11-3-2008

Open Source Software Collaboration: Foundational Concepts and an Empirical Analysis

Charles M. Schweik
University of Massachusetts Amherst

Robert English
University of Massachusetts Amherst

Sandra Haire
University of Massachusetts Amherst

Open Source Software Collaboration: Foundational Concepts and an Empirical Analysis

Charles M. Schweik [1,2,3]
Robert English [1,2]
Sandra Haire [3]

[1] *National Center for Digital Government*
[2] *Center for Public Policy and Administration*
[3] *Department of Natural Resources Conservation*
University of Massachusetts, Amherst

NCDG Working Paper No. 08-002

Submitted November 3, 2008

This material is based upon work supported by the National Science Foundation under Grant No. 0131923. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

This paper was prepared for the Minnowbrook III Part II conference, Syracuse University Maxwell School

> "Public managers now find themselves not as unitary leaders of unitary organizations. Instead, they find themselves facilitating and operating in multi-organizational arrangements ***to solve problems that cannot be solved, or solved easily, by single organizations***. In many instances, the needed skill set of public managers has changed to one that heavily emphasizes ***collaborative problem solving*** and negotiation. These skills have become increasingly important both for network management purposes and as public managers strive to become more deliberative and ***inclusive***. "
>
> -- Minnowbrook III Conference Website (our emphasis)

Introduction

One of the reasons we were particularly excited about participating in the Minnowbrook III conference is that, as the title of the conference suggests, the central overarching theme is the *future* of public administration and public management. The research that we present in this paper is not about what was, or even really what is; rather, it is about what could be as it relates to the Internet and collaboration. These days we often hear phrases like "the Network Society," Networked Governance," "Collaborative Public Management," the "Conductive" public organization, etc., which signals real interest in collaboration in general, and Internet-based collaboration more specifically.

This paper has three primary goals. First, we provide an overview on some foundational concepts – "peer-production," "user-centric innovation," "crowdsourcing," "task granularity," and yes, open source and open content – for they are key elements of Internet-based collaboration we see today. Second, through this discussion on foundational concepts, we hope to make it clear why people interested in collaborative public management and administration *should* care about open source and open source-like collaboration. After this argument is made, we provide a very condensed summary of where we are to date on open source collaboration research. The goal of that research is to learn about the factors that lead

to successful or abandoned collaborations in the open source domain, in part to help us understand how "open source-like" collaborations can be deployed in areas outside of software. We have a lot to cover. Let's get right to it.

Foundational Concepts: Peer-Production, "Open Source and Open Content," "User-Centered Innovation," "Crowdsourcing," and "Task Granularity"

For a number of years, I (the lead author) have been trying to "wrap my head around" the rapidly changing phenomena we call the web, and its implications for collaboration. This section summarizes some of the important elements I see after several years of reflection.

Peer Production

We imagine almost everyone at Minnowbrook III is aware of the web search engine Google.com, or the online bookstore Amazon.com. Perhaps a slightly smaller number are familiar with the web-based encyclopedia Wikipedia, or the video sharing site YouTube.com. An even smaller group may be users of social networking websites like MySpace.com and FaceBook.com, and a roughly equal number may use news sharing sites like Digg.com and Slashdot.com or the web-bookmarking site called "del.icio.us." Regardless of whether my estimates are right, my point is that all of these are examples of high profile websites – built upon what is now commonly referred to as "Web 2.0" technologies – where users interact to some degree with the site, rather than just read static text.

In other words, what these sites and others like them have in common is that they harness the productive power of their users. Yochai Benkler (2006) refers to this as "Commons-based Peer Production." To Benkler, Peer-Production describes a special kind of production system where individuals act in response *to their own needs and interests and in a*

decentralized manner. In the case of Google, users are actively searching for things they want to find for whatever work they are doing. But behind the scenes, Google's PageRank algorithm uses the hyperlinks created by individual web authors as a "vote" for the importance of such pages (Google, 2008). (We'll return to this in the discussion below on task "granularity").

A similar situation exists with Amazon.com. Users, based on their own self interests, actively look and purchase books. But as this is done, Amazon's technology keeps a database of the kinds of books that you bought and, based on that data, provides recommendations of other books you might like based on the purchase history of others. The PageRank technology in Google and the book recommendation system in Amazon are examples of efforts to employ the work of end users who are doing tasks motivated by their own interests to create systems of accreditation and relevance (Benkler, 2006).

