18 DEVELOPING OPEN SOURCE SOFTWARE: A Community-Based Analysis of Research

Joseph Feller
Patrick Finnegan
David Kelly
Maurice MacNamara
*University College
Cork, Ireland*

Abstract   *Open source software (OSS) creates the potential for the inclusion of large and diverse communities in every aspect of the software development and consumption life cycle. However, despite 6 years of effort by an ever growing research community, we still don't know exactly what we do and don't know about OSS, nor do we have a clear idea about the basis for our knowledge. This paper presents an analysis of 155 research artefacts in the area of open source software. The purpose of the study is to identify the kinds of open source project communities that have been researched, the kinds of research questions that have been asked, and the methodologies used by researchers. Emerging from the study is a clearer understanding of what we do and don't know about open source software, and recommendations for future research efforts*

1   INTRODUCTION

The licensing and distribution terms of OSS create the potential for the inclusion of large and diverse communities in every aspect of the software development and consumption life cycle. This social inclusion takes many forms. For example,

* A third-party is potentially able to deliver services in level-field competition with the creator of the software, leveraging the open nature of the source code; this lowers barriers to entry for smaller service providers and provides autonomy to

software users who are no longer locked-in to a single vendor (Woods and Guliani 2005).

• User and developer communities can potentially share or assume the burden of innovation, a process traditionally located privately within the firm (Von Hippel 2005).

• Low acquisition costs, user-empowering freedoms, and the removal of information asymmetry potentially serve as powerful tools in combating the digital divide and in creating autonomy and local knowledge resources in the developing world (see, for example, Feller et al. 2003; James 2003; Steinmueller 2001; Yee 1999).

• OSS is seen as a mechanism by which public bodies can improve the transparency of, and provide wider access to, government services (see, for example, the archive of policy documents and case studies in the Center of Open Source and Government at http://www.egovos.org/).

Given these potential implications, it is unsurprising that, since the coining of the term in 1998, OSS has enjoyed a wide-spread surge in interest among users, developers, for-profit and nonprofit organizations, governments, and—last but certainly not least—researchers. However, with 6 years of research behind us, we still don't know exactly what we do and don't know about the phenomenon, nor do we have a clear idea about the basis for our knowledge. This paper presents an analysis of 155 research artefacts, mostly peer-reviewed, published between 1998 and 2004. The analysis focuses on three questions.

1. What types of OSS projects have been the subject of research?
2. What areas or topics have been the subject of research?
3. What methodologies have been used?

The rationale behind asking these questions is fundamental. OSS is surrounded by hype and hope—that it will revolutionize software development, the software industry, and, potentially, the information society. To separate hype from reality, and to help realize the potential benefits of OSS, the academic community needs to take stock of the research to date and clearly articulate the work that remains to be done.

2   THEORETICAL FOUNDATION AND RESEARCH METHOD

It has been argued that the structured identification of required future research directions is important, particularly in an emerging research area (Culnan 1987). The examination of previous work enables both the determination of progress made (Farhoomand 1987), and also the identification of work required in the future to further develop a field of study (Alavi and Carlson 1992). Such an approach has been used at both a disciplinary level—for example, within Information Systems (Alavi and Carlson 1992; Chen

and Hirschheim 2004; Claver et al. 2000; Farhoomand and Drury 1999; Orlikowski and Baroudi 1991), Software Engineering (Glass et al. 2002), and Computer Science (Ramesh et al. 2004)—and at a thematic or subfield level (e.g., Romano and Fjermestad 2002). Such works have surveyed the existing literature to investigate the paradigmatic approaches to research (e.g., Chen and Hirschheim 2004; Orlikowski and Baroudi 1991), the research methods or strategies used (e.g., Chen and Hirschheim 2004; Claver et al. 2000; Farhoomand 1987; Orlikowski and Baroudi 1991), and topics investigated (e.g., Farhoomand 1987; Glass et al. 2002; Ramesh et al. 2004; Romano and Fjermestad 2002).

Previous efforts at assessing the state of knowledge within an field have limited their sample to a specific number of outlets in order to investigate the methods being used (e.g., Chen and Hirschheim 2004; Farhoomand and Drury 1999) or the paradigmatic focus into which the research falls (e.g., Chen and Hirschheim 2004; Orlikowski and Baroudi 1991). In order to take into account the multidisciplinary nature of research in the area of OSS, the literature search was not confined to specific publications within any one discipline. Rather, a strategy of exploring a range of outlets was employed, following similar efforts by Romano and Fjermestad (2002).

Candidate papers were discovered through keyword searches of citation indices (e.g., EBSCO, Science-Direct, IEEE, ACM Portal), by using existing bibliographies of OSS research, and through recursion using the references cited within papers. In addition to a range of journals from various disciplines (see Table 1), a variety of international conferences and three books (DiBona et al. 1999; Koch 2004; Raymond 1999) were also reviewed. In total, 155 research artefacts published since 1998 were reviewed. Of these, 99 were journal papers, 37 were conference papers, and the remaining 21 consisted of various books, reviews and commentaries.

