation, and associated Internet resources such as those found in the sourceforge.net suite are essential to enable and facilitate the OSS model. As was the case with IBM, management can choose to provide innovative new tool sets to enhance the collaborative process; at the least, management can permit and encourage the use of existing tool sets that are available for examination and experimentation in the OSS community. These tools greatly enhance the “self-management” process in teams, by improving communications, coordinating developers, tracking projects, specifying and enforcing standards, and promoting the highly beneficial collaborative community atmosphere. Standardized tools and sites also help to alleviate some of the technology incompatibilities that can occur under the traditional model of distributed development.

CONCLUSION

Although we only provide an overview of the full range of processes involved in software development in both the OSS and PSS arenas, we believe that there is something to be learned here. It is true that all development projects and teams are not created equal, however most have the same control structures and needs. OSS and GVT teams share many of these needs. As potentially beneficial and successful as the OSS approach to software development may be, and there are certainly instances of failed projects developed through the Sourceforge resource to consider as possible counterpoint to our enthusiastic promotion of the OSS model for development. Even so, the OSS model remains largely a successful global virtual team strategy; it’s just not extensively practiced by companies for commercial development, as yet. The able capabilities of virtual teams cooperating in the development of the Linux Kernel and thousands of open source application projects around the world, connected by computer mediated communications, is a real-world example of how innovative firms can apply virtual team structures in support of global development projects, to their competitive advantage in the

marketplace. While we would caution against the blind wholesale adoption of open source methods, we suggest investigation and potential adoption of OSS-style methods, tools and practices as a means of maximizing the effectiveness of global development teams operating in virtual structures.

REFERENCES

1. Booch, G. (2001) Developing the future, *Communications of the ACM*, 44, 3, 118-121.
2. Cubranic, D., and Booth, K. (1999) Coordinating open source software development. *Proceedings of IEEE Eighth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*.
3. Dempsey, B.J., Weiss, D., Jones, P. and Greenberg, J. (2002) Who is an open source software developer? *Communications of the ACM*, 45, 2, 67-72.
4. Dubé, L. and Paré, G. (2001) Global virtual teams, *Communications of the ACM*, 44, 12, 71-73.
5. Edwards, H.K. and Sridhar, V. (2005). Analysis of software requirements engineering exercises in a global virtual team setup. *Journal of Global Information Management*, 13,2, 21-41.
6. Feller, J. and Fitzgerald, B. (2000) A framework analysis of the open source software development paradigm, *Proceedings of the 21st International Conference on Information Systems*, Brisbane, Queensland, Australia.
7. Goodbody, J. (2005) Critical success factors for global virtual teams, *Strategic Communication Management*, 9,2, 18-21.
8. Harvey, M, Novicevic, M.M. and Garrison, G. (2005) Global virtual teams: A human resource capital architecture, *International Journal of Human Resource Management*, 16,9, 1583-1599.
9. Harvey, M, Novicevic, M.M. and Garrison, G. (2004) Challenges to staffing global virtual teams, *Human Resource Management Review*, 14,3, 275-294.
10. Herbsleb, J.D., Mockus, A., Finholt, T. and Grinter, R.E. (2001) An empirical study of global software development: Distance and speed, *Proceedings of the 23rd International Conference on Software Engineering*, Toronto, Ontario, Canada.
11. Kayworth, T.R. and Leidner, D.E. (2001) Leadership effectiveness in global virtual teams, *Journal of Management Information Systems*, 18,3, 34-40.
12. Kerber, K.W. and Buono, A.F. (2004) Leadership challenges in global virtual teams: Lessons from the field, *SAM Advanced Management Journal*, 69,4, 4-10.
13. Kshetri, N. (2004) Economics of Linux adoption in developing countries, *IEEE Software*, 21,1, 74-81.
14. Maznevski, M.L., and Chudoba, K.M. (2000) Bridging space over time: Global virtual team dynamics and effectiveness, *Organization Science*, 11, 473-492.
15. Montoya-Weiss, M.M., Massey, A.P. and Song, M. (2001) Getting it together: Temporal coordination and conflict management in global virtual teams, *Academy of Management Journal*, 44,6, 1251-1262.
16. Neus, A. and Scherf, P. (2005) Opening minds: Cultural change with the introduction of open-source collaboration methods, *IBM Systems Journal*, 44, 2, 215-225.
17. O'Reilly, T. (1999) Lessons from open-source software development, *Communications of the ACM*, 42, 4, 32-37.
18. Paul, S., Samarah, I.M., Seetharaman, P. and Mykytyn, P.P., Jr. (2004) An empirical investigation of collaborative conflict management style in group support system-based global virtual teams, *Journal of Management Information Systems*, 21,3, 185-222.
19. Raymond, E. (2000) The cathedral and the bazaar, <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/current/1/30/06>.
20. Saunders, C., Van Slyke, C. and Vogel, D.R. (2004) My time or yours? Managing time visions in global virtual teams, *Academy of Management Executive*, 18,1, 19-31.
21. Schmidt, D.P. (2004) Intellectual property battles in a technological global economy: A just war analysis, *Business Ethics Quarterly*, 14,4, 679.
22. Shen, X. (2005) Developing country perspectives on software: Intellectual property and open source – a case study of Microsoft and Linux in China, *International Journal of IT Standards & Standardization Research*, 3,1, 21-43.
23. Von Krogh, G. (2003) Open-source software development, *Sloan Management Review*, 44, 3, 14-18.

24. Weber, S. and Bussell, J. (2005) Will information technology reshape the North-South asymmetry of power in the global political economy? *Studies in Comparative International Development*, 40, 2, 62-84.
25. Wu, M. W., and Lin, Y. D. (2001) Open source software development: an overview. *Computer*, 34, 6, 33-38.
26. Zakaria, N., Amelinckx, A. and Wilemon, D. (2004) Working together or apart? Building knowledge sharing culture for global virtual teams, *Creativity & Innovation Management*, 13,1, 15-29.
27. Zhao, Luyin, and Elbaum, Sebastian (2003) Quality assurance under the open source development model. *The Journal of Systems and Software*, 66, 65-75.