The video sharing site Youtube.com is more interactive (at least compared to Google's search system), in that it not only allows people to search and view video (keeping track of how many people watch each one), but it also relies on end users to provide YouTube with actual content (new videos). This is true as well with some of the other high profile sites we mentioned earlier. MySpace, FaceBook, Digg, Slashdot, and others all rely on this idea of peer-production.

The other important attribute of commons-based peer production besides the fact that they rely on users doing things that interest them for content, is that these efforts thrive in "crowd-like" (Surowiecki, 2004) situations where a huge number of potential users exist. Most, if not all, of the high-profile websites I've listed above have users providing content from across the globe. This leads me to the next foundational topic: open source.

Open Source and Open Content

For those who may unfamiliar, open source is a term that describes a phenomenon that began in the mid-1980s that has occurred in computer programming.[1] To summarize greatly, open source differs from traditional proprietary software in that the computer source code – the internal logic of the program – is made available for anyone to access and read. This differs substantially from proprietary code that is delivered in a binary format that only computers can read. The great innovation made in the early days of open source (what then was called "free/libre" software) was its innovative use of copyright law, a concept sometimes referred to as "copyleft" (Deek and McHugh, 2008). A copyleft license provides the user with the right to copy, modify and redistribute new derivatives of that software, but mandates that the derivative be licensed the same as its "parent" software. This, in and of itself, was a great innovation, and has inspired others to develop similar licenses for digital products other than software. The most famous of these are the Creative Commons licenses developed by intellectual property scholar Lawrence Lessig and others with the organization of the same name (www.creativecommons.org). Creative Commons licenses are now ubiquitous on the net, attached to products such as papers, images, music, and photographs.

Benkler (2006: 63) refers to open source software collaboration as the "quintessential instance of commons-based peer production." As he puts it, open source "depends on many individuals contributing to a common project, with a variety of motivations, and sharing their respective contributions without a single person or entity asserting rights to exclude either from the contributed components or from the resulting whole." The problem we have had with Benkler's depiction of open source as peer-production is the issue of team size. Several recent studies, beginning with Krishnamurthy (2002) and including one of our own (Schweik

[1] For a history of open source, see Weber (2004). For relatively up-to-date and more detail on the subject, see Deek and McHugh (2008)

and English, 2007), have shown that most open source projects are usually small teams. Open source projects do not have massive teams of contributors like the websites above. This is an important point as to why open source peer production is potentially important for public sector collaboration that we will return to in the conclusion of this section of the paper.

User-Centered Innovation.

In addition to the copyleft licensing innovation, there are two other surprising points to make related to the open source phenomenon. First, at least until about five years ago, the majority of software developed (which was a sizable amount) was written by *volunteer* developers. These were people with technical skills, who wrote software in their free time, and who may or may not have been gainfully employed. Stebbins (2001) refers to this concept as "serious leisure," a term he coined back in 1982 before the idea of open source existed. Now, however, more developers are *paid,* as businesses, governments and nonprofit organizations have entered the open source game, leading to a significant change in the composition of the "open source participant ecosystem."

The second important point is that historically, the majority of the software produced was developed by programmers who are also *users* of the software (von Hippel and von Krough, 2003).[2] The idea of users as innovators, adds significantly to the incentives driving people to contribute, as well as the quality of their contributions (von Hippel, 2005a). The existence of open source collaborations as "user-centered innovation networks" (von Hippel, 2005b), is somewhat a surprise to many, in that these innovators would freely reveal their innovations. But the open source community demonstrates that this indeed happens, and in a major way.

[2] My sense is that at this point in time, the emphasis on developers being users may not be as strong, as open source matures.

Research over the last 5 years has helped to explain the incentives that drive volunteer contributors to behave this way (Lakhani and Wolf, 2005; Ghosh, 2005). Solving a specific need (the user centric component) is one common motivation. Others include the enjoyment of a challenging problem (serious leisure), learning and skill building through the collaboration with others, and signaling skills to others for ego gratification or possible future job opportunities. In addition, recent studies by Krishamurthy (2005), Riehle (2007) and Deek and McHugh (2008: 272-279) show how firms are making a profit using a business model built around or upon open source products. For example, there are businesses who (1) build complete systems to solve a client need (system integrators); (2) provide technical support services; (3) distribute open source products; (4) create new software products built with open source components; or (5) dual-license their software (one open source, one proprietary). For our purposes a detailed understanding of these business models is not important. What is important is that these businesses also have their own "user-centric" needs, and as a result are increasingly committing their own resources (e.g., employees, monetary donations) to open source projects.