A number of limitations are evident in the approach used for identifying artefacts. In employing a strategy of exploring as many outlets as possible, the research sources used were not limited to ranked journals, as was the case in similar research evaluation efforts (e.g., Farhoomand and Drury 1999). Additionally, research on OSS in specific application spaces exists in specialized publications that were not included, thus the identification of all relevant literature can not be guaranteed. Finally, by not including publications prior to 1998, there is an implication that there was no literature in the area of OSS produced prior to that date. Such a view would be inaccurate, however, as the majority of publications prior to this year were both descriptive and published in non-peer-reviewed outlets, thus they were not included in the study.

Classification systems provide a means to communicate the contents of a field of study, and thereby enable the generalization and communication of findings (Vessey et al. 2002, 2005). For this study, each artefact was first analyzed to determine the type(s) of development community the research artefact investigated. Following the characterization by primary OSS community, each artefact was further categorized by (1) research focus and (2) research method. Within research focus, artefacts were categorized as software engineering issues, economic and business model issues, socio-cultural and organizational issues, and software application space. Table 2 illustrates some sample topics associated with each focus area. Table 3 illustrates the methodological labels.

Table 1. List of Journals Covered

ACM Transactions on Software Engineering and Methodology	IEEE Review
Briefings in Bioinformatics	IEEE Software
Business Horizons	IEEE Transactions on Consumer Electronics
Communications of the ACM	IEEE Transactions on Software Engineering
Computer	Information Systems Journal
CPA Journal	International Review of Industrial Property and Copyright Law
Electronic Markets	Journal of Law, Economics and Organization
European Journal of Information Systems	Organization Science
First Monday	Research Policy
IEE Proceedings–Software	Information, Technology, and People
IEE Proceedings–Software Engineering	Sloan Management Review
IEE Proceedings–Software Engineering,	The Information Society
IEEE Computer	Transactions in GIS

Table 2. Research Focus Areas with Sample Topics

Focus Area	Sample Topics
Software Engineering Issues	• Version Control • Software Architecture • Development Methodology
Economic and Business Model Issues	• Revenue Models • Resource Allocation • Market Drivers
Socio-Cultural and Organizational Issues	• Conflict Resolution • Motivation • Legal Issues
Software Application Spaces	• Specific Vertical Sector (Automotive, Health, etc.) • Specific Horizontal Sector (Financials, Human Resources, etc.) • Software Acquisition and Management

Table 3. Research Methdology Labels with Definitions

Label	Definition
Anecdotal/Descriptive	Little or no formal data-gathering methodology.
Secondary	(Re-)Analysis of previous research.
Case Study	Formal, high-depth data gathering focused on a single research site.
Cross-Case	Formal, medium-depth data gathering focused on 2-3 research sites with comparative analysis.
Field Study	Formal, low-depth data gathering across a wide number of research sites with comparative analysis.
Survey	High-volume structured questionnaire.
Experiment	Laboratory or field-experiment.

Table 4. Analysis of Artefacts per Publication Outlet

Year	Journals	Conferences
1998	3	0
1999	12	0
2000	6	7
2001	17	5
2002	17	8
2003	20	12
2004	24	5
Total	99	37

3 FINDINGS

OSS research has been increasing steadily in recent years. Analysis of the outlets used in the publication of such research (see Table 4) shows that journals (particularly special issues) have been the predominant means for the communication of findings.

3.1 Communities of OSS Development

Our analysis began by classifying the unit (community) within which the software was developed. The objective of this exercise was to provide a means of describing the groups involved as a set of organizational forms (see Doty and Glick 1994). In doing so, four community types were identified: *ad hoc* communities, standardized communities, organized communities, and commercial organizations. The key differentiating characteristics of these community types are summarized in Table 5.

Table 5. Characteristics of OSS Community Types

	Ad Hoc Communities	Standardized Communities	Organized Communities	Commercial Organizations
General Structure	Small, informal communities of practice collaborating "in the wild" on OSS projects of limited size (as measured by number of users and developers)	More mature (older, more stable) communities of practice with more formalized software development and management standards in place to address larger project sizes.	Very mature communities of practice which go beyond the creation of standardized practices to the formal (legal) establishment of an organizational entity.	Communities of practice embedded in formal (legal) profit-seeking firms.
Environment	Internet-based collaboration tools, generally hosted by a third-party (e.g., the SourceForge repository).	Self- or third-party hosted Internet-based collaboration tools. Generally used in conjunction with a self-hosted or sponsored identity-building environment (e.g., a group (rather than project) Web page).	Self-hosted Internet-based collaboration tools. Mature identity building environment. Possible physical collocation of some project members.	Self-hosted Internet-based collaboration tools and mature identity building environment. Physical collocation of some project members. Explicit integration with corporate development, communication and management structures.
Goals	Driven by individual goals, e.g., to meet personal computing requirements, to collaborate with others, to share output with the community, to gain personal reputation, to learn, etc.	Goals from ad hoc communities plus group-focused goals like quality assurance, project management, standardization, all towards the overall goal of building a public good.	Goals from standardized communities plus the need to provide legal protection for contributors and engage in organization-to-organization relationships with other organizations, firms, governments, etc. Desire to give project a "life of its own" independent of individual members.	Many of the goals from organized communities plus the desire to effectively utilize OSS dynamics, and to interact with wider communities, in order to generate share-holder value.

