

# Theorization of the open source software phenomenon: a complex adaptive system approach

Raj Agnihotri  
William Paterson University

Murali Shanker  
Kent State University

Prabakar Kothandaraman  
William Paterson University

## ABSTRACT

Academic research on open source software has outlined the functionalities of open source software to a great extent, but a comprehensive theoretical framework comprising different aspects of open source software is still missing. A theoretical understanding of the open source software phenomenon will not only be a significant contribution to the academia but also to the businesses looking for more effective and innovative approaches for software development. To address this issue, first, the current research reviews and analyzes various theoretical frameworks, their applicability in the open source software context, their contributions, and their limitations. Secondly, the structure of open source software projects through *complex adaptive systems* is explored. The current research contributes to the existing literature by providing a detailed discussion of various theoretical perspectives on open source software, and offers a theorization of open source software based on complex adaptive systems.

Keywords: Open source software, information system, software development, open source community.

## INTRODUCTION

The amazing success of open source software community is increasingly receiving attention by industry and academia alike. Following this movement, businesses are getting involved in open source communities and offering support ranging from ‘participation in existing projects’ to the ‘release of new software’ licensed in open source (Lundell et al., 2010). Academia’s interest in the field of OSS has resulted in a special call for research from multiple publication sources (e.g., *Information System Journal* in 2001 and 2002; *Research Policy* in 2003; *Management Science* in 2006). Researchers have done a tremendous amount of work investigating, analyzing, and exploring this unique phenomenon; however, several issues, such as the theorization of OSS model, require more in-depth research (Fitzgerald and Feller, 2001; von Krogh and von Hippel, 2006).

Past literature definitely enhances our understanding of OSS development; however, the unique topological and evolutionary nature of OSS makes it difficult to explain this phenomenon accurately. OSS projects represent extraordinary organizational characteristics (e.g., self-organization, self-regulation, no ownership structure) (Garzarelli, 2004), and OSS communities contain unique and complex features such as open or indeterministic boundary structure, evolving relationships among participants, and presence of feedback loops. Such attributes reflect the complex nature of OSS, yet present a great opportunity to research and theorize the structure of this emerging phenomenon. In pursuit of a comprehensive theoretical framework of OSS, the first critical step is to exhaustively review the current state of OSS conceptualization efforts. To initiate the process, within this paper, a review of various theoretical frameworks is presented. Also, considering OSS projects as evolving and self-governing systems with an adaptive social architecture, the current research discusses and outlines the dynamic structure of the OSS projects by taking a complex adaptive system approach. This effort will offer tremendous assistance to those who attempt to apply the beneficial principles of OSS practices to other areas and systems.

## THEORETICAL DEVELOPMENTS

There are multiple approaches to frame OSS development in contemporary practice and scholarship. Coupled with this, there are many different frameworks and various methodological approaches to understand OSS project structure and operations, which have combined to yield a partitioned and fragmented literature. The current research’s objective is to comprehend the theoretical structure of OSS projects within a common realm. In the following sections, different perspectives and theoretical lenses that are used over time to understand the structure and functionality of OSS are discussed.

### Motivation Perspective

Those factors which motivate developers to participate in OSS projects have now long been a topic of research. Some researchers argue that participants are motivated due to internal factors that do not include monetary rewards, such as challenge seeking, self-learning, and sense of giving back to society (Shah, 2006; Hertel et al., 2003). Others theorize that external factors act as the primary motivational force (Lakhani and von Hippel, 2003). These external motivational factors include developers’ personal software needs, professional concerns, and

status building in the field. As per theories of motivation, intrinsic and extrinsic motivations have different mechanisms and effects, and therefore, should be treated differently (Amabile et al., 1994). These propositions are also aligned with past research that supports the differentiation between intrinsic and extrinsic motivation. For example, an intrinsically motivated individual would perform an activity to directly satisfy his/her own needs, where as an extrinsically motivated individual would be geared towards satisfying needs indirectly (Deci, 1975).