Crowdsourcing.

The idea of business needs leads us to another relevant concept, called "crowdsourcing" (Howe, 2006a). Howe (2006b) defines it this way: "Crowdsourcing is the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call." In other words, a company posts a problem they are facing on the Internet, individuals submit solutions, winning ideas are rewarded, and the company mass produces the innovation for profit (Brabham, 2008). Crowdsourcing is an idea that tries to capture the idea of mining ideas from large groups of people, as highlighted by James Surowiecki in his 2004 book *The*

Wisdom of Crowds. The idea, in its current form, embraces the peer-production and user centric innovation concepts, but differs from open source in that the request for help comes from a firm and the innovation becomes their product, compared to open source where the product remains in the public domain (Brabham, 2008a). In a more recent study of participants in iStockphoto.com (described below), Brabham (2008b) finds that, like open source participants, they are motivated by enjoyment and fun, but also, naturally, by the prospect of making money. However, unlike what is thought to be true in open source, they do not appear to participate for peer recognition or to build a network of collaborators. So, there appear to be motivational differences between crowdsourcing and open source. Crowdsourcing also differs from peer-production efforts like Wikipedia, and the other web examples above, in that in the former an organization is creating a kind of contest for help, whereas in the latter, the actions are driven solely by the user's own interests and motivations.

Recently a number of crowdsourcing efforts have emerged. Newly established firms now try to match challenging research and development problems that other companies have to individuals capable of solving those problems (e.g., InnoCentive, http://innocentive.com/; Innovation exchange, http://www.innovationexchange.com/). Threadless.com, a tee-shirt company, allows end users to submit tee shirt designs and vote on submitted entries. Monetary rewards are given to the submitters if a design is accepted. IStockphoto.com, mentioned earlier, provides another example where photographers upload and sell their images for use by others in almost anything – brochures, websites, business presentations, etc. The photographer is given 20 percent of the purchase price every time their image is downloaded (Brabham, 2008).

Interestingly, the NASA Clickworkers project (NASA, 2001) is an example of a crowdsourcing-type effort that was a precursor to all of these – and driven by a government

agency's needs with no monetary reward attached. In Clickworkers, volunteers were solicited to help digitize and categorize craters found on images of the Martian surface, taking advantage of serious leisure amateur astronomers. The initial project was successful enough to lead to a second such effort which began in 2007 (Nasa, 2008). Other examples of non-monetary compensation peer production crowdsourcing efforts have emerged as well. One, similar to NASA's Clickworkers, is the Digital Proofreaders (DP) project (DP, 2008), which asks volunteers to help digitize books in the public domain. Serious leisure volunteers utilize a web-based interface to compare one scanned page with the same digital text read by an optical character recognition reader, and to spot and to fix problems with the character recognition process. (The "one page at a time" concept is important and relates to the idea of granularity discussed below.)

My favorite of these kinds of examples is the ReCAPTCHA project (ReCAPTCHA, 2008), which, like DP is an effort to convert scanned images of books into digital text e-books, but at the same time, simultaneously helps protect interactive websites (and email addresses) from spam. ReCAPTCHA is the name for a small bit of software code that can be added to interactive websites which is invoked when the user is entering in some information into the website. Similar to DP, ReCAPTCHA requests the user to prove he or she is a human and not an Internet spam "bot" by having them read two scanned words that could not be interpreted correctly by an OCR reader, and type them in. ReCAPTCHA software collects these two typed words for the new digital text version of the book (ReCAPTCHA, 2008).[3] The people who really have the incentive to use reCAPTCHA are not the end-users of websites but the webmasters who want to protect their systems from spam (although indirectly, this helps the users of their website as well). In other words, ReCAPTCHA's in a way gets "forced"

[3] One reason that digital or ASCII text is better than scanned pages is that the digital text takes up less computer memory, making it easier to be used in equipment like e-book readers.

volunteers through webmaster's concerns for Internet security. But like the others, it takes advantage of peer production or a kind of crowdsourcing, to get a problem solved.