	Ad Hoc Communities	**Standardized Communities**	**Organized Communities**	**Commercial Organizations**
Methods	Individual-based methods with little formal documentation or standardization.	Standardization and documentation of key development methods and processes.	Standardization and documentation of key development methods and processes, formal project and organizational management.	Standardization and documentation of key development methods and processes, formal project and organizational management.
User Community	Users are generally other developers, early-adopters and power users.	Users are generally other developers, early-adopters and power users.	Users are both other developers, early-adopters and power users as well as main stream end users.	Users are both other developers, early-adopters and power users as well as main stream end users. Users are treated as customers.
Licensing	Generic licensing	Generic licensing, possibly project specific licensing.	Project specific licensing.	Corporate licensing.

3.1.1 General Structure

The use of general formal structure in the development of classifications based on organizational characteristics on the basis of formalization, specialization, and levels has been proposed by McKelvey (1978). Here, general structure refers to the organizational size, formality of structures, and legal standing held by each software production community. Gacek and Arief (2004) found that size alone is not a distinctive measure of an OSS project, with the code base and community varying between projects. It is with this in mind that the formalization of software development practices and management standards, and the establishment of a formal (legal) organizational entity are included as additional differentiating factors within the general structure dimension.

In the case of *ad hoc* communities, the size of project groups (as measured by the number of developers and users) is small with little formal structure. Illustrative of the size of such projects are the results of the Orbiten Free Software Survey, which found that 75 percent of projects had only one author participating (Ghosh and Prakash 2000). Capiluppi et al. (2003), following an analysis of the FreshMeat portal (http://freshmeat.net), reported that 57 percent of projects have one or two developers, and 80 percent have less than 11 subscribers (a proxy of users of the project's application).

The addition of formalized software development and management standards separates *ad hoc* OSS communities from standardized communities within the dimension of general structure. As projects mature, increased numbers of users, a growing code base, and a need to facilitate larger scale, distributed development requires the implementation of project management tools and techniques.

The key differentiating factor between standardized communities and organized communities is the move beyond formal standardized practices, and to becoming a formal legal organizational entity. The establishment of such noncommercial organizations enables the formalization of the projects' administrative functions in a board of directors, while allowing its further development and advancement to remain with willing individual developers. Well known examples of projects within the organized community classification are Apache and Mozilla.

The final community classification identified is that of a commercial organization. The general structure of communities within this grouping is that of a community of practice, which is a formal (legal) profit-seeking firm. Projects falling within this classification can be categorized as those emerging from organizations with either "pure-play" open source business models, or with a hybrid (i.e., a mix of both proprietary and open source) business model. Examples of projects classified under the commercial organization heading are OSS projects emerging from Red Hat, Sun, IBM, and Sony.

3.1.2   Environment

The development environment refers to both where development occurs and the tools utilized to facilitate communication and collaboration in the development process. *Ad hoc* communities use primarily Internet-based collaboration tools, which are generally hosted by a third-party, for example the SourceForge (http://sourceforge.net) repository. Such projects are generally of a size that does not warrant the maintenance of an independent Web presence (Feller and Fitzgerald 2002). The use of such Internet-based collaboration tools enables geographically dispersed project members to communicate and share source code (Moon and Sproull 2000). Development within a standardized community generally takes place using either a self- or third-party hosted Internet-based collaboration environment. Often, in conjunction with this, a self-hosted or sponsored identity-building Web page is also maintained. While development within standardized communities is generally distributed, within an organized community there is also the possibility of physical collocation of project members (e.g., Lussier 2004). In addition, identity building is again supported by group (versus project) Web pages, for example, the Mozilla Organization (www.mozilla.org) and the Apache Software Foundation (www.apache.org). Within commercial organizations, development can also rely on higher levels of physical collocation of project members and involves explicit integration with corporate development, communication, and management structures. As with organized communities, commercial organizations are more likely to use self-hosted Internet-based collaboration tools in order to facilitate development with geographically dispersed members.

3.1.3   Goals

The motivation of the participants involved in OSS production has been the subject of a wide range of research (see, for example, Hars and Ou 2001; Hertel et al. 2003; von Hippel and von Krogh 2003). At the level of the *ad hoc* community, participants are driven primarily by individual goals. Such individual goals can be to meet personal

computing requirements or fill a void in functionality (Nakakoji et al. 2002), or to learn new programming languages (Feller and Fitzgerald 2002; Ye et al. 2002). Other individual goals that have been identified are to share output with a community (Berquist and Ljunberg 2001; Feller and Fitzgerald 2002) or to enhance reputation with peers (Berquist and Ljunberg 2001; Gacek and Arief 2004). Within the standardized community, while the individual goals are still present, more group focused goals aimed at project development also emerge. These include quality assurance, project management, and standardization, all of which contribute to the good of the overall project or community. Within the organized community space, again the goals of the previous two communities are subsumed. There is also the objective of both providing a level of legal protection to contributors, and enabling the interaction between the community and other established legal entities, such as nonprofit organizations, companies, or governments. The establishment of a formal (legal) entity also allows the project to be given a life of its own, independent of individual contributors. The final community classification, commercial organizations, subsumes many of the goals of the previous community types. Such organizations also have the objective of leveraging the knowledge and competencies of the wider OSS development community in order to generate shareholder value (Brown and Booch 2002).

3.1.4 Methods

Development method used gives another dimension by which to classify communities. In the area of *ad hoc* communities, individual-based methods with little formal documentation or standardization are used. Given the low numbers of contributors and users of such projects, as discussed in the "General Structure" section above, the need for more formalized development methods and processes in such projects is relatively low. Higher levels of development method and process standardization and documentation can be found within projects classified as belonging to a standardized community. Such formalization is required to deal with growing project code and contributor community sizes and to enable wider distributed development. As with the standardized community, projects classified as belonging to an organized community rely on the standardization and documentation of methods and processes in order to facilitate larger scale distributed development. An example of such a formalized approach to development is evident within the PyPy project where agile development methods are combined with the frequent use of Sprints, involving the collocation of project members for a short time, to complement distributed development (During 2005). In addition to formalization, the establishment of a formal organizational structure also includes more structured approaches to project and organizational management (for example, Mockus et al. 2002). The level of formalization within commercial organizations reflects that identified in organized communities. However, hybrid development approaches consisting of aspects of traditional and Open Source development approaches (Mockus et al. 2000, 2002) may also be present as organizations attempt to take advantage of the strengths of OSS development methods (see, for example, Dinkelacker et al. 2002).

3.1.5   User Community

Research on OSS user profiles is lacking (Feller and Fitzgerald 2002). It has been argued that OSS diffusion has occurred primarily in areas where the end-user is technically sophisticated, many of whom are developers and wish to modify the source code themselves (Lerner and Tirole 2000; West and Dedrick 2001). Within the community classification, users of projects within the *ad hoc* and standardized community spaces are generally other developers, early-adopters, and "power-users" that possess the technical skills to use, adapt, and maintain such software. Within the organized community space, there is again an overlap between these types of users, and more mainstream end-users. An example of such a situation is the ongoing development of the Mozilla range of software products which are aimed at the personal end-user or desktop market segment. Within the commercial organization community, the user group is similar to that of the organized community. Users within this area are, however, treated as customers with the possible inclusion of value-added services such as technical support, training, documentation, or consultancy.

3.1.6   Licensing

The choice of license within OSS is used to impose a variety of restrictions on users (Lerner and Tirole 2005). Such licenses often specify the conditions under which the software may be used, modified, or distributed and the restrictions in place on copyright and the software's open source status.[1] The legal implications resulting from the use of different licensing structures can vary widely (Ruffin and Ebert 2004). This results in the choice of licence imposed impacting a range of stakeholders, including the community of programmers working on the project, the end users, other open source projects that will later compete with or complement the project, and commercial vendors or support providers (Lerner and Tirole 2005). Given both the range of licenses available and their differing uses (Lerner and Tirole 2005; Wu and Lin 2001), license alone does not provide a useful means to classify OSS community types. This is because the level of restriction placed on projects through their license can vary depending on the environment in which the software is to operate, the intended user audience, and the maturity of the project (Lerner and Tirole 2005). The type of licensing structure imposed does, however, provide an additional dimension upon which a broader classification can be based.

Within *ad hoc* communities generally, generic licensing will be used, perhaps because of the range of generic licensing available and the fact that project initiators within this space may not want to develop specific licensing. Examples of such generic licensing options are GNU's GPL and LGPL, as well as public domain licensing options. Within the standardized community space, licensing options can include either generic licences or licences developed specifically for the project. Organized community OSS projects are more likely to implement project-specific licences, with

---

[1]Adapted from the Open Source Initiative, Open Source Definition, Version 1.9 (http://www.opensource.org/docs/definition.php).

examples being the Apache Software License and the Mozilla Public Licenses. Commercial organization communities are more likely to use corporate licensing structures specific to their project, often with the objective of retaining a level of control over the software's development and subsequent use. Examples of such corporate licensing structures include the Apple Public Source License, the IBM Public License, the Zope Public license, as well as a range of non-OSI (Open Source Initiative) approved proprietary licensing structures.

3.2 Characterization of Research on OSS Communities

As shown in Figure 1, nearly two-thirds of the research artefacts analyzed drew data from the standardized and organized community types. It would appear that this is at least partially because the "headliner" OSS projects, such as Apache, Mozilla, Perl, and Linux fall into these categories and have each been quite extensively researched. Within the remaining one-third of artefacts, there was a 2:1 ratio of commercial organizations to *ad hoc* communities. Again, this is partially because of high profile commercial organizations like Sun and IBM, and also due to the fact that commercial organization focused research covered a wider range of topics than the *ad hoc* community focused research.

In looking at the OSS production communities which have been studied (Figure 2), it is interesting to note that there has been little change in the numbers of artefacts focusing on commercial organizations since 2001. It is instead the *ad hoc* and organized community classifications that have shown consistent growth in recent years. There are a number of possible explanations for such findings. The growing awareness of and interest in the "headliner" open source projects within both the research and business communities has led to increasing levels of research being conducted. The second possible explanation is that research in the area of OSS in general (i.e., not specifically relating to individual projects or community types) has been increasing. As the community classification scheme used assigns multiple classifications to such general research, it is possible that the trends presented have been influenced.
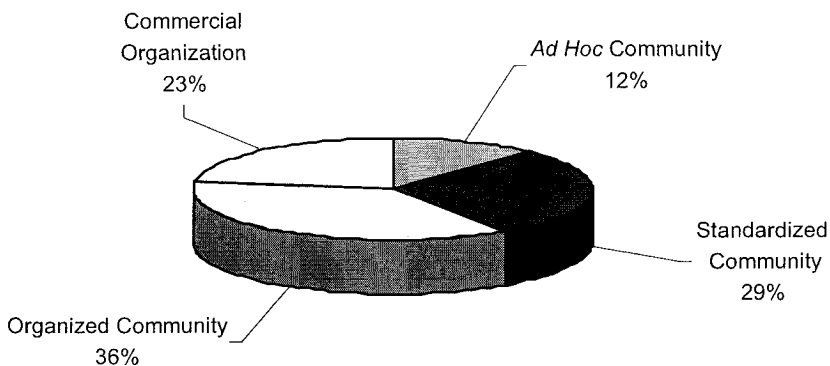

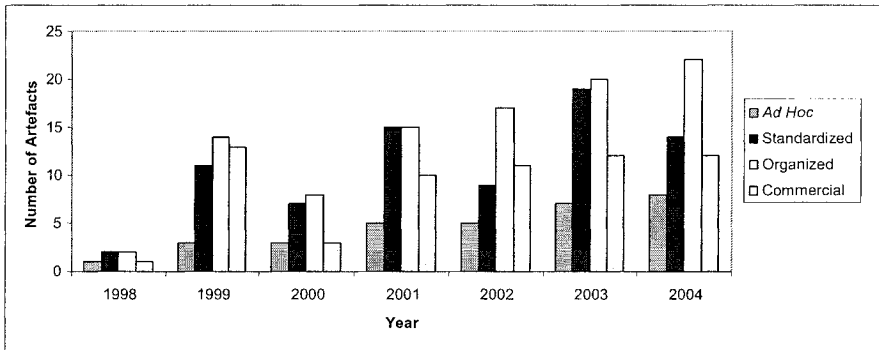*Figure 1.* Overall Percentage of Artefacts Classified by Community

Figure 2. Yearly Analysis of Communities Researched

As shown in Figure 3, half of the research artefacts focused on socio-cultural and organizational issues, and nearly one-third focused on software engineering issues. This is a provocative finding given the strong software engineering/information systems nature of the conferences and journals in which the artefacts were published/presented. While more extensive analysis is needed, it would appear that the research community finds the collaborative "human" aspect of OSS to be more "research-worthy" than the technical aspects. Figure 3 also shows that a roughly equal number of artefacts focused on economic and business model issues and software application spaces (approximately 10 percent each). These artefacts within the area of software application spaces tend to be more recent, as illustrated in Table 6, and are arguably evidence of a shift in focus from OSS production to OSS consumption and exploitation.
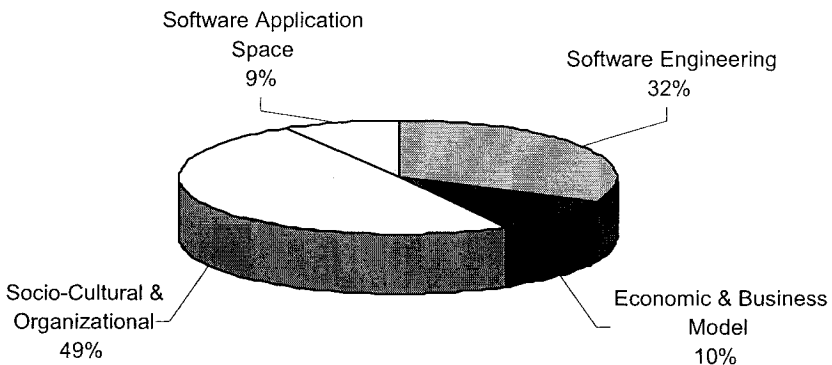

Figure 3. Overall Percentage of Artefacts Classified by Research Area

Table 6. Yearly Analysis of Research Areas

Year	Software Engineering	Economic & Business	Socio-Cultural & Organizational	Application Space	n
1998	0%	33%	67%	0%	3
1999	9%	26%	56%	9%	23
2000	43%	0%	50%	7%	14
2001	20%	20%	48%	12%	25
2002	41%	4%	42%	13%	24
2003	34%	13%	25%	28%	32
2004	33%	12%	40%	15%	33

Table 7 presents a more detailed view of the data, highlighting intersections between the OSS community types and the areas of research. Within the *ad hoc* communities space, there is no research focused on economic and business model issues and software application spaces; rather, the research is divided between socio-cultural and organizational issues and software engineering issues, in roughly the same proportions as the overall collection. Within the standardized and organized communities categories, we again see the same dominance of socio-cultural and organizational issues and software engineering issues, with a gradual increase in the other research areas. This is not particularly surprising, as much of the research on economic and business model issues and software application spaces requires projects and products to be of a much higher level of complexity than can be found in *ad hoc* communities. Finally, the commercial organizations space breaks free from the overall distribution pattern discussed previously, with all four research areas represented in a more-or-less even way. Unsurprisingly, research on economic and business model issues dominates this space.

Figure 4 provides an overview of the research methods used to study OSS development communities. Most significant here is the identification that the dominant form of research present within the area is informal work, with 42 percent of research to date conducted using anecdotal or descriptive methods. While there has been some fluctua-

Table 7. Count by Community/Focus Intersection

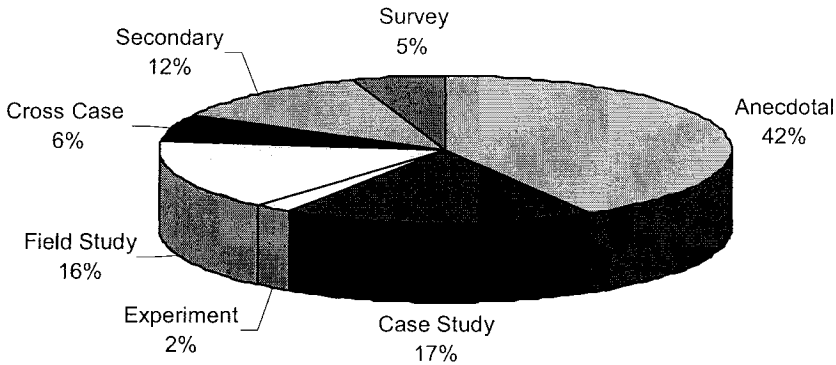|	*Ad Hoc* Community	Standardized Community	Organized Community	Commercial Organization	Subtotals
Software Engineering	13	28	31	14	86
Economic & Business Model	0	2	7	18	27
Socio-Cultural & Organizational	19	43	53	18	133
Software Application Space	0	4	7	12	23
Subtotals	32	77	98	62	**269**

Figure 4. Overall Percentage of Artefacts Classified by Research Method

tion in the amount of informal work conducted between 1998 and 2004, it forms the basis for much of the OSS body of knowledge. Such findings are perhaps illustrative of the relative immaturity of the field.

Within the research on *ad hoc* communities, the sampling approach dominates; anecdotal data and broad, shallow field studies are the most common, with only a few deeper single case studies. As we move into the standardized communities, there is an increase in case study based research, particularly in the area of software engineering issues. To a certain extent, this can be explained in terms of accessibility: case study research requires a clearly bounded subject, and this is more readily accomplished with standardized communities than with *ad hoc* communities. Also, by definition, standardized communities display more concrete "researchable" software engineering practices. The trend toward increasing depth of research continues with the work focused on organized communities, where we see very strong single and cross-case analysis. There is also a marked increase in the gathering and analysis of quantitative data. Again, this may be due to accessibility: the stability of organized communities provides more opportunity for this type of research. Interestingly, and importantly from the point of view of the gap-analysis, within the commercial organizations space, we see a certain swing back toward anecdotal data collection and analysis.

The low number of in-depth, empirical research artefacts in the areas of both economic and business model issues and software application spaces stand out as noteworthy findings when the intersection of topic and methodology is reviewed (Table 8). When viewed within the context of increasing levels of commercial interest in OSS, the dearth of established research in these areas is particularly significant.

4   CONCLUSION

Based on our analysis, we argue that the OSS research literature requires greater discipline and rigor—deeper research, more quantitative data, and more robust cross case-analysis. There is also a need for greater understanding of the similarities and differences between community types (cross-community analysis) and for more inter-

Table 8. Research Topic/Research Methodology Intersection

	Software Engineering	Economic & Business Model	Socio-Cultural & Organizational	Software Application Space
Anecdotal	11	14	26	16
Case Study	14	2	11	4
Experiment	1	0	1	2
Field Study	9	4	7	0
Cross Case	4	1	5	0
Secondary	3	0	10	1
Survey	2	0	6	0

disciplinary research (cross-topic analysis). Building on these deeper descriptions, the research community could then address the more fundamental gap in our knowledge, namely the relative lack of robust models and theories.

It is evident that commercial organizations are underrepresented in the research, not just in terms of quantity, but more importantly in terms of depth of research. In addition, our understanding of economic and business models and software application spaces is also quite limited. These are critical gaps in the body of OSS knowledge in the context of social inclusion, as commercial and community OSS groups need to understand each other to realize the social inclusion potential of OSS highlighted in the introduction to this paper. Thus, we argue that future OSS research must address the convergence—and potential conflict—of the goals of the different communities that constitute the open source environment: individual and organizational users, software developers "in the wild," in nonprofit organizations, and in commercial firms, and policy makers seeking to make sense of the role of software in the wider information society.