Applying to the OSS context, internal factors that are grounded in an individual's psychology and external factors that emerge from the environment should be studied simultaneously to identify the potential factors that motivate developers to participate in OSS projects. For example, if an OSS project is realized solely through developers' intrinsic motivation, it will not become a functional public product as users possess heterogeneous needs in several contexts. This will become a limitation in comparison to commercial projects' approach, and will also create a dilemma for researchers. If individual needs are the only driving force, then in a situation where users' community is different from developers' community, an OSS project will become disjointed because developers' motivation to participate will not necessarily be aligned with users' needs. For example, Franke and von Hippel (2003) report that heterogeneity is prevalent when it comes to the security needs of users of Apache security software. Due to this very reason, many users are not satisfied with the standard functionality of Apache. Notably, being an open source product, Apache does offer a customizability option and innovation toolkits to its users, but still this problem cannot be ignored completely.

### **Social Identity Theory**

Social identity theory provides a basis for an identity-based motivation model in which the contents of one's self-concept are based on social roles or group membership. The self-concept (e.g., ideal self) is embedded in a social identity and can be attained by self-regulation (Higgins, 1996; Abrams and Hogg, 1999). Self-regulation is the capacity to coordinate neural, cognitive, affective, and behavioral processes so that one can plan, sustain, and sequence actions to attain goals (Oyserman, 2007). It involves the directing of energy, effort, and motivation towards a goal, as well as the strategies required to achieve that goal. Moreover, self-regulatory capacity is described as a motivational resource that can influence the process of pursuing one's goals (Oyserman, 2007). Based on sociology research, Bagozzi and Dholakia (2006) suggested that "with ongoing participation, many members [of OSS community] form meaningful interpersonal relationships within the community and identify with the community" (p. 1105). Social identification within the communities has been considered a critical reason as to why individuals contribute to OSS projects (Hars and Ou, 2002).

Researchers remain focused on the SIT viewpoint that group identification drives programmers' behaviors to perform even routine tasks in OSS projects, yet fail to point out that motivation also originates from practical motives such as improving one's own software, getting recognized among peers, and satisfying the need for fame and esteem. For example, programmers receive feedback from the community that provides them with self-reinforcement. Also, considering the fact that OSS members mainly operate from different places and coordinate their work through mostly electronic media (Hertel et al., 2003), the community resides in a nearly 'virtual domain'. Unlike a regular organization where individuals' co-presence within the same physical space helps create emotional closeness that is the basis for human relationships (Leamer and Storper, 2001), members in an OSS community are tied

together through invisible threads such as common goals. One can have reasonable doubts regarding the generalization of identification arguments within an OSS context.

### **Public Goods Theory**

Importantly, successful outcomes of OSS have demonstrated that even in the absence of ownership rights or authorities involvement, public goods can be efficiently offered (Bessen, 2005). Public goods theory or public expenditure theory (Samuelson, 1955) classifies public and private goods, and also outlines the mechanism that involves people and their motivation to contribute to public goods. The primary focus of this theory is on 'joint consumption' and 'non-excludability'. It characterizes a public consumption good as a good that is "provided for each person to enjoy or not, according to his tastes", and it could be "varied in total quantity...and each man's consumption of it" (Samuelson, 1955; p. 352). Later, Marwell and Oliver (1993) suggested that for the provision of public goods, often, a *critical mass* is needed and there are some key variables that influence this process such as group size, contributors' interdependence, homogeneity or heterogeneity of the participants' level of interest.

