

Foreword

This book provides an edited collection of original papers that examine how open source software (OSS) represents a new way to think about how organizations learn, or how they can support learning practices using the ways and means of OSS development. OSS, in many ways, is a revolutionary approach to the development of new IT applications, and thus, potentially a disruptive means for stimulating both organizational and IT innovation, as well as new ways and means for acquiring, representing, and operationalizing new forms of organizational knowledge. This book thus provides a refreshing look at the topic of organizational learning in terms of focal themes that include how OSS communities can support organizational development, how knowledge management systems and capabilities within organizations can benefit from or exploit OSS methods, and how government agencies and educational institutions can benefit from the adoption and assimilation of OSS applications, methods, and communities. We can consider each of these themes in turn, as well as some subthemes that arise across many chapters where these themes are found. Similarly, we can further reflect on what topics related to OSS and organizational learning are not addressed in these few chapters, and thus point to opportunities for further study in this area.

The first theme addresses how communities of interest or communities of practice arise to support the ongoing development and support of successful OSS systems, and how these communities then may expand to support user communities in different application areas. The chapter by Reinhardt and Hemetsberger examines how the KDE OSS project community ensures the ongoing success and continuous development of the multimillion lines of source code that constitute the K Development Environment through the continuous turnover and migration of apprentice contributors and expert software developers. The chapter by Demetriou, Koch, and

Neumann examines how the evolution of the OpenACS framework for developing community-oriented Web sites had to endure a history of organizational transformations and transitions as different commercial interests and investors sought to ensure (or exploit) its commercial success. The chapter by Francq examines how “social browsing” software designed to support knowledge sharing can be constructed from user contributed profiles of their areas of interest, which in turn can be assembled into communities of interest through this software. These three chapters collectively underscore how the ongoing development of OSS depends on the concomitant development of its developer-user community, and conversely, the development of this social community depends on the ongoing development, use, and support of the community of software components and artifacts that collectively form the OSS system within the community. OSS systems and communities can thus be recognized once again as socio-technical interaction networks that cannot be separated into social systems and technical systems whose ongoing development and evolution are independent (Scacchi, 2005).

The second theme examines the development of knowledge management systems and methods that are built with OSS components or built to encourage or exploit sharing and reuse of knowledge through OSS development practices. The chapter by Rech, Ras, and Decker describes the Riki system, which is intended to facilitate the sharing and reuse of knowledge that arises in traditional software engineering projects. The chapter by Hocht and Rech complements the preceding one by showing how the system they design for managing the knowledge artifacts that arise in agile software engineering projects must be human centered in order to enable effective software engineering knowledge sharing. In contrast, the chapter by Butler, Feller, Pope, and Murphy presents a case study that describes their effort to construct and deploy a portable knowledge asset development system that supports the open elicitation and capture of knowledge found in governmental and nongovernmental organizations in Ireland. Each of these three chapters thus stresses the role of the development of OSS systems as a core capability for enabling organizational learning and thus organizational transformation in different organizational settings.

The next theme examines how government agencies can benefit from the adoption and assimilation of OSS systems, methods, and communities. As noted previously, the chapter by Butler and associates examines the development of an OSS system targeted for adoption by government agencies within Ireland. The chapter by Gotze, Herning, Wernberg-Tougaard, and Schmitz provides a scheme for evaluating when and how government organizations should adopt OSS systems for use in providing public or administrative services. Similarly, the chapter by Castilho, Sunye, and associates outlines policies and strategies they have put into practice for encouraging the adoption and use of OSS tools they have developed, and brought together for use, within 2,100 public schools. Together, these chapters help to reveal that government organizations need to learn how to most effectively and efficiently adopt, deploy, use, and sustain OSS systems, and that the knowledge that must be learned and put

into practice differs in many ways from that of traditional IT systems provided by commercial vendors or contractors that support government agencies.

The last theme found in this volume examines how educational institutions can benefit from policies that encourage and guide the adoption and assimilation of OSS applications, methods, and communities. This turns out to be perhaps the dominant theme, as it is the focus of four chapters. The chapter by Damiani, Mezey, Pumilia, and Tamaro (with other contributors) draws attention to the need for educational organizations to adopt an open source organizational model that encourages an open culture for education through knowledge sharing, interoperable learning support systems, and reusable educational system components whose quality can be continually assessed and readily assured. The chapter by Vuorikari and Sarnow examines policies that support initiatives of the European Schoolnet, focusing on the development and exchange of open learning resources. The chapter by Bouras and Nani examines similar issues and concerns in building and accessing learning resources and online courses for distance education. Finally, as noted previously, the chapter by Castilho, Sunye, and associates focuses attention on policies and strategies that support the adoption and deployment of OSS-based educational resources. Overall, much like the situation for government organizations, educational authorities and institutions both require and benefit from policies and strategies that help learn how to effectively to guide their own adoption and deployment of OSS-based educational resources.