One thing related to crowdsourcing that is now becoming apparent is that if it isn't employed in a carefully planned way, it can potentially produce lots of data or products that are not helpful. One such example was the deployment of a crowdsourcing effort on the Amazon.com's Mechanical Turk project (http://www.mturk.com/mturk/), which is an effort to match up people who want to do small tasks in their spare time for pay per task. In 2007, this platform was used to harness the labor of as many as 50,000 volunteers to look through, online, aerial photo images of Nevada for plane wreckage of adventurer Steve Fossett. The effort led to a significant number of false leads sent to the search coordinator, with no helpful results (Friess, 2007).[4]

Recently, Lukensmeyer and Torres (2008) proposed the idea of applying the idea of crowdsourcing to government citizen engagement efforts. They acknowledge there are several reasons to be cautious or skeptical (p. 218). First, citizens are more sensitive when it comes to privacy when dealing with their government. Second, government problems are often more challenging compared to problems found in the private sector. Third, getting acceptance of government agencies toward these kinds of innovative practices is harder than in the private sector. Fourth, the present "policy framework" for citizen engagement and its potential reform moves at a glacial pace, making it hard to implement such a radical idea. But they also note one reason to forge ahead: the gap between how citizens and industry use the Internet and government will continue to widen, leaving a disenchanted citizenry. The authors

[4] In this research we read through some comments to the story posted by Freiss (2007) by actual participants who conducted the search about this outcome. Some were arguing that the crowdsourcing idea was a good one, but from their perspective it wasn't implemented correctly, for example, in terms of the instructions that were provided. Another point was that endusers who were doing the searching were bypasses a kind of "chain of command" when they found what they thought was a lead. Instead of contacting the Mecahnical Turk people or the people that funded the project, they were contacting the search and rescue official directly via email and phone.

emphasize this point with the example that a very useful peer production-like application during the Katrina hurricane disaster, called "Peoplefinder," a relatively simple "GoogleMaps mashup" application to help people locate family and friends, was implemented by a company rather than a government agency.

Task Granularity.

The final foundational concept we wish to introduce is "task granularity," which is embedded in all of the examples we have discussed so far. Benkler (2006: 107) reminds us that in terms of systems of production, we face two primary scarcities: (1) human creativity, time and attention, and (2) the computation and communication resources used in information production and exchange. Computing and the Internet, of course, have greatly reduced the cost of the latter. But this hasn't changed the fact that human creativity, time and attention, is a scarce commodity. We all are constantly making decisions about how we use our work and leisure time. This is why the concept of task granularity is especially important.

Task granularity refers to "the size of the modules, in terms of the time and effort that an individual must invest in producing them" (Benkler, 2006: 100). It is an important concept in peer-production commons, because it influences people's decisions on whether to contribute or not. Task granularity "sets the smallest possible individual investment necessary to participate in a project," and "if this investment is sufficiently low, then 'incentives' for producing that component of a modular project can be of trivial magnitude" (Ibid).

*** Table 1 about here ***

In Table 1, we provide some common tasks found in some of the peer production websites referenced earlier. In this table we build upon Benkler's granularity concept by

introducing a 5 category ordinal scale. At one end, is the “extremely coarse grained task,” which will require the participant to use a large amount of his or her time. Examples we provide are taking the lead author role in a Wikibook (collaborative writing of an entire book using wiki technology as the authoring mechanism), or participating as a lead developer in an open source programming project. These kinds of tasks will require a significant time commitment over a substantial time period. At the other end of the ordinal scale is a term we call the “Transparently Grained” task. These are tasks that peer production participants undertake unknowingly; that is, the technical infrastructure they are using takes advantage of the information they are providing to create new information that is useful for another purpose.

This brings us back to the examples that opened this section of the paper. As we mentioned earlier, the Google PageRank search formula capitalizes on web page authors’ use of hyperlinks. Web page authors don’t link to other pages to help Google’s search system operate better. They place hyperlinks on their web pages because, for some reason, it is useful for their own purposes. It is this work that the web authors do for their own self interest that Google capitalizes on. Google reads these hyperlinks and builds PageRanks from that information. A similar situation exists with Amazon.com. Their site keeps track of the buying activities of other users undertaking their own self-motivated book purchases, and uses this information they capture to recommend books you might want to read. Amazon users don’t buy books to help Amazon make recommendations to others. But Amazon takes advantage of the situation by collecting and managing that information. The same “transparently grained” type tasks exist in the ReCAPTCHA or DP examples we described earlier as well. Tasks that have transparent granularity are ones requiring (usually) very small bits of time and are accomplished by technology taking advantage of work you would do anyway.