In the context of OSS, it is very difficult to recognize the critical mass (i.e., a stage where a public good has enough attention and resources to be self-sufficient) because each project has different and unique objectives, and success cannot be measured by one scale (Christley et al., 2004). Another concern regarding the public goods production is related to the protection of common goods. Addressing this issue, O'Mahony (2003) points out that OSS communities have serious concerns regarding 'public goods' final destiny while providing open access to their work. OSS project members, in fact, actively promote agreement with the open source license terms related to their project. Interestingly, these members may come up with trademarks and logos to make sure that these public goods remain in the commons (O'Mahony, 2003). Moreover, raising concerns regarding the Utopian characteristics of OSS such as collective public good, some researchers have argued that "culture of OSS is highly individualistic and reputation based, and that the opportunities for personal gain, both financial and otherwise, are enormous" (Fitzgerald and Feller, 2001; p. 274).

### **Social Network Perspective**

As per social network theory (SNT), society structure can be modeled as a graph containing nodes and edges; notably, individuals or organizations are characterized as nodes (or social actors) and relationships (or ties) among them are represented as edges of the graph (Jin et al., 2001). In a similar vein, social network analysis is defined as "the disciplined inquiry into the patterning of relations among social actors, as well as the patterning of relationships among actors at different levels of analysis (such as persons and groups)" (Breiger, 2004; p. 505).

Madey and colleagues (2002) model OSS as collaborative social network and suggest that individual developers represent nodes of a graph and cooperative associations on an open source project are collaborative links among these developers. The findings of their study suggested that "the open source movement is not a random graph (i.e., new nodes attach to existing nodes with uniform probabilities), but a graph displaying preferential attachment of new nodes (i.e., some nodes have higher probability of attachment than others)" (Madey et al., 2002; p. 1810). The reason for this non-randomness could be the fact that visibility and attractiveness of projects will differ from each other and these differences will, in turn, influence developers'

participation and activeness. Another issue is related to the evolving interaction patterns within OSS projects. Contrary to the past assumptions that the network plot has a flat structure, the recent research by Long and Siau (2007) suggests a layer structure that implies diminishing group centralization in the long run and a propensity of 'core/periphery' structure in OSS projects.

### **Organization Perspective**

In a commercial set-up, software products are developed through strictly managed processes. The company and software developers are bound together through contracts and regulations. The company management attempts to organize the project from top to bottom and cost reduction is considered a top priority. Management practices vigilance to keep programmers on track so that deadlines can be met along with quality maintenance. OSS projects differ vitally from such organizational practices. Considering two key attributes of open source economic organization (i.e., self-organizing and self-regulating), Garzarelli (2004) attempted to explain the working of this system. This approach is aligned with the open source viewpoint of "self-correcting spontaneous" and "more elaborate and efficient than any amount of central planning could have achieved" as suggested by Raymond (2001, p. 52).

Garzarelli (2004) utilized the organization theory of professions to explain the mechanism of OSS. Also, based on Savage's (1994) work, Garzarelli (2004) outlined several organizational implications such as professionals are "independent yet interact in a coordinated and fertile fashion", they are "decentralized networks", and they can be categorized as "self-organizing organizations" (p. 9). However, recent research highlights the emergence of a central structure within a decentralized network. For example, Von Krogh and colleagues (2003), in a clinical study of Freenet, an OSS project for peer-to-peer computing, suggest that the project's organizational functionality is governed by a community of programmers and only they can assign code to the authorized version of the software. Developers seeking to get entry into this club follow some *virtual guidelines*, such as putting in significantly more complex and technically difficult work in comparison to regular participants. Moreover, these aspirants offer *donations* in terms of sophisticated software modules and attributes that have capabilities to strengthen the architecture and smooth the functionality of software.

### **OPEN SOURCE PROJECTS AS COMPLEX ADAPTIVE SYSTEMS**

Past attempts to theorize the open source phenomenon provide us with solid groundwork to better understand this unique phenomenon, yet there are numerous aspects that are still uncovered and need immediate attention. In an OSS system, there is not a single element that directs the collective behavior of the system and in fact, it is a dynamical system where individual's behaviors are undeterministic and unreproducible. Organizations can be categorized as social structures but, unlike a natural social network, individuals in an organization collectively represent a hierarchal arrangement and follow a set of rules (Carley, 1995). However, some organizational science researchers have argued that participants of organizational social network are not always governed by rules and, moreover, these rules cannot be considered as antecedents to participants' behaviors (e.g., March and Olsen, 1976; Scott, 1992). Building on this theoretical groundwork, Anderson (1999) proposed that "agents in CAS [Complex Adaptive System] models need not be the prisoners of a fixed set of rules" (p. 221). A

social order will evolve through interactions between participants of this social arrangement and the behaviors of participants will be guided by their own schemata. Anderson (1999) defines schemata as “a cognitive structure that determines what action the agent takes at time  $t$ , given its perception of the environment” (p. 221).

Studying social systems through cybernetics, Scott (2001) suggested that dynamical and self-organizing systems survive by amending themselves as per their surroundings. Notably, during this process, energy gets imported from external environment into the self-organized system. Aligned with this notion, CAS needs a continuous inflow of energy from various resources such as participants, partners, and customers. Importantly, recruitment of these energy sources can be made possible “by motivating stakeholders” (Anderson, 1999; p. 222). A self-organized social entity will sustain only till its members offer their skills and abilities to keep the system alive. Within the theoretical realm of self-organizing networks, members of an organization are viewed as social workers who ordain themselves and construct their environment (Weick, 1979). Proponents of the concept of bounded rationality (e.g., March and Simon, 1958) have suggested that economic agents not always make rational decisions because of the situational complexity and their inability to predict the usefulness of their actions for the system.

Aligned with this notion, CAS theory suggests that agents attempt to “optimize their own fitness, not that of the organization”, and in doing so, they “coevolve with one another” (Anderson, 1999; p. 223). Proposing arguments that support the operational similarity between social and physical systems is not a new phenomenon in academics (e.g., von Bertalanffy, 1968). Interestingly, several natural science perspectives (e.g., biological organisms, electromagnetism, and so on) have been utilized to explore and understand social phenomena such as self-organizing systems and enduring organizational mechanisms (Scott, 2001).

Organizational scientists, especially those who focus on social relationships between organizations and their agents, build their studies on the basis of interlocking concept. For example, Stearns and Mizuchi (1986) suggested that organizations can utilize interlock notion to organize or lead their environments (i.e., interorganizational approach), and members of a dominant social class can employ interlock perspective to devise and synchronize its common benefits (i.e., interclass approach). Additionally, an agent’s (i.e., individuals or organizations) reputation and influence within a class will depend on his/her place within a social network (Palmer, 1983). This logic of forming, breaking, and reconstituting social relationships provides CAS models with a critical attribute. As Anderson (1999) postulated, CAS models will “evolve over time through the entry, exit, and transformation of agents” (p. 220) and, moreover, relationships among agents will change over time, “shifting the pattern of interconnections, the strength of each connection, and its sign or functional form” (p. 220). Aligned with CAS theories, OSS models not only provide agents with schemata, linkages, and adaptive behavior, but also allow them to evolve over time.

## **The Community**

In open source projects, participants construct a community glued together by their common interests. The open society feeling develops a virtual bond among community members. Such participants are not assigned roles by any central authority. Instead, they pick duties related to their personal interests. Making decisions and providing leadership are some of the most atypical issues within open source community. In the case of any conflict that arises, behavioral

standards and trends are the means by which to resolve it. Exploring OSS communities, researchers, such as Xu and Madey (2004) and Ye and Kishida (2003), classify such communities based on the different roles participants play in a project. Members of an open source community do not necessarily perform a fixed role. The realization of a role is completely dependent on the involvement and contribution of any individual member.

Each member of open source community internalizes one basic principle conveying the message that there is no authority or management structure to assign roles and responsibilities. Members are very well aware of the fact that there is nothing that can stop them to leave the community at any given time. With different motivations and aspirations, various participants transform their roles within a project. Different from traditional practices, open source projects are in fact a result of active users' participation. Thus, the level of interdependence between users and developers is quite high. Open source provides developers and users with a platform where they can intermingle, interact, learn, and change together. The developers' quest to recognize the actual problems and to identify novel ideas actually augments the intensity of the interdependence. It is argued that interdependence among resources is one of the key elements in the evolution of open source projects (Christley et al., 2004). Notably, this interdependence influences the flow of information between and within different user and developer communities.

## **The Process**

The functionality of open source resembles a distributed software development process. A typical OSS project is comprised of hundreds of central developers who do most of the coding and thousands of peripheral members who contribute in a more indirect and irregular manner. Exploring the work mechanism of open source, Weber (2004) reported that the traditional software development process is based on the 'engineering archetype' while the OSS development process is aligned with the 'evolution archetype'.

The primary motive of open source is not to make a profit, but to resolve a problem and to come up with the best solution. Although developers are aware of the problem definition, they try distinct courses of action to reach the potential solution constructing an environment of parallel problem solving. During this search for resolution and for optimizing their own capabilities, developers coevolve with each other. Any alteration, enhancement, or development made to the software not only evolves the software product itself, but also redefines the participants' role and the overall social dynamics. Users' feedback and participation provides developers with specific and detailed recommendations on the product prototype. In turn, developers modify and improve the software based on users' feedback and also based on their own vision, skills, and capabilities.

This evolving software product is constantly available for open use. Notably, it is very possible that the resulting product is beyond the vision of users, and hence, it introduces them to a unique and radically new application. Importantly, in this process both users and developers are transformed. Over time, new members join the community as they see the solution of their problems in this emerging software product. Some existing members may leave the community as they see that the change of course does not fulfill their goals and this new direction may not solve their problems. Also, existing members may migrate from their current role upstream or downstream, depending on their involvement and contribution to the project.

## **The Outcome**

The software product of an open source project is utterly distributable and offered for public use, without restrictions and concerns regarding piracy. Moreover, users are encouraged to make as many copies as they like and to distribute to others. Interestingly, promotional activities are to persuade people to participate even if they are not programmers. One can join a typical OSS product's mailing list and contribute in other ways, such as to simply use the software product and report any bugs, if found. Additionally, help can be offered in the areas of documentation, site maintenance, and software promotion. Citing the advantages of OSS, one can argue that acceptance of such software products will make code readily amendable and adjustable to meet an individual's specific needs and requirements. Some key benefits will include the trimmed cost for development, alterations, and licensing of the software, and a superior product in result due to more public access. In case of deficiencies, coming up with the modifications will be faster. There could be wide availability of support services and alternatives. Furthermore, eradication of licensing requirements will reduce many legal issues and costs, as well as the costs and the workload coupled with tracking of software copies and usage. Success of many OSS products such as Linux, Firefox, etc. has demonstrated the fact that OSS is growing and the people are opening themselves to this new idea. However, the challenges faced by OSS movement presents us with issues that need further thought process. Some organizational systems are very rigid and unique in terms of operation and approach, and therefore, may not be perfectly fit with the model of open source.

Critiquing the outcomes of open source movement, opponents argue that in the long run, open source practices could have adverse effects on quality research by purging the rewards, especially monetary, associated with it. Prevalence of OSS would create an environment where a proprietary software vendor's expertise will no longer be compensated (von Hippel and von Krogh, 2003). Making the case worse, it would have a negative impact on intellectual property rights. Commercial software developers claim that OSS is very limited in scope and particular in application. In another point, the opponents put forward problems regarding security by claiming that mischievous elements may use the unrestricted access to documentation and open reporting of the presence of deficiencies to take advantage of the situation before the open source community can rectify the weakness. However, nobody can deny the security issues and breaches in proprietary software systems too. Even though organizations spend a lot of money to affirm the security, their software is not at all perfectly secure.

## **CONCLUSION**

The triumph of open source movement depends not only on getting the right people to participate but also on how and what these people contribute to a collective cause. It opens a path of learning for software organizations struggling to cope up with the pace of information flow and the abundance of knowledge. Theorization of open source not only needs collaboration between disciplines, but also a platform sharing knowledge across disciplines. It should be an interdisciplinary study of the complex and evolving system, especially uncoupled communication processes, self-control mechanisms, and feedback principles. The structured representation of the OSS phenomenon will also be a great help to businesses, as to succeed in today's competitive and fast changing market environment it will be necessary to think outside the box and adapt new strategies for survival.



## REFERENCES

- Abrams, D. and Hogg, M.A. (1999), *Social Identity and Social Cognition*, Blackwell, Oxford.
- Amabile, T. M., Hill, K.G., Hennessey, B.A. and Tighe, E.M. (1994), "The work preference inventory: Assessing intrinsic and extrinsic motivational orientations", *Journal of Personality and Social Psychology*, Vol. 66 No. 5, pp. 950-967.
- Anderson, P. (1999), "Complexity theory and organization science", *Organization Science*, Vol. 10 No. 3, pp. 216-232.
- Banerjee, A. (1992), "A simple model of herd behavior", *Quarterly Journal of Economics*, Vol. 107 No. 3, pp. 797-818.
- Bagozzi, R.P. and Dholakia, U.M. (2006), "Open Source Software User Communities: A Study of Participation in Linux User Groups", *Management Science*, Vol. 52, pp. 1099- 1115.
- Bessen, J.E. (2005), "Open source software: Free provision of complex public goods", Available at SSRN: <http://ssrn.com/abstract=588763> or DOI: [10.2139/ssrn.588763](https://doi.org/10.2139/ssrn.588763) .
- Breiger, R.L. (2004) "The analysis of social networks", in Melissa Hardy, M. and Bryman, A. (Eds.), *Handbook of Data Analysis*, Sage Publications, London, pp. 505-526.
- Carley, K.M. (1995), "Computational and mathematical organization theory: Perspective and directions", *Computational and Mathematical Organization Theory*, Vol. , pp. 39-56.
- Christley, S., Xu, J., Gao, Y. and Madey, G. (2004), "Public goods theory of the open source development community", *Agent2004*, October, Chicago, IL.
- Deci, E. (1975), *Intrinsic Motivation*, Plenum Press, New York, NY.
- Fitzgerald, B. and Feller, J. (2001), "Open source software: Investigating the software engineering, psychosocial and economic issues", *Information Systems Journal*, Vol. 11, No. 4, pp 273-276.
- Franke, N. and von Hippel, E. (2003), "Satisfying heterogeneous user needs via innovation toolkits: the case of Apache security software", *Research Policy*, 32, pp. 1199- 1215.
- Garzarelli, G. (2004), "Open source software and the economics of organization", in Birner, J. and Garrouste, P. (Eds.), *Markets, Information and Communication*, Routledge, New York, NY, pp. 47-62.
- Hars, A. and Ou, S.S. (2002), "Working for free? Motivations for participating in open-source projects", *International Journal of Electronic Commerce*, Vol. 6 No.3, pp. 25-39.
- Hertel, G., Niedner, S. and Herrmann, S. (2003), "Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel", *Research Policy*, Vol. 32, No.7, pp. 1159-1177.
- Higgins, E.T. (1996), "The self-digest: Self-knowledge serving self-regulatory functions", *Journal of Personality and Social Psychology*, Vol. 71, pp. 1062-1083.
- Jin E. M., Girvan, M. and Newman M. (2001), "The structure of growing social networks", *Physical Review E.*, Vol. 64 No. 4.
- Lakhani, K.R. and von Hippel, E. (2003), "How open source software works: 'Free' user-to-user assistance", *Research Policy*, Vol. 32 No. 6, pp. 923-943.
- Leamer, E. and Storper, M. (2001), "The economic geography of the internet age", *Working Paper 8450*, Washington: National Bureau of Economic Research.
- Long, Y. and Siau, K. (2007), "Social network structures in open source software development teams", *Journal of Database Management*, Vol. 18 No. 2, pp. 25-40.
- Lundell, B., Lings, B. and Lindqvist, E. (2010), "Open source in Swedish companies: where are we?" *Information Systems Journal*.