Beyond these four major themes, two additional subthemes can be seen across many of the 13 chapters in this volume. These include the first subtheme, that the adoption, deployment, and use of OSS systems both requires and benefits from an understanding of the development practices, processes, and community dynamics found in different OSS projects (Scacchi, Feller, Fitzgerald, Hissam, & Lahkani, 2006). Adopting and using OSS systems seems to require that the people in an organization need to learn how the particular OSS systems they will use are developed, and how, as users, they may need to contribute to the project communities that are developing and sustaining the systems at hand. This need seems to differ in kind from that associated with IT systems provided as proprietary systems from commercial vendors, whereby users are not typically expected to be required to learn about how such systems were developed or why, though perhaps such involvement might prove beneficial. The second subtheme is that a growing number of OSS systems increasingly are expected to serve in the role as “social software”—that is, as software whose intended usage is to help people in different organizational settings to cooperate (share), coordinate (interoperate), or collaborate (work together with open or transparent learning resources). Both of these subthemes point to opportunities for further study into how OSS can support or facilitate organizational learning.

Finally, there are a number of topics that were not addressed in the 13 chapters in this volume. Of course, no single volume can be expected to be comprehensive or exhaustive, especially when the topic of the edited collection of chapters is new and

unexplored, or when prior studies have appeared infrequently in different journals, conferences, or workshops. So by pointing to these additional topics, the intent is merely to help seed future research studies, thereby complementing and building on the contributions appearing in this first volume on OSS and organizational learning.

For example, what roles can (a) OSS development project communities, (b) OSS-based knowledge management systems, (c) social software, or (d) OSS development practices play in encouraging, facilitating, or inhibiting organizational learning? Similarly, how might OSS development practices, processes or community dynamics encourage, facilitate, or inhibit organizational learning in large corporations? How do large OSS project communities that are organized and governed by foundations (e.g., Apache, Mozilla, Eclipse, Gnome) learn how to improve both how they develop and how they support large software systems with globally decentralized, loosely coupled development teams? Next, how do different kinds of organizations learn from the patterns of success and failure in OSS development projects? (Here, it can be noted that the vast majority of OSS projects found in Web portals like SourceForge.net have less than two developers and have never released any software source code running or not. Thus though there are many success stories in the world of OSS development projects, there are many more incomplete or unsuccessful efforts that have so far garnered little research attention or publication). Last, what are the empirically grounded models or theories that account for OSS-based organizational learning, do they offer testable predictions or refutable hypotheses, and how do they compare to prior models/theories of organizational learning that do not assume a central role for OSS systems, development practices, processes, or project communities? As before, even this list of additional topics is by no means complete or exhaustive; instead, it is merely suggestive of the breadth, depth, and intellectual richness of the opportunities that lie ahead for further studies of open source software and organizational learning. Finally, as should be clear from the diversity of topics and application areas addressed in this volume, future studies will likely employ scholarly resources and methods drawn from multiple fields of study. Therefore, this volume constitutes the starting point for the studies that will follow and build from those presented here.

Walt Scacchi

Institute for Software Research

University of California, Irvine

References

- Scacchi, W. (2005). Socio-technical interaction networks in free/open source software development processes. In S. T. Acuña & N. Juristo (Eds.), *Software process modeling* (pp.1-27). New York: Springer Science+Business Media Inc.
- Scacchi, W., Feller, J., Fitzgerald, B., Hissam, S., & Lakhani, K. (2006). Understanding free/open source software development processes. *Software Process: Improvement and Practice*, 11(2), 95-105.

* * *

A volume investigating the underlying mechanisms of open source organizations, and how learning and organizational practices are affected by adopting open source methods, is very timely. Open source is clearly more than just a method for building software. It would be interesting enough if it were just an innovative “mode of production” for the new artifact called software. But, as this volume investigates, the ideas behind open source software, those of intellectual property approaches that guarantee access and not exclusion, and the learning and application of knowledge at distributed nodes that are coordinated but not centrally controlled, point to even more revolutionary changes in the world. These open approaches to work and development contain more than just a perspective on how people could work freely together. They point to ways of realizing more of the creative human spirit and economic efficiencies simultaneously. The investigations into just what open source processes are, and how they can be effective in producing both products and organizational change, help us see this broader context of the open source and free software approaches.

The advent of widely accessible, easily approachable global communications, through the Internet and the Web, has driven the costs of collaboration and co-operation in the construction of software, and other knowledge artifacts, through the floor. It is not by chance that many of the early software contributions in this arena themselves contribute to making communication and collaboration easier, and hence help build the communities that are their very creators. All this has brought together communities of interest and practice that have been busy realizing value for themselves and others, and at the same time creating new ways of value-creation. Simply in the process of going about their business, they have created new ways of doing business. Understanding how this has happened is an urgent task; a task this book provides significant contributions to.

As more parts of the world become essentially digital, as models of pharmaceutical chemicals become central containers of our knowledge about those chemicals, for instance, or buckets of bits hold the movies and music we want to see and hear, or

more and more of our education goes online and advanced education increasingly involves apprenticeships in partially or wholly online communities, more and more arenas of creativity and knowledge become open to the disruptive forces of open source thinking and methods. If effective drugs can result from open source efforts, as effective software has, then we should endeavor to understand, as quickly as we can how these methods can be applied in that industry.

The freedom to modify and redistribute these collective efforts, and the ability of individuals to choose where and how they contribute to a product are at the center of these new approaches. This is more than just a way of distributing work. It is an engine for creating intellectual property that can benefit organizations and the public alike. Learning how open source practices contribute to learning and how organizational evolution results from adopting such practices is a worthwhile theme for such a collection as this. The authors here make a welcome contribution to both our self-understanding and our future modes of working.

Joseph Hardin

Sakai Foundation

School of Information

University of Michigan