Table 1 provides examples of tasks that fall between the two extremes of "extremely coarse grained" peer production tasks and "transparently grained" tasks. We'll leave it to the reader to review these other examples, and will end this discussion with one conjecture related to granularity: Peer production efforts that can modularize and create fine-scale or no grained tasks will have a higher likelihood of success compared to ones that require coarse-grained efforts.

By working through these fundamental concepts – peer production, open source and open content, user centered innovation, crowdsourcing, and task granularity – we had two goals. First, we wanted to shed some light to readers about these important concepts related to peer production and major Internet-related collaborative technologies most of us interact with in some form. Second, we wanted to create a foundation that would allow me to emphasize a point we have made previously (see Schweik and Semenov, 2003; Schweik, Evans and Grove, 2005) but perhaps with more clarity. Lukensmeyer and Torres (2008: 219) articulate the same idea very nicely as it pertains to crowdsourcing and citizen engagement:

> "By deftly harnessing the creativity that is unleashed when people come together informally and around shared interests and passions, crowdsourcing offers a dynamic, complex and emergent model of public problem solving."

While we see great value in Lukensmeyer and Torres' articulated vision,[5] our interests diverge from Lukensmeyer and Torres' goal of achieving peer-production based citizen engagement. We agree with them when they acknowledge there are great challenges ahead to implement those kinds of collaborations, with privacy issues perhaps at the top of the list. My interests are to investigate whether "scaled down" peer production is possible, with a

[5] Although we readily admit that, like any area where we talk about government-citizen interaction using computing networks, this does raise broader questions related to equity and voice issues centered around the digital divide issue.

focus on cross-agency, inter-governmental or "epistemic community" (Haas, 1992) collaboration in the public sector.

As we noted earlier, open source collaboration is the peer production area with the longest track record, and the area that is not dependent on huge numbers of people to get work done. In other words, open source collaborations are typically not the work of crowds. The majority of open source peer production efforts are smaller teams of like-minded individuals working toward a common vision. This said, and in the spirit of new derivative works found in open content communities, we'd like to modify the above quote by Lukensmeyer and Torres to say this (our revisions are in italics):

> By deftly harnessing the creativity that is unleashed when people come together informally and around shared interests and passions, *principles of open source and open content collaboration* offer a dynamic and emergent model of public problem solving *in policy, administration and management*.

A Vision of What Could Be:
Inter-Governmental Peer Production in Urban Simulation Modeling

Let us now close this section of the paper with an example of *what could be*. Several years ago, we put some coarse-grained effort in trying to encourage collaboration between local governments in urban simulation modeling (see Schweik, Evans and Grove, 2005 for more discussion). Given our interest in landuse change issues, and open source and open content, we discovered a terrific open source urban simulation project called "UrbanSim," developed by an interdisciplinary team at the University of Washington. The UrbanSim model is being used by a number of major cities in the US, and internationally, to assist policy makers, analysts, and we expect, public managers in urban planning. There was already a substantial user community of local government participants that existed, who were implementing UrbanSim models for their respective local jurisdictions. They represent an epistemic community (Haas, 1992) in urban simulation modeling specifically, and urban policy

and management more generally. This, and the idea that UrbanSim was an open source licensed model, intrigued us in its collaborative potential.

After attending the first UrbanSim users conference (approximately 2004), it became apparent that there were substantial pockets of knowledge in the user community that, if captured in some way, would be helpful to others in other cities who are less up to speed on the use of the model or in the database construction process required for the model. In other words, new participants could likely learn a great deal from the experiences of other analysts in other cities who had already done what they wanted to do. In that meeting, we asked the group of 30 or more from around the country if they'd be willing to participate in an "open content" effort to share experiences in UrbanSim modeling. Most in the room gave me the impression that they saw a potential benefit in the idea, although visually a few appeared lukewarm to the proposal.

Armed with a small bit of funding from the Forest Service, we set out to build an open content platform, a wiki[6], that we called the UrbanSim Commons, with the goal of trying to get people to contribute modeling-related documentation any locality had already in digital form, or any information or simple nuggets of wisdom they were willing to write up and enter into the wiki. On our end, we set up the wiki platform, helped a few willing participants from cities create their own wiki page, which described their goals and where they were in the modeling process, and simply tried to act as an editor of the wiki and offered any assistance we could (such as converting documents to wiki format). After about a year of trying to mobilize the community in this way, we gave up.

[6] For readers unfamiliar with wiki technologies, these are web-based systems that allow multiple people to edit pages and that record histories of these changes through their web browsers. One of the challenges to some in using this technology is that it is not quite as user friendly as a word processor. Users typically need to know a set of simple codes to create headings, lists, etc., which sometimes creates some resistance.

Now this is an important point. Our giving up should not at all reflect poorly on the UrbanSim project or its community. Both of these remain vibrant and communication between the user community and developers can be readily seen on an almost daily basis on their email listserv. But we did learn from the experience that it wasn't as easy as setting up a communication channel to motivate public sector epistemic communities to communicate and collaborate in a peer production setting. The public sector employees we interacted with during that short time were caring, committed, hard working individuals (as are the UrbanSim developers). At whatever stage these local officials were at, they were trying to implement the model in an effort to better understand the landuse dynamics in the local jurisdictions they served. However, asking them to take time to convert existing documentation, write up new documentation about the knowledge they had, learn wiki technology, take time out of their day to explicitly go to the wiki and enter these information, cumulatively, was too coarsely grained a request for them to take on. There continues to be fine-grained activity all the time on the project listserve. We see back-and-forth emails where they are assisting each other in quick "how to" or problem solving questions and answers.

Moreover – and this is purely conjecture on our part – we doubt that taking precious time at the office to share ideas or help others in other cities are seen by many as a high priority compared to other tasks that were relevant to their immediate jobs and their own jurisdictions. These analysts in cities are not paid to help others in other cities or evaluated on these kinds of outreach activities. Other researchers thinking about collaboration in the public sector lend support to this conjecture. Bardach (1998) argues that getting public organizations to collaborate is difficult. Agranoff (2008) emphasizes the need for tangible benefits to contributing organizations. Lukensenmeyer and Torres (2008) note that incentive structures need to be put in place by the organizations that encourage the collaborative innovation and reward success (Lukensmeyer and Torres, 2008). These are key issues and

real barriers to collaboration that clearly need to be addressed, but are beyond the scope of this paper.

With these challenges acknowledged, let us still try to imagine a peer production system that could support such exchange between local government officials working on similar problems. Let me emphasize that what we are talking about here is not really an effort to take advantage of "serious leisure" participants. What we are describing is an effort to create a peer production system that connects professionals in the workplace.

Imagine a web-based "commons" that provides the functionality to support the free exchange of ideas in an open content (i.e. new derivatives allowed) manner between these kinds of individuals. A place where fine scaled tasks were available, such as posting notes or short articles related to urban transportation issues, or modeling specifically. A place where local modelers can interact with other modelers working in other jurisdictions on similar problems. A place where co-development of new model functionality is possible or the sharing of policy analysis-related documentation could be posted, shared, and have ideas perhaps borrowed and deployed elsewhere. Perhaps even a place that capitalizes on the idea of transparently grained tasks, where new information is collected and fed to others in the course of doing their day to day jobs. To me, this seems like a worthy goal that we should be striving for.

A key question related to this vision is whether peer production systems are "downward scalable." By that we are asking whether the same principles of peer production can be harnessed in smaller group situations. Most of the peer production examples noted earlier have potential user communities in the millions, and all across the globe. In public policy, administration or management settings, we won't enjoy such numbers. In my urban simulation example, there are probably 100's or possibly 1000's of people who might be interested in collaborating. But that still is a fairly large group of potential participants.

Another key question comes down to incentives to encourage workers to contribute to such a peer production commons.

This brings me to the very reason studying open source collaboration is important, and why we include the second half of this paper. Open source projects are a form of peer production that has perhaps the longest history, and also involves collaborations of, for the most part, small teams. These collaborations involve people who are not employed within any one particular organization, and in some instances are from different parts of the world. Moreover, especially in the last five years or so, organizations (firms, nonprofits, even governments) have embraced open source and contributed their own resources (e.g., financial support, paid employee work time, etc.) to the effort.

Some readers will be surprised when we mention this next statistic. One major open source hosting site, Sourceforge.net, now hosts over 130,000 open source projects. However, from a collaboration standpoint, many of these will become abandoned (English and Schweik, 2007). In order to move toward the vision of collaborative peer production in public policy, administration, management, or in almost any other field imaginable, it is important that we learn from the open source software world that, in some ways, leads the peer production effort. The second section of this paper summarizes where we are currently at in such a study.

Findings to Date in our Study of Open Source Collaborations

In 2005 my collaborators and began a study funded by the National Science Foundation to study open source collaborations – what we call “open source commons,” since with their licensing, they are a form of common property regime. The goal of the study is to identify “design principles” that lead these projects toward successful collaborations rather than abandoned efforts. Since that time, we’ve done an extensive review of relevant

theoretical and empirical literature, interviewed of open source developers, and are currently completing quantitative analysis of thousands of open source projects that use the hosting site Sourceforge.net. In this section, we provide a very broad summary of the work we have accomplished so far.

The Open Source Ecosystem – It's not just volunteers anymore…

In our review of what has been occurring in open source in recent years, it is apparent that it is moving from an environment made up of mostly volunteer, serious leisure participants, to one with much more diversity in participant types. The major shift is that more people are participating who are being paid, mostly by firms, but also by government agencies, nonprofit organizations, and academic institutions. In the previous section on User-Centered Innovation, we briefly described the motivations for volunteer developers as well as business models. In the interest of brevity, we won't repeat them here. However, let me very briefly describe the motivations for these other groups in participating in open source commons.

The relationship between government agencies and open source is complex – too complex to do it justice in the space available here. However, it is fair to say that the interest in it is most prevalent outside the United States, but there is a growing interest emerging in the U.S. as well. At least three categories of motivations drive this interest: financial, public good and independence/economic development. First, from the financial perspective, many governments – especially national and state governments – have sizable deployments of IT and through the use of open source alternatives avoid annual licensing fees which can lead to significant cost savings (Muffatto, 2006). Moreover, there are potential cost sharing advantages by collaborating on software projects with other governments or government

agencies (Hamel and Schweik, under review; see also GOSCON.org). Second, we would argue that the force driving public sector interest in open source, at least in the United States, is not its financial benefits but rather its public-good properties. Since about 2003, the debate in the United States has moved from the question of "open source versus proprietary technologies" to the question of "interoperability and open standards." Governments need to be able to seamlessly communicate and share digital information, maintain security in their technologies, and retain the ability to recover archived data over long periods of time. "Interoperable" systems built upon agreed upon "open standards" are critical to meet these needs (Simon, 2005). Third, governments other than the United States have implemented or are considering IT procurement policies that either mandate or show preferential treatment toward open source-based technologies (Maxwell, 2006). In addition to the financial and interoperability reasons, these countries wish to reduce their reliance on foreign software companies, and want to build up their own domestic software industry (Aigrain, 2005; Maxwell, 2006). China is one prominent example (Lewis, 2007). Germany, Italy and Brazil are others.

Nonprofit organizations are thought to participate in open source for financial and public good reasons. A recent survey by the Nonprofit Open Source Institute (NOSI, 2008) reports that open source technologies currently in use by nonprofits are primarily web server technologies (e.g., Apache, MySQL databases, Content Management Systems like Drupal), and desktop applications (e.g., Firefox web browser, Open Office, MySQL) running on proprietary operating systems such as Windows. Interest in saving money through the use of freely available servers and desktop applications motivate these nonprofits to use open source, and likely motivate some of the technicians to participate in certain open source projects. In addition, open source technologies provide opportunities to reuse older

computers for firewalls or for low-level office computing needs (McQuillan, 2008). Peizer (2003), however, rightfully warns that free open source software may not necessarily lead to cost savings. He notes that nonprofits differ from businesses (or some governments, for that matter) in that nonprofits can't as easily recover from a poor choice of technology strategy, and that the total cost of open source in many instances may be as high as or higher than comparable proprietary applications. That said, there are, at least a few, nonprofit organizations who participate in the development of open source software specifically to meet other nonprofit groups' needs. For example, For example, a project called "CivicCRM" (CiviCRM, 2008) is an open source "constituent-relationship management" system that allows a nonprofit to manage fundraising efforts, as well as manage and track volunteers, donors, employees, clients, and vendors. Based on the analysis above, CivicCRM could be classified as a common-property project being coordinated by CiviCRM LLC, with its financing going through the nonprofit "Social Source Foundation" (CiviCRM, 2008). Another potential motivation for nonprofits with sufficient technological expertise to participate in open source development is its "collaborative" and "public good" philosophy, which meshes nicely with what many nonprofit organizations are concerned about as well (McQuillan, 2008).

In addition to the above, nonprofit organizations are involved in open source in a completely different way. Open source projects have established nonprofit foