9-18-2006

A Research Agenda for Studying Open Source I: A Multi-Level Framework

Fred Niederman
St. Louis University, niederfa@slu.edu

Ashley Davis
University of Georgia

Martina E. Greiner
University of Georgia

Donald Wynn
University of Georgia

Paul T. York
University of Georgia

Follow this and additional works at: https://aisel.aisnet.org/cais

A RESEARCH AGENDA FOR STUDYING OPEN SOURCE I: A MULTI-LEVEL FRAMEWORK

Fred Niederman
Saint Louis University
niederfa@slu.edu

Ashley Davis
Martina E. Greiner
Donald Wynn
Paul T. York
University of Georgia

ABSTRACT

This paper presents a research agenda for studying information systems using open source software A multi-level research model is developed at five discrete levels of analysis: (1) the artifact; (2) the individual; (3) the team, project, and community; (4) the organization; and (5) society. Each level is discussed in terms of key issues within the level. Examples are based on prior research. In a companion paper, [Niederman, et al 2006], we view the agenda through the lens of referent discipline theories.

KEYWORDS: open-source, multi-level theory

I. INTRODUCTION

At the time of writing, Spring 2006, open source products (particularly GNU/Linux and Apache web servers) are widely diffused throughout the world and in significant and constant use. Sourceforge.net hosts more than 100,000 open source development projects at varying levels of development with more than one million registered site users [Sourceforge, 2005] The OpenOffice organization reports that more than 40 million downloads of its software were recorded as of April 2005 [OpenOffice, 2006]. Moreover, organizations, such as Compiere Inc. and SugarCRM, offer open source code and development techniques for ERP (Enterprise Resource Planning) and CRM (Customer Relationship Management) software.

One would expect MIS scholars to investigate the potentially important open source phenomenon in its own right and as a potential influence on the larger information systems domain. Research questions naturally address:

- the variations in the technology as an artifact;

- its development processes;

- the motivations and results of its developers and users; and

- the interaction between the diversity of technology and terms of its use with the economic and social effects upon individuals, organizations, and society at large.

A comprehensive view of open source is necessary to address this wide range of issues. To create sufficient rigor, individual studies typically limit themselves to specific and measurable variables. Such studies can be of great value, but do not provide sufficient breadth to understand the implications of open source.   A relationship that holds true for a particular open source community and how its governance impacts the nature of the artifacts it creates, may not hold true for other types of communities and artifacts. To investigate the domain of open source software in its full richness, such a study needs to be set in the context of many studies that fully explore the range of communities and the range of artifacts. It is through such an approach that the limits, if any, of the effectiveness of the open source approach can be observed. For example, particular sorts of hierarchical communication and decision making in open source communities may work extremely well for infrastructure artifacts, but only modestly well (or even poorly) with enterprise artifacts. Therefore, conclusions based on studying the range of open source variables may look quite different than conclusions from a particular study.

In this paper, we develop a multi-level framework as a lens through which such cumulative results can be observed. We propose examination of the open source domain from five levels:

1. the artifact,

2. the individual,

3. the group/ the project/ the community,

4. the organization, and

5 . the broader societal perspective.

Not all of the levels within such a multi-level view are equivalent in addressing open source issues. Differentiation of open source from traditional software is most clearly seen at the artifact level where studies can contrast artifacts using open source licensing and those that are not. Communities developing software within a clearly open source organization can be distinguished from traditional hierarchical business models. Organizations investing in open source can also be distinguished from those that do not invest. Firms in different industries may realize differing returns from equivalent investments. On the other hand, Issues of organizations, communities, individuals, artifacts and projects include many issues unrelated to open source. Therefore, these variables/terms are used in this paper specifically as they apply to open source software, rather than in reference to all of their many aspects. Referring to the variables simply as organization or individual is done to keep the resulting nomenclature from becoming overly stilted and burdensome.

The two goals of this study are:

1. To view open source research as addressing issues at several levels of analysis. By viewing the field this way, individual studies can be compared and their findings collected to broaden the overall understanding even if their areas of focus overlap only partially.

2. To map key existing research to the proposed framework. We sketch the range of what was already observed about open source software and show where new research can provide extensions of the existing literature.

WHAT IS OPEN SOURCE[1]?

The central tenet of open source software is that the source code is available for anyone who wants to use or modify it. Beyond that broad definition, a continuum of "openness" exists. The variations in licensing serve to define categories of differing amounts of restrictiveness on the use of "open source" software.

The classic scenario for open source software occurs when an individual wants others to share in a relatively large project (more than the individual wants to do alone) primarily because the individual wants to use the software created. The individual posts the project to a website and asks for contributions. If interest is sufficient, a core group of programmers and designers begins serious volunteer work to develop the software. A larger group reviews the output, adding significant patches and a still larger group tests and finds weaknesses in the software that need repair [Mockus, Fielding, and Herbsleb, 2002]. For highly successful projects, such as GNU/Linux, Apache, and Mozilla, the stable software created is released to literally millions of users.

Particular social structures, including communities and a volunteer workforce, are generally viewed as part of open source software; however, the specific nature of the communities and work arrangements show more variation than the stereotypical image would suggest. Krishnamurthy [2002], for example, shows that in the majority of cases open source code was developed and continues to be managed by only a few or even a single developer.

More and more traditional proprietary software companies are releasing (fully or selectively) the source code for otherwise commercial products. Microsoft, for example, reportedly released source code for selected products to selected customers [Cukier, 2005]. However, the consensus among researchers seems to be to use the Open Source Initiative (OSI) [OSI, 2006] definition[2]. This definition effectively means that any software distributed under an OSI approved license is 'open source' and anything distributed under a non-OSI approved license is not open source. This definition, would, for example, exclude Microsoft's shared source initiative from being considered a form of open source.

The term "free software" [Free Software Foundation, 2006] is frequently used in addition to "open source". The emphasis of the Free Software Foundation is on preserving a range of freedoms for the acquisition, use, distribution, and modification of software beyond simply allowing for direct access to source code.

In this paper, we use the term open source to include both philosophical positions.

WHY IS IT IMPORTANT TO DEVELOP AN OPEN SOURCE RESEARCH AGENDA?

While a significant number of research papers investigate aspects of open source, these papers are not necessarily framed in a larger context. Almost none considers how the particular study fits into open source overall. We do not argue that every study needs to address all aspects of open source. Focusing in detail on various open source components exposes much that is hidden about how these components work. However, studying the detail with a background conceptualization of where it fits into a larger picture is also helpful. Outcomes or dependent variables at one level of analysis may be important at another level. For example, studies of quality of code based on performance measures at the artifact level may show increasing value

---

[1] This section is identical to the same section in the companion paper [Niederman et al. 2006].

[2] OSI defines open source on its website as: When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing [OSI, 2006].

for "better" code, but may not account for effects on economic value at the organizational level (e.g. if the code is more costly to maintain or if it is more difficult to train operators to use open source software) that offset such more narrowly defined benefits. This study seeks to contribute to the development of the fuller context of open source phenomena to help illuminate the contributions and limits of individual studies.

Over time, an effective research agenda will aid in accumulating knowledge about a particular field. Such an agenda does not guarantee comparability of methods, measures, or even the naming of variables or constructs. It does, however, provide an opportunity for researchers to view prior work more easily and build upon it, rather than needing to invent new terminology and schemas.

DATA SOURCES

In approaching this study, we examined open source publications in refereed journals such as *IEEE Software* and *Organization Science* and in many on-line publications. Mainstream MIS journals such as *Information Systems Research* or *MIS Quarterly* were referenced infrequently because little has been published in these journals on open source. Work from other disciplines represents an opportunity to enhance the MIS perspective by drawing on and integrating economic, technical, and policy thinking. We used the literature to illustrate the categorizations and points we suggest about open source phenomena, but do not claim that this examination is comprehensive.

OPENS SOURCE PERSPECTIVES

The open source literature contains many approaches to gathering data. Case studies [e.g. Mockus et al., 2002; Watson, Wynn, and Boudreau, 2005], surveys [Ghosh & Prakash, 2000; Hertel, Niedner, and Herrmann, 2003; Stewart and Gosain, forthcoming-a], interview based research [e.g. Mahanmohan and De, 2004], and use of statistical analysis of archival data [e.g. Crowston and Howison, 2005; Krishnamurthy, 2002] were found, as were many essays and "think pieces". Koch and Gonzalez-Barahona [2005] indicate that on-line repositories of data will create many opportunities for archival research based on by-products from the open source process itself.

Although existing literature can be expected to generate many useful lessons, the data is no substitute for additional inquiry perspectives. We anticipate that future research by ourselves and others would profit from a broad mix of research methods. It will be important for interpretive researchers to consider the meanings and viewpoints of those both producing and using open source software to develop an understanding of their purposes and experiences. It will also be important for design science to investigate closely both the products and detailed methods for producing open source software, noting where they resemble and differ from traditional development methods. We anticipate a wide range of viewpoints expressed through preference for different methods to create many observations of the open source domain that address a wide range of different questions.

In this paper, we use a synthesis perspective so that varied points of view can be seen in relationship to one another.

RELATION TO THEORY PAPER

This paper presents a research agenda for open source. It assumes that methods for analyzing the research studies exist. In a companion paper [Niederman et al., 2006] which immediately follows this paper we discuss the available theoretical approaches for analyzing the results of the research. Readers are urged to read both papers to obtain a fuller understanding of the research proposed.

ORGANIZATION OF THIS PAPER

In Section II we present a multi-level open source framework that provides a detailed description of each of our five levels in terms of attributes that vary from one instantiation of open source software to another. It is our view that an understanding of the entire domain of open source activity accounts for the full range of these values, even if individual studies focus more specifically on particular relationships for subsets of instances. In Section III we propose that using multi-level theory provides a mechanism for defining relationships between variables at different levels. These relationships suggest specific research questions (some of which are starting to be addressed in the existing literature), whereas others represent new areas for investigation. We conclude in Section IV with a discussion of future research opportunities, and the limits of this paper.

II. AN OPEN SOURCE FRAMEWORK

We view the field of open source as large and complex with a variety of stakeholders, outcomes, influences, and evolving conditions. As a result, we use a multi-level approach [Klein et al., 1999], to help sort through such complexity. There is a significant history of using multiple levels of analysis in the development of multi-level theory in the behavioral sciences. Klein, et al. [1999] introduced a special issue of the *Academy of Management Review* dedicated to building multilevel theory, pointing out that "…although multilevel theories are necessarily complex, their complexity may yield important practical insights (p. 243)." We believe that, by sorting out the elements of the open source software discussion, an explicitly multilevel set of theories, perhaps a unifying theory can emerge. In this section we describe five distinct levels. While acknowledging that some levels could potentially be decomposed using additional distinctions, for parsimony we grouped some entities that we believe share significant similarities and may be difficult to segregate from one another with sharp boundaries. The five levels we identify are:

- the software artifact,

- the individual,

- the team/project/community,

- the organization, and

- society.

These levels are summarized in Table 1. Study within a particular level of analysis tends to include a significant number of descriptive studies that examine the nature of the variables at that level. In some cases, research questions may emerge about the relationship between two or more levels; for example among artifact type, licensing strategy, and product quality. We did not observe much research to date that studies "within level" questions. We anticipate that many of the most interesting research questions will involve relationships that cross levels. These questions are discussed below.

THE SOFTWARE ARTIFACT

Ultimately, the outcome of open source software projects is code that can be run to fulfill individual or organizational purposes. These artifacts (as with proprietary software) may differ significantly in type or functionality. We suggest that at least three types of software are sufficiently distinctive to merit individual attention. Using Madanmohan and De's [2004] taxonomy:

- infrastructure software (operating systems, middleware, database management systems, and support software),

- software tools (e.g. spreadsheets, web development kits such as Front Page or Dreamweaver),

- application software (e.g. payroll, accounts receivable), and cross-functional application software (e.g. CRM, customer relationship management, or ERP, enterprise resource planning).

MIS levels of analysis,	Variables	Research issues
Artifact	Artifact type (infrastructure, package, application, cross-functional application) License type (restrictiveness) Quality of product (fewer bugs, better security)	Contrasting open source and proprietary artifact characteristics Precursors to the choice of license type Effect of license type on diffusion and use of software
Individual	Developer User	Motivations for participation Choice of project Adoption decisions
Group, project, community	IV – organization governance (hierarchy, use of decision committees) Mechanics for artifact creation	Mixtures of paid and volunteer developers Processes for modularizing projects, "assigning" work tasks, for evaluating and integrating new code Communication processes and patterns
Organization	Developer Distributor Users	Business models for developers and distributors of open source software Total cost of ownership for investing in open source Mixtures of open source and proprietary software over a whole MIS department
Global	Influence on society	Diffusion of the open source "philosophy" to other areas such as licensing of intellectual property Governmental policies regarding the use of open source versus proprietary software

Table 1. Representation of Research Issues by Level of Analysis.

These software types are distinct by where they fall on the continuum between direct contact with hardware at one extreme and direct contact with cross-functional organizational business processes on the other. Contrasting the strengths and weaknesses of open source versus proprietary software within each of these software types may yield different results in either artifact level variables (e.g., quality or licensing arrangements) or cross level variables, (e.g., project management, organization style, or organizational total cost of ownership).

Mockus et al. [2002] present detailed analysis of the various artifact quality measures in their two case studies of Apache and Mozilla. Software artifacts can be examined for their intrinsic characteristics. Outcome measures apply to issues such as [Mockus et al., 2002]:

- reliability
- lack of bugs or errors

- quality of architecture
- flexibility and ability to incorporate new features,

- performance in particular environments or for particular tasks,
- ability to be applied across varied platforms and tasks

Researchers will, of course, be concerned with antecedents to the quality of the artifact. These antecedents are likely to come from other levels of analysis and will be discussed below in the section on cross-level analysis.

Another dimension on which software artifacts vary is their licensing requirements. For example, the Free Software Foundation (FSF) [Free Software Foundation, 2006] promotes the use of free software generally as distinguished from the use of proprietary software. They promote the most widely used license, the GNU General Public License (GPL) which contains two important restrictions [Stewart, Ammeter, and Maruping, 2005]:

1. that modified versions also be open (the "copyleft" provision), and

2. that the code may only be combined and distributed with code that also follows the openness provision (the "viral" characteristic).

The Open Source Initiative (OSI) [Open Source Initiative, 2006] generally promotes a broad set of licenses which enables the combination of free/open source software with proprietary software in many different licensing schemes. A less restrictive alternative is the Berkeley Software Distribution (BSD) which allows broader latitude for redistribution [Karels, 2003]. Feller and Fitzgerald [2000] present a taxonomy of licensing alternatives based on variation in price, redistribution policy, limitations on users/usage; available source code, and source code modifiability. Rosen [2005] provides an extensive discussion of the implications of open source licensing models. Other researchers suggest that organizations can benefit from using licensing considerations as part of the software development process [Al Marzouq et al., 2005].

Competing licensing alternatives number in the dozens, and include some that provide no restriction to how the open source artifact is used. Specification of licensing arrangements can be expected to influence allocation decisions by individual developers (e.g. the more restrictive the more likely the developers will receive benefits from their labor) and by individual or organizational users (e.g. the less restrictive the more likely they might be able to profit from incorporating the code in other projects and less likely to be challenged by innovative uses) [Stewart, et al 2005]. Clearly the specific licensing is a decision made by the project team or community responsible for the particular artifact, thus we consider it an attribute of the artifact as licenses are distributed with source code.

THE INDIVIDUAL

The level of the individual is clearly one where the user/developer role distinction is important. Many studies focused on the developer[3] and the roles played by different individual members of open source communities. The open source movement evolved from an early state where most users were also developers and single stakeholders. The present state, includes non-developer users [Jin, Robey, and Boudreau, 2005] and, perhaps, non-using developers. We focus on the

---

[3] Note that the developer may also be a user of the software artifact.

roles of "developer" and "user" recognizing that the same individuals may be involved in both roles. We see the developer role as that of creator, tester, and debugger of the software artifacts. We see the user role as anyone running the software artifact for personal or organizational purposes.

Among developers, an immediate distinction is between those who are salaried employees of corporations and those who completely or predominantly provide labor voluntarily. Individual developers play roles of initiator, release coordinator, core developers, co-developers, active users, and passive users [Crowston and Howison, 2005]. Different roles also exist within the framework of specific project/community structures.

Attributes of the individual user who contributes modestly, if at all, to the development or testing of the software may closely resemble those of the user of any software package whether open source or proprietary. Perens [2005] provides insight into the likely effect on professional software programmers suggesting that if open source products begin to displace proprietary ones, the demand for software will not go down. However, individual IS workers are likely to move to organizations "that can produce Open Source software in an economically successful manner."

Numerous papers address the motivation of individuals to spend time and effort in the development of open source artifacts without the direct extrinsic reward of payment (e.g., Hars & Ou, [2002]). Within a largely volunteer environment, traditional human resource measures such as job satisfaction and intention to turnover may not fit precisely. However, some roughly equivalent measures may target satisfaction with the particular project, with the particular artifact, and with the return on effort made. Turnover measures may target likelihood to participate in future projects (or continuing with current ones). Lerner and Tirole [2002] use labor economics to address this issue at some length. Outcome measures regarding individual motivation should include workplace issues such as productivity, particularly in contrasting work on open source with work on reasonably equivalent proprietary systems (e.g. [Mockus et al., 2002]). Bergquist and Ljungberg [2001] define motivation in terms of non-commercial transactions, such as gift giving.

Motivation is rarely a simple function of tangible compensation (though clearly compensation plays a significant role in work decisions). Von Hipple and von Krogh [2003] hold that developers may "profit" from learning, enjoyment from code writing, and from access to privileged status in the development community. Such "profit" may, in turn, lead to new opportunities and interesting challenges valued by the individual. In the long run, successful participation in open source activities can lead to commercial opportunities for selling services with particular open source artifacts, or to employment opportunities with firms seeking to capitalize on particular open source artifacts. It may also lead professionals in a given company, who also volunteer in open source projects, to become internal experts on behalf of their organization [Perens, 2005] or attract venture capital to build a business on top of the software artifact [Lerner and Tirole, 2002].

An additional alternative is to package proprietary and open source code into new applications. Such composite products require significant investigation into the terms and conditions of the licensing of the open source component [Ruffin and Ebert, 2004]. Madanmohan and De [2004] identified issues in finding and preparing to use open source code within other software products. They found that the first issue was assessing correct functionality, then checking on licensing and agreement with the "owner", and finally user interfaces and performance. For the decision to actually use the code, the same issues are faced as for using proprietary software (cost, ability to customize the software, performance attributes, licensing, and maintenance and support), although the weighting of these issues may vary between open source and proprietary code.

It is much easier to determine the research space for individual developers than for individual users. The concept of diffusion and adoption of technology would suggest considering:

- the patterns by which numbers of adopters change over time, and

- the kinds of decision making that individuals go through in selecting proprietary versus open source software for personal use.

The decision making process involves selecting software for use on personal discretion tasks within an organizational environment.

Outcome variables for individuals range from decision to install, decision to use, the amount and type of use, and the value (if any) gained from use. Again, this would likely be in contrast to similar decisions revolving around proprietary software, and by contrasting different types of users or different specific open source artifacts.

Another topic for study is the transition from passive user to active developer ([von Krogh et al., 2003]) and community member, as well as the reverse; transition from active developer to passive user or community "lurker".

THE TEAM/PROJECT/COMMUNITY

On the surface, teams, projects, and communities involve significantly different attributes. We've clustered these attributes here, however, considering:

1. their commonality of having both structure and dynamics, and

2. their existence as entities consisting of multiple individuals rather than all aspects of an entire organization.

In prior open source software research, these terms sometimes refer to the same thing; some papers refer to a team that works on a specific project for a particular artifact, and others refer to a community that works on or uses a specific project for a particular artifact. Both teams and community generally denote groups of individuals connected by common purpose. The community is generally the larger entity and may be comprised of multiple teams and, perhaps, individuals not specifically assigned to a team. The community includes people with an interest in, and perhaps contributing to, the artifacts being created. Teams, like individuals, may work on one or more artifact either at the same time or consecutively. Teams and communities vary in their membership, the boundary conditions for membership (e.g. individuals slipping in and out of membership), and the roles that various members play at a given time. Teams and communities are distinct from the collected attributes of their members. For example, it can happen over time that all members of a team leave and are replaced, with the overall team continuing its operation, perhaps smoothly, perhaps not. The work on one artifact may be a single project. However, at some size and complexity of project, work tends to be divided into multiple interrelated projects. In the open source world, the discovery of a major bug, disagreements among key participants, the adoption of the artifact for an additional platform, or the inclusion of a major new feature can generate a new project even if related to the same artifact.

Teams, projects, and communities are studied extensively in the social science, business, and communications literatures. Outcomes are often split into task and process related categories. Task-related categories include quantity, quality, efficiency, and effectiveness of outputs (which are often specific to the task). Process related categories are typically related to measures of the means by which these tasks were created – levels of participation, congruence with known effective techniques – and to overall satisfaction with the experience. Such outcomes include interest in continuing to work with the group and likelihood of taking on another activity.

The approach taken to work by open source team/project/community can be contrasted to traditional development methods. If the work is largely performed by volunteers, the leverage of open source leaders and coordinators vary from those of traditional managers. While Gallivan [2001] notes that control rather than trust was highlighted as an important factor in the set of open source case studies he observed, other empirical work shows an important role for trust in affecting outcomes in open source projects [Stewart and Gosain, 2006]. Crowston, et al. [2004] proposes a framework for examining team performance in the open source context based in part on coordination theory.

A research framework to guide research on open source user communities is proposed by Jin et al. [2005]. Specifically, they suggest and describe four potential areas of investigation: the creation of open source user communities, their characteristics, their contributions, and how they change.

A research agenda specifically focused on the development processes of open source communities based on the traditional reporting questions (who, what, where, when, why, and how) is proposed by Feller and Fitzgerald [2000].

Scacchi [2004] examines development methods in actual use by open source development communities. His findings include observation of five types of processes:

1. a use of threaded conversation and code functionality to guide requirements definition, rather than formal documentation;

2. controlled version control, builds, and release reviews;

3. maintenance as evolutionary redevelopment, reinvention, and revitalization;

4. project management, and

5. career development.

Crowston and Howison [2005] investigate whether open source communities are homogenous or differentiated in their structure, more specifically in their communication styles. The premise is that the study of a few successful cases showing the value of open source do not account for the range of open source projects/communities. Furthermore, they do not show whether it is open source per se that endows these projects with success, or if these cases are selected from a particular kind of open source project that is both successful and dependent on particular communication structures. This study found something approaching a normal probability distribution of projects from low to high levels of centralization of communication and a negative correlation between centralization and size. No large projects were highly centralized by the measures used in this study.

The differences in outcomes and processes of the typical open source project in contrast to typical traditional software development may challenge long-held management assertions. Studying users who do not participate in developing the underlying code offers opportunities to investigate methods for implementing and sustaining open source in the business environment. Measuring project management in the open source environment is challenging because the traditional measures of time and cost are not as clearly drawn when work is performed by volunteers.

Some project measures such as time between problem discovery and the release of a validated new version are relevant to, and can be used to contrast, open source and proprietary approaches. Another topic at the team, project, and community level pertains to the interactions of groups comprised solely of volunteers, solely of paid professionals, and of mixtures of the two. Does the group composition affect the quality of the underlying artifact? Given potentially differing philosophies, are their issues with establishing a unified leadership and coordination when teams are composed of volunteers and professionals? Of course, the way that these groups are treated by firms and as cooperatives may affect the outcomes beyond the simple count of different worker types.

Another area of study is the tools used and the organization of work in the virtual environment of open source. Much of the open source movement is based on computer mediated communication. Such communication provides an opportunity to test methods and tools developed for use in other domains such as organizational training or work flow processes in the open source environment and in traditional development.

ORGANIZATIONS

Organizations range in their level of involvement in the creation of new software artifacts. We identify three essential types, the user organization, the open source support organizations, and the professional open source organization.

User Organizations

User organizations use open source software much as they would proprietary software. Typically for these organizations, software itself is not the main product. Software supports other business functions. These organizations are concerned with:

- the net economic benefit/loss from choices to use or not use open source software;

- the costs and benefits of a mixed portfolio of open source and proprietary code within the same integrated domain;

- the basis of selection of open source software by task (e.g. infrastructure versus routine application versus mission critical application);

- the quality of implementing such software; and

- long term effects (e.g. reducing or increasing dependence on outside vendor or supplier).

To a large extent, the study of open source from the perspective of user organizations involves decisions about whether or not to invest in any use of open source technologies. As pointed out by Perens [2005], much of the open source software used by organizations does not distinguish them from competitors in their economic space. That is, open source software is unlikely to create distinctions among firms because it is relatively easily obtained, but it can potentially lower cost for the non-differentiating software needed by firms. On the other hand, investment in open source software could affect training and staff skills, compatibility among applications, challenges of customization or upgrading over time, and pressures for participation in the on-going support of the acquired systems.

It is not clear if there are short or long term security issues. The concept of many eyes examining artifacts arguably helps find bugs. Perhaps, the philosophical approach to open source will somewhat insulate these systems from malicious attacks. However, the net security advantage of open source is not held universally (e.g. Glass, 2005). Open source software community interactions are known to become difficult in some circumstances, resulting in the differentiation of programs into non-compatible sets. This problem has been largely averted in open source communities [Glass, 2005]. However, such possibilities do present a long term risk from open source in general.

After the organization's initial decision to invest, in open source, it faces further choices about :

- the best ways to implement open source artifacts;

- how to integrate them with the rest of the IT portfolio; and

- how to maintain and eventually retire them, as they would other software.

Issues such as the extent to which internal staff should be trained for integration and maintenance activities versus hiring professional open source support vendors will shape both human resource management planning and the nature of staff resources available for extensions and repairs. User organizations may acquire both responsibilities and privileges when using open source software, depending on the licensing arrangements. A topic for research is whether user organizations are willing to contribute to the continued evolution of open source artifacts by participating in communities and projects.

Open Source Support Organizations

Open source support organizations facilitate the development of open source projects. (e.g., Sourceforge.net) or focus on a particular artifact (e.g., Apache Foundation) or serves as primary sponsor for a particular license (e.g. FSF for GNU). The boundary between these supporting organizations and single artifact open source communities may be blurred. In many instances they will be similar on key attributes such as size, manner of organizing, level of formality, and types of products administered. The key outcomes for organizations such as these are the ability to attract and retain volunteer labor, ability to create quality artifacts, ability to generate enthusiasm, and diffuse products to a larger community. Significant data is made available to researchers for the study of Sourceforge.net activities through National Science Foundation support. Procedures for accessing research data can be obtained from Sourceforge.Net [2006].

Competing Proprietary Software Companies

Competing proprietary software companies may experience significant indirect effects from open source. These effects may affect the economic viability of some proprietary vendors whose loss of income from license fees may not be recoverable through services or other business models. This outcome is a reasonable result if licensing per se is one of the vendor firm's competitive competencies. However, vendor firms adapted to changing market forces in the past. They can be expected to develop new strategies and approaches for going head to head with open source providers. One such approach may be to retain proprietary software artifacts, but to expand community services for users. For example, SAP developed an extensive user network through its SAP Developer Network (SDN) Web site and announced intentions to create extensive user programs based on "open communities", "community process", and "community structure" [SAP France, 2006].

Professional Open Source Organizations

Professional open source organizations are built around open source code, but make money by providing services, documentation, and/or customization of the open source code for clients. These organizations may be distinguished by the degree to which they provide customization and software installation for client firms like a proprietary vendor versus simply providing documentation. Lerner and Tirole [2002] identify three approaches of software companies to commercialization of open source artifacts:

1. "living symbiotically" in relation to the artifact, providing complementary services and products;

2. releasing otherwise proprietary code and "creating some governance structure for the resulting open source process" in order to build a broader base of developer involvement and sell larger numbers of related products or services; and

3. provide certification for open source development programs, in essence acting as an agent.

These organizations represent a test bed for considering the ability of different artifact types (e.g. system software versus application software), different licensing arrangements, and alternative business models to translate into profitable enterprises. From an organization theory perspective, the open source and community movements represent an alternative to the market and hierarchy as means of organizing transactions [Watson et al., 2005].

The professional open source category provides an opportunity to test the sustainability of open source in the economy and the mixture of open source with other arrangements in the same firm or industrial segment, as in the case of JBoss [Watson, Wynn, and Boudreau, 2005] in the application server market. One industry trade publication states that "JBoss now has more users than either IBM WebSphere or BEA WebLogic [Technews, 2006]."

A second example of a professional open source firm is Compiere[4] which provides enterprise and CRM products as a free download, but requires payment for additional integration and support. A third example is open source startup Pentaho which focuses on business intelligence software. Pentaho claims more than 30K downloads a month with a 1-2% conversion rate for purchasers of the professional version [Sheina 2006].

For organizations as users of open source products, Madanmohan and Krishnamurthy [2005] examine four key issues:

- the motivation of commercial firms to interact with open source software,

- the range and type of involvement commercial firms can have with open source software,

- the challenges that commercial firms face with open source software communities, and

- coordination strategies that such firms use.

The term commercial firm is used by these researchers to differentiate firms that develop and distribute open source software from those that are clear arms-length users. Madanmohan and Krishnamurthy [2005] present a continuum of levels of involvement (or types of relationships) between commercial firms and open source software communities from simply observing community activity to sponsoring new projects.

Studies of professional open source organizations will include examining their allocation of costs and risks, the decision making and implementation processes, and secondary effects such as staffing profiles and the standardization/customization tradeoffs needed in software selection decisions.

Li et al. [2005] found that the human capital costs of adopting OSS can be quite high. The five levels are likely to interact. For example, selecting and implementing teams and projects and the hiring and retention of individuals with particular skills are likely to affect the success organizations experience with open source.

SOCIETY

The preponderance of the MIS open source research that we found examines issues at the artifact, individual, and organizational levels. Key dependent variables tend to focus on:

- the attributes of IT artifacts that make them more or less valuable,

- the use of technology by individual workers, groups and teams, and

- the role of information and technology in supporting organizational decision making or providing competitive advantage.

In a number of niche areas, the effects of MIS on industries, nations, and society at large are growing in importance. Some of these areas include [Niederman et al., 2002]:

- the cross-national outsourcing of work and jobs,

- the role of IT in development, and

- privacy and security in multinational organizations, institutions, and global IT consumers.

The increasing diffusion of open source software potentially impacts areas of society beyond the production, distribution and use of software artifacts. For example, von Hippel and von Krogh

---

[4] www.compiere.com

[2003] discuss open source as presenting a "novel and successful alternative to conventional innovation models" largely by involving the users in the innovation process. In addition, the tendency of some open source developers to work as volunteers from home may influence social phenomena, such as telecommuting, work-life distinctions (separation of work and personal life become blurred), and impacts on community and mental health that may derive from high levels of computing activity and Internet engagement.

Although we did not encounter such research literature, we expect to see open source studies about society at large in the following areas:

1. Forms of licensing applied to software such as "copyleft" and viral licensing to extend into other areas of intellectual property such as music, art, educational course materials, and product branding[5].

2. Increasing dispersion of open source approaches to the creation of knowledge assets through activities such as wikis where individuals collaborate to gather material for diverse enterprises including encyclopedias.

3. Use of open source rather than proprietary software to support business in developing countries and as a specific policy of governments in these countries to support low-cost adaptation of software for public needs.

We see initial signs that are consistent with open source approaches in countries like China that seek homegrown products based on their own standards in an effort to save on significant licensing fees. On the other hand, software commoditization may affect the GDP of exporting countries such as Ireland and Israel.

Researchers might well investigate whether, as open source code becomes more prevalent in traditional profit-making firms, the open source philosophy and development approach accompanies the adoption of the code. It is also possible that the open source philosophy of preferring the addition of assets to the public domain, rather than for proprietary use, may spread through society. The culture of the typical open source community and open source projects is frequently quite different from that of most for-profit organizations. Bergquist & Ljungberg, [2001] describe open source as founded on a "gift-giving" notion.   This spirit of gift-giving may shift the manner in which organizations make decisions or set some of their priorities.

III. CROSS LEVEL RELATIONSHIPS

The open source domain has many facets. As a result there are many potentially interesting avenues available for exploration. In this section, we begin with considering the key attributes of the artifact and consider how these may be examined in relation to variables at other levels of analysis. We start with the artifact because in many ways it is central to the open source domain. Clearly there is no open source domain without software products that aren't created and distributed through open source mechanisms. We continue by suggesting some aspects of individuals as developer, user, or both and how these might be related to the remaining levels of the open source domain. We next consider teams, projects, and communities in terms of how they might relate to organizations. We do not further consider the role of open source and society here because this level does not lend itself as clearly to specific variables and the same sort of research questions as do the other levels of analysis.

We would envision aspects of the artifact as natural dependent variables. The level of quality by any measure and the type of license should result from values of individuals, groups, teams,

---

[5] We already see this phenomenon happening. See for example the open content license at http://www.opencontent.org/openpub/   or   the   open   Music   licenses   at http://openmusic.linuxtag.org/modules/freecontent/content/openmusic/

A Research Agenda for Studying Open Source I: A Multi-Level Framework by F. Niederman, A. Davis, M.E. Greiner, D. Wynn and P.T. York

communities, and organizations, and to how they are organized and configured for interacting with the software itself. On the other hand, the artifact, particularly in terms of its licensing, can

Level	Relationships	Sample Research Questions
Artifact – individual	Quality by motivation Type by motivation Type by adoption License by motivation License by adoption	Does the variation in motivation or compensation (e.g. paid, volunteer, various combinations) affect artifact quality outcomes? Does developer motivation vary by type of artifact? Do developers work equivalently on infrastructure and cross functional artifacts? Do users adopt open source artifacts differentially based on artifact type? Does the type of licensing affect developer motivation or project selection (e.g. Stewart)? Does the type of licensing affect user adoption of open source artifacts (e.g. Stewart)?
Artifact – group, project, community	Quality by governance structure Quality by implementation mechanism Type by governance structure Type by implementation mechanism License by governance structure License by implementation mechanism	Do variations in the governance structure of open source communities significantly affect artifact quality? Do variations in implementation mechanisms significantly affect artifact quality? Both descriptively and normatively does the governance structure change with artifact type? Are implementation mechanisms equally effective given different artifact types? Does the governance structure of the open source development community affect choice of licensing of a particular artifact? Are particular implementation mechanisms aligned with particular types of licensing?
Artifact – organization	Type by developer/distributor Type by user License by developer/distributor License by user	Does the type of artifact influence the range of business models that are viable for open source developers? If so, how? Does the type of artifact influence the total cost of ownership for potentially adopting organizations? If so, how? Do differences in license type affect the business models' viability for developing organizations? If so, how? Do differences in license type affect the total cost of ownership for potentially adopting organizations? If so, how?

| Artifact – society | Type by government policy License by government policy | Will government policies toward open source artifact vary by type? If so, how? |
| | | Will government policies toward open source artifact vary by license arrangement? If so, how? |

Table 2. Cross Level Relationships Involving Artifact and Various Levels of Analysis.

be expected to act as an independent variable – affecting the reactions of individuals, groups, teams, communities, and organizations in the way they utilize it and in their continuing involvement with new versions and improvements. The study of these mutual impacts will be difficult for individual studies. However in collecting the results of many investigators, some sense of the mutual feedback loop may emerge. A selection of proposed research questions pertaining to artifacts and cross level investigation is presented in Table 2.

The individual developer and user can be expected to be influenced by the nature of the group, project, community, and organization (See Table 3). We would expect that the motivation of the developer would be heavily influenced by the type of governance structure of the development community and, perhaps, by the end using organizations. A possible experiment would result from the hypothetical arrangement of communities exhibiting the range of governance options and observation of developer choice for which to begin working with, and which best retains such efforts over time. We would also expect that individual attributes would influence the decision to adopt particular open source (or proprietary) software for varied purposes. It is not clear if the interaction between individuals, groups, communities, and organizations is independent of the artifact or if the nature of the artifact mediates such a relationship. For example, the relationship between individuals and organizations may turn out to be fairly stable across a particular type of software (e.g. for infrastructure or for enterprise software) but be different as regards other software types.

Level	Relationships	Sample Research Questions
Individual – group, project, community	Motivation by governance structure Motivation by implementation mechanism Adoption by governance structure Adoption by implementation mechanism	Do variations in the governance structure of open source communities significantly affect the motivation of individual developers? Do variations in implementation mechanisms significantly affect the productivity of individual developers? Are individual users affected in their adoption decisions by the open source community governance structure? Are individual users affected in their adoption decisions by the specific implementation mechanisms of the open source community?

Table 3. Individual and Group, Project, Community Relationships

The group, project, and community will vary particularly in governance structures and mechanisms used for development and project management. Organizations potentially adopting open source software will be in different industries, including both software creators as a product and software users only. Differences in the way that groups and communities are structured can be expected to influence the confidence organizations have in them, and the likelihood they will invest in their products. The way that projects are run and the methods used should also influence the willingness of organizations to participate in, and contribute to, the development process particularly in terms of maintenance and improvements.

Level	Relationships	Sample Research Questions
Group, project, community -- organization	Governance structure and developer organization	Do variations in the governance structure of open source communities significantly affect the motivation of organizations to adopt their artifacts?
	Governance structure – user organization	
	Implementation mechanism and developer organization	Do variations in implementation mechanisms significantly affect the likelihood of organizations to adopt their artifacts?
	Implementation mechanism -- user organization	Do variations in the governance structure of open source communities significantly affect the motivation of organizations to participate in and contribute to on-going maintenance and improvements?
		Do variations in implementation mechanisms significantly affect the likelihood of organizations to participate in on-going maintenance and improvements?

Table 4. Cross-Level Relationships between Group, Project, Community and Organization.

We have presented in this section a number of examples of how the combination of variables at multiple levels of analysis can trigger research questions for open source investigators. It is unlikely that all of these questions will prove to be profitable avenues of investigation. We can, however, envision a point where the accumulation of answers to these questions, even if presented provisionally in a set of narrower studies, will provide significant insight and, perhaps, predictability, for practitioners in the open source arena.

IV. CONCLUSION

As can be seen, the domain of open source research is extensive. Already a significant body of research has accumulated in this domain.   The number of stakeholders, variables, and extensions into other domains among varied elements of interest is large and the relationships potentially highly complex. Although many research targets are worthy of continued investigation, we see a prime target for research, particularly within the MIS community, to be organizations as users of open source artifacts. Such research would focus on the issues such organizations face entering into initial open source usage, integrating open source into their portfolio; deciding on levels of community participation; and assessing the economic, organizational, and technical impacts of open source on operations, tactical and strategic business practices.

LIMITATIONS

As with all studies, this one has limitations. The method used for developing this paper is based on the discussions and thinking primarily among the authors and colleagues. In the end, we focused on the presentation of a multi-level view of the open source domain. Although a wide range and a large number of open source related papers were identified and reviewed, there can be no guarantee that coverage across the range of studies was comprehensive. We focused our attention on the content of findings in the various studies considered, rather than on details of their methodology.   It is possible that findings from studies referenced will in the future be supplanted by new work.

FINAL POINTS

A multi-level view provides a helpful background context identifying the broad range of issues into which specific studies may be viewed as providing narrow but important contributions. We believe that it is only through the development of a broad and inclusive framework that highly contrasting pieces of information can be viewed as contributing to a larger mosaic for understanding the full open source domain. Although a good deal of research already addresses open source phenomena, it is our observation that many opportunities have been surfaced to address new issues and to extend current knowledge.

ACKNOWLEDGEMENTS

REFERENCES

Editor's Note: The following reference list contains hyperlinks to World Wide Web pages. Readers who have the ability to access the Web directly from their word processor or are reading the paper on the Web, can gain direct access to these linked references. Readers are warned, however, that

> 1. these links existed as of the date of publication but are not guaranteed to be working thereafter.
> 2. the contents of Web pages may change over time. Where version information is provided in the References, different versions may not contain the information or the conclusions referenced.
> 3. the author(s) of the Web pages, not AIS, is (are) responsible for the accuracy of their content.
> 4. the author(s) of this article, not AIS, is (are) responsible for the accuracy of the URL and version information.

Al Marzoug, M., Zheng, L., Rong, G., Grover, V. (2005) "Open source: Concepts, benefits, and challenges", *Communications of the Association for Information Systems* (16), pp. 505-521.

Bergquist, M. and J. Ljungberg (2001) "The Power of Gifts: Organising Social Relationships in Open Source Communities", *Information Systems Journal* (11)4, pp. 305-320.

Crowston, K., H. Annabi, et al. (2004) "Towards a Portfolio of FLOSS project Success Measures",

Crowston, K., H. Annabi, J. Howiston, & C. Masango (2004), "Towards a Portfolio of FLOSS Project Success Measures", Proceedings of the Open Source Workshop of the International Conference on Software Engineering (ICSE 2004).

Cukier, K. (2005) "A Market for Ideas: A survey of patents and technology", *The Economist*, pp. 1-16.

Feller, J. and B. Fitzgerald (2000) "A Framework Analysis of the Open Source Software Development Paradigm", The 21st International Conference in Information Systems (ICIS 2000).

Free Software Foundation (2006) http://www.opensource.org/.

Gallivan, M. J. (2001) "Striking a Balance between Trust and Control in a Virtual Organization: A Content Analysis of Open Source Software Case Studies." *Information Systems Journal* (11)4, pp. 277-304.

Ghosh and Prakash (2000), "The Orbiten Free Software Survey", http://orbiten.org/ofss/01.html.

Glass, R.L. (2005) "Standing in Front of the Open Source Steamroller," In (Eds. Feller, J., Fitzgerald, B., Hissam, S.A., and Lakhani, K.R.), *Perspectives on Free and Open Source Software*, The MIT Press:Cambridge Massachusetts, chapter 4, pp. 81-92.

Hars, A. and S. Ou (2002) "Working for Free? Motivations for Participating in Open-Source Projects", *International Journal of Electronic Commerce* (6)3, pp. 25-39.

Hertel, G., Niedener, S., & Herrmann, S. (2003) "Motivation of Software Developers in Open Source Projects: An Internet-based survey of contributors to the Linux Kernel", *Research Policy*, (32)7, pp.1159-117.

Jin, L., Robey, D., and Boudreau, M.C. (2005) "Beyond Development: A Research Agenda for Investigating Open Source Software User Communities," Proceeding of the International Resources Management Association Conference, San Diego, May 15-18, pp. 642-645.

Karels, M.J. (2003). "Commercializing Open Source Software," *Queue*, July/August, pp. 46-55.

Klein, K.J., Tosi, H., and Canella Jr., A.A. (1999) "Multilevel Theory Building: Benefits, Barriers, and New Developments", *Academy of Management Review* (24)2, pp. 243-248.

Koch, S. and J. M. Gonzalez- Barahona (2005) "Open Source software engineering - The state of research", *First Monday*, http://www.firstmonday.org/issues/special10_10/koch/index.html (10)10.

Krishnamurthy, S. (2005) "An Analysis of Open Source Business Models," In (Eds. Feller, J., Fitzgerald, B., Hissam, S.A., and Lakhani, K.R.), *Perspectives on Free and Open Source Software*, The MIT Press:Cambridge Massachusetts, chapter 4, pp. 81-92..

Krishnamurthy, S. (2002) "Cave or Community? An Empirical Examination of 100 Mature Open Source Projects", *First Monday*, http://www.firstmonday.org/issues/issue7_6/krishnamurthy/index.html (7)6.

Lerner, J. and J. Tirole (2002) "Some simple economics of open source", *The Journal of Industrial Economics* (50)2, pp. 197.

Madanmohan, T.R. and De, R. (2004) "Open Source Reuse in Commercial Firms", *IEEE Software*, (21)6 pp. 62-69.

Madanmohan, T. R. and S. Krishnamurthy (2005) "Can the Cathedral Co-exist with the Bazaar? An Analysis of Open Source Software in Commercial Firms", *First Monday*, http://firstmonday.org/issues/special10_10/madanmohan/index.html Special Issue # 2.

Mockus, A., R. T. Fielding, et al. (2002) "Two Case Studies of Open Source Software Development: Apache and Mozilla", *ACM Transactions on Software Engineering and Methodology* (11)3, pp. 309-346.

Niederman, F., Boggs, D. and Kundu, S. (2002) "International Business as a Referent Discipline for Global Information Management Research: Ten Years of Experience," *Journal of Global Information Management* (10)1.

Niederman, F, A. Davis, M.E. Greiner, D.Wynn, & P.T. York, (2006), "Research Agenda for Studying Open Source II: View Through the Lens of Referent Discipline Theories," *Communications of the Association for Information Systems*, (18)8, pp. 129-150.

OpenOffice.org. (2006) download from website, http://stats.openoffice.org/index.html#stats, May 22, 2006.

Opens Source Initiative (2006) Open Source Initiative (http://www.opensource.org/)

Perens, B. (2005) The Emerging Economic Paradigm of Open Source, *First Monday* (10)10, http://www.firstmonday.org/issues/special10_10/perens/index.html.

Rosen, L. (2005) *Open Source Licensing: Software Freedom and Intellectual Property Law*, Upper Saddle River, NJ: Prentice Hall.

Ruffin, M. and C. Ebert (2004) "Using Open Source Software in Product Development: A Primer." *IEEE Software* (21)1, pp. 82-86.

SAP France, "Shai Agassi Declares Mature Partner Ecosystem Will Be Key Factor for Customers' Success in the Future," press release, April 18, 2006, accessed at http://www.sap.com/france/Company/Press/Press.epx?PressID=6117, May 31, 2006.

Scacchi, W. (2004) "Free and Open Source Development Practices in the Game Community." *IEEE Software* (21)1, pp. 59-66.

Sheina, M. (2006) "Pentaho Readies Commercial Open Source BI Platform," Computer Business Review on-line, 13th April 2006. http://www.cbronline.com/article_news.asp?guid=C6C93A17-FC04-423A-8CA7-6C0C49BCC422. Accessed May 31, 2006.

Sourceforge.net (2006). http://www.nd.edu/~oss/Data/data.html (Last accessed August 8, 2006)

Stewart, K.J., Ammeter, A.P., and Maruping, L.M. (2005) "A Preliminary Analysis of the Influences of License Choice and Organizational Sponsorship on Success in Open Source Software Development Projects", in the proceedings of the 38th Hawaii International Conference on Systems Sciences.

Stewart, K. J. and S. Gosain (2006). "The Impact of ideology on Effectiveness in Open Source Software Development Teams." *MIS Quarterly* 30(2), pp. 291-314.

TechNews, (2006),"Open Source Business Intelligence Platform, Pentaho BI Project", http://www.technologynewsdaily.com/node/811, accessed May 313, 2006.

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

von Krogh, G., S. Spaeth, et al. (2003) "Community, joining, and specialization in open source software innovation: a case study", *Research Policy* (32)7, pp. 1217-1241.

Watson, R. T., et al. (2005). "Governance and Global Communities." Journal of International Management 11(2): pp. 125-142.

Watson, R.T., Wynn, D.E., Boudreau, M.C. (2005). "JBOSS: The Evolution of Professional Open Source Software," *MIS Quarterly Executive* (4:3), September 2005, pp 329-341.

ABOUT THE AUTHORS

Fred Niederman serves as the Shaughnessy Endowed Professor of MIS at Saint Louis University. He has published broadly in *MIS Quarterly*, *Decision Sciences*, *DATABASE, Journal*

of Global Information Management, *Communications of the ACM,* and *Communications of AIS*. He currently serves as associate editor for the *Journal of Global Information Management*. He is former chair of the ACM special interest group on Computer Personnel Research and has been active in supporting its annual conferences. His research interests include virtual teams, groups, and meetings; global information management, and IT personnel and workforce.

Ashley R. Davis is a PhD student in the Terry College of Business at the University of Georgia. Her research interests include open source business development, enterprise resource planning systems, information technology infrastructure decisions, and service oriented architecture.

Martina Greiner is a PhD student in the MIS department of the Terry College of Business, the University of Georgia. Her current research areas are open source software and open content communities. She has published in the *Journal of International Management,* as well as in several IS conference proceedings.

Paul T. York is a Ph.D. Student in the Management Information Systems department at the University of Georgia. His current research interests include open source business models, information technology outsourcing, and the interaction between organizational policies and the global political economy.

Donald Wynn, Jr. is a PhD student in the MIS department of the Terry College of Business, the University of Georgia. His current research interests include open source software communities, ecosystems, business models, technology innovation, and information systems security. He has published in the *Journal of International Management* and the *Journal of the Academy of Marketing Science (JAMS)*.

.