- Madey, G., Freeh, V. and Tynan, R. (2002), "The open source software development phenomenon: An analysis based on social network theory", *Americas Conference on Information Systems (AMCIS2002)*, Dallas. pp. 1806-1813.
- March, J.G. and Olsen, J.P. (1976), *Ambiguity and Choice in Organizations*, Bergen, Norway.
- March, J.G. and Simon, H.A. (1958), *Organizations*, John Wiley, New York, NY
- Marwell, G. and Oliver, P. (1993), *The Critical Mass in Collective Action: A Micro-Social Theory*, Cambridge University Press, Cambridge, UK.
- O'Mahony, S. (2003), "Guarding the commons: How community managed software projects protect their work", *Research Policy*, Vol. 32 No. 7, pp. 1179-1198.
- Oyserman, D. (2007), *In Social Psychology: Handbook of Basic Principles*, Kruglanski, A.W. and Higgins, E.T. (Eds.), Guilford Press, New York, NY.
- Palmer, D. (1983), "Broken ties: Interlocking directorates and intercorporate coordination", *Administrative Science Quarterly*, Vol. 28, pp. 40-56.
- Raymond, E.S. (2001), *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, O'Reilly, Sebastopol, CA.
- Samuelson, P.A. (1955), "Diagrammatic exposition of a theory of public expenditure", *Review of Economics and Statistics*, Vol. 37, pp. 350-356.
- Savage, D.A. (1994), "The Professions in theory and history: the case of pharmacy", *Business and Economic History*, Vol. 23 No. 2, pp. 129-60.
- Scott, R.W. (1992). *Organizations: Rational, Natural and Open Systems*, Prentice-Hall, Englewood Cliffs, NJ.
- Scott, B. (2001), "Cybernetics and the social sciences", *Systems Research and Behavioral Science*, Vol. 18 No. 5, pp. 411-420.
- Shah, S.K. (2006), "Motivation, governance, and the viability of hybrid forms in open source software development", *Management Science*, Vol. 52, No. 7, pp. 1000-1014.
- Stearns, L.B. and Mizruchi, M. (1986), "Broken-tie reconstitution and the functions of interorganization interlocks: A reexamination", *Administrative Science Quarterly*, Vol. 31, pp. 522-539.
- von Bertalanffy, L. (1968), *General System Theory: Foundations, Development, Applications*. G. Braziller, New York, NY.
- von Hippel, E. and von Krogh, G. (2003), "Open source software and the 'private-collective' innovation model: Issues for organization science", *Organization Science*, Vol. 14 No. 2, pp. 209-223.
- von Krogh, G. and von Hippel, E. (2006), "The promise of research on open source software", *Management Science*, Vol. 52 No. 7, pp. 975-983.
- Von Krogh, G., Spaeth, S. and Lakhani, K.R. (2003), "Community, joining, and specialization in open source software innovation: A case study", *Research Policy*, 32, pp. 1217-1241.
- Weber, S. (2004), *The Success of Open Source*, Harvard University Press, Cambridge, MA.
- Weick, K.E. (1979), *The Social Psychology of Organizing*, Addison- Wesley, Reading, MA.
- Xu, J. and Madey, G. (2004), "Exploration of the open source software community", in *NAACSOS Conference, 2004*, Pittsburgh.
- Ye, Y. and Kishida, K. (2003), "Toward an understanding of the motivation of open source software developers", *Proceedings of 2003 International Conference on Software Engineering (ICSE2003)*, Portland, Oregon, pp. 419-429.