Acknowledgments

References

Alvai, M., and Carlson, P. "A Review of MIS Research and Disciplinary Development," *Journal of Management Information Systems* (3:4), 1992, pp. 45-62.

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

Brown, W. A., and Booch, G. "Reusing Open-Source Software and Practices: The Impact of Open-Source on Commercial Vendors," in C. Gacek (ed.), *Software Reuse: Methods, Techniques, and Tools*, New York: Springer-Verlag, 2002, pp. 123-136.

Capiluppi, A., Lago, P., and Morisio, M. "Characteristics of Open Source Projects," in *Proceedings of the 7ᵗʰ European Conference on Software Maintenance and Reengineering*, Los Alamitos, CA: IEEE Computer Society Press, 2003, pp. 317-327.

Chen, W., and Hirschheim, R. "A Paradigmatic and Methodological Examination of Information Systems Research from 1991 to 2001m" *Information Systems Journal* (14:3), 2004, pp. 197-235.

Claver, E., Gonzales, R., and Llopis, J. "An Analysis of Research in Information Systems," *Information and Management* (37:4), 2000, pp. 181-195.

Culnan, M. J. "Mapping the Intellectual Structure of MIS, 1980-1985: A Co-Citation Analysis," *MIS Quarterly* (11:3), 1987, pp. 340-353.

DiBona, C., Ockman, S., and Stone, M. *Open Sources: Voices from the Open Source Revolution*, Cambridge, MA: O'Reilly and Associates, 1999.

Dinkelacker, J., Garg, P. K., Miller, R., and Nelson, D. "Progressive Open Source," in *Proceedings of the 24ᵗʰ International Conference on Software Engineering*, Orlando, FL, May 19-25, 2002, pp. 177-186.

Doty, D. H., and Glick, W. H. "Typologies as a Unique Form of Theory Building: Toward Improved Understanding and Modeling," *Academy of Management Review* (19:2), 1994, pp. 230-251.

During, B "Agile Methods in the PyPy Project," paper presented at the Second International Calibre Conference Limerick, Ireland, September 9, 2005.

Farhoomand, A .F. "Scientific Progress of Management Information Systems," *DataBase* (18:4), 1987, pp. 48-56.

Farhoomand, A. F., and Drury, D. H. "A Historiographical Examination of Information Systems," *Communications of the Association for Information Systems* (1:19), 1999.

Feller, J., and Fitzgerald, B. *Understanding Open Source Software Development*, Harlow, UK: Pearson Education Ltd., 2002.

Feller, J., Fitzgerald, B., Ljungberg, J., and Berquist, M. "Open Source and Free Software: Organizational and Societal Implications," in M. Korpela, R. Montealegre, and A. Poulymenakou (eds.), *Organizational Information Systems in the Context of Globalization*, Boston: Kluwer Academic Publishers, 2003, pp. 461-464.

Gacek, C., and Arief, B. "The Many Meanings of Open Source," *IEEE Software* (21:1), 2004, pp. 34-40.

Ghosh, R. A., and Prakash, V. V. "The Orbiten Free Software Survey," *First Monday* (5:7), 2000.

Glass, R. L., Vessey, I., and Ramesh, V. "Research in Software Engineering: An Analysis of the Literature," *Information and Software Technology* (44:8), 2002, pp. 491-506

Hars, A., and Ou, S. "Working for Free? Motivations of Participating in Open Source Projects," in *Proceedings of the the 34ᵗʰ Hawaii International Conference on System Sciences* (Volume 7), Los Alamitos, CA: IEEE Computer Society Press, 2001, p. 7014.

Hertel, G., Niedner, S., and Herrman, S. "Motivation of Software Developers in Open Source Projects: An Internet-Based Survey of Contributors to the Linux Kernel," *Research Policy* (32:7), 2003, pp. 1159-1177.

James, J. *Bridging the Global Digital Divide*, Cheltenham, UK: Edward Elgar Publishing Ltd.., 2003.

Koch, S. *Free/Open Source Software Development*, Hershey, PA: Idea Group Publishing, 2004.

Lerner, J., and Tirole, J. "The Simple Economics of Open Source," Working Paper 7600, National Bureau of Economic Research, Cambridge, MA, 2000.

Lerner, J., and Tirole, J. "The Scope of Open Source Licensing," *The Journal of Law, Economics, and Organization* (21:1), 2005, pp. 20-56.

Lussier, S. "New Tricks: How Open Source Changed the Way My Team Works," *IEEE Software* 21:1), 2004, pp. 68-72.

McKelvey, B. "Organizational Systematics: Taxonomic Lessons from Biology," *Management Science* (24:13), 1978, pp. 1428-1440.

Mockus, A., Fielding, R., and Herbsleb, J. D. "A Case Study of Open Source Software Development: The Apache Server," in *Proceedings of the 22nd International Conference on Software Engineering*, New York: ACM Press, 2000, pp. 263-272.

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

Moon, J. Y., and Sproull, L. "Essence of Distributed Work: The Case of the Linux Kernal," *First Monday* (5:11), 2000.

Nakakoji, K., Yamamoto, Y,, Nishinaka, Y,, Kishida, K., and Ye, Y. "Evolution Patterns of Open-Source Software Systems and Communities," in *Proceedings of the International Workshop on Principles of Software Evolution*, New York: ACM Press, 2002, pp. 76-85.

Orlikowski, W., and Baroudi, J. J. "Studying Information Technology in Organizations: Research Approaches and Assumptions," *Information Systems Research* (2:1), 1991, pp. 1-28.

Ramesh, V., Glass, R. L., and Vessey, I. "Research in Computer Science: An Empirical Study," *Journal of Systems and Software* (70:1-2), 2004, pp. 165-176.

Raymond, S. E. *The Cathedral and the Bazaar*, Sebastapol, CA: O'Reilly and Associates, 1999.

Romano, N. C., and Fjermestad, J. "Electronic Commerce Customer Relationship Management: An Assessment of Research," *International Journal of Electronic Commerce* (6:2), 2002 pp. 61-113.

Ruffin, M., and Ebert, C. "Using Open Source Software in Product Development: A Primer," *IEEE Software* (21:1), 2004, pp. 82-87.

Steinmueller, W. E. "ICTs and the Possibilities for Leapfrogging by Developing Countries," *International Labor Review* (140:2), 2001, pp. 193-210.

Vessey, I., Ramesh, V., and Glass, R. L. "Research in Information Systems: An Empirical Study of Diversity in the Discipline and its Journals," *Journal of Management Information Systems* (19:2), 2002, pp. 129-174.

Vessey, I., Ramesh, V., and Glass, R. L. "A Unified Classification System for Research in the Computing Disciplines," *Information and Software Technology* (47:4), 2005, pp. 245-255.

Von Hippel, E. *Democratizing Innovation*, Cambridge, MA: MIT Press, 2005.

Von Hippel, E., and von Krogh, G. "Open Source Software and the 'Private-Collective' Innovation Model: Issues for Organization Science," *Organization Science* (14:2), 2003, pp. 209-223.

West, J., and Dedrick, J. "Proprietary vs. Open Standards in the Network Era: An Examination of the Linux Phenomenon," in *Proceedings of the 34th Hawaii International Conference on System Sciences* (Volume 5), Los Alamitos, CA: IEEE Computer Society Press, 2001, p. 5011.

Woods, D., and Guliani, G. *Open Source for the Enterprise*, Sebastapol, CA: O'Reilly Associates, 2005.

Wu, M., and Lin, Y. "Open Source Software Development: An Overview," *IEEE Computer* (34:6), 2001, pp. 33-38.

Ye, Y., Kishida, K., Nakakoji, K., and Yamamoto, Y. "Creating and Maintaining Sustainable Open Source Software Communities," in *Proceedings of International Symposium on Future*

Software Technology, Software Engineers Association, Wuhan, China, October 23-26, 2002 (CD-ROM).

Yee, D. "Development, Ethical Trading and Free Software," *First Monday* (4:12), 1999.

About the Authors

Joseph Feller is a senior lecturer in Business Information Systems, University College Cork, Ireland. His work on open source software includes coauthorship of two books (*Perspectives on Free and Open Source Software*, The MIT Press, 2005, and *Understanding Open Source Software Development*, Addison-Wesley, 2002) as well as international conference and journal papers. He coauthored "A Framework Analysis of the Open Source Software Development Paradigm," which was awarded Best Paper on Conference Theme at the 21st International Conference on Information Systems (ICIS 2000). Joseph was the lead organizer of the IEE/ACM workshop series on Open Source Software Engineering from 2001 through 2005, and has been a speaker and panelist on the topic at academic conferences, industry workshops, and European Commission briefings and roundtables. Joseph is a member of the EU FP6 Coordination action project CALIBRE (www.calibre.ie), co-leading the dissemination and awareness work package and conducting research on open source software business models. He can be reached at jfeller@afis.ucc.ie.

Patrick Finnegan holds a Ph.D. in Information Systems from the University of Warwick, and is currently a senior lecturer in Management Information Systems at University College Cork. His research interests include electronic business and IS strategy. He has published his research in a number of international journals and conferences including *The International Journal of Electronic Commerce, Information Technology & People, DataBase, Electronic Markets, The Information Systems Journal*, the European Conference on Information Systems, the International Conference on Information Systems, and the Americas Conference on Information Systems. Patrick can be reached at pfinnegan@afis.ucc.ie.

David Kelly is currently pursuing a Ph.D. in Information Systems at University College Cork. His research interests include innovation adoption, ambient intelligence, standards, and open-source software. David can be reached at d.kelly@ucc.ie.

Maurice MacNamara is currently pursuing a Ph.D. in Information Systems at University College Cork. His research focuses on the collaborative development of software systems for knowledge sharing. Maurice can be reached at mmacnamara@bismail.ucc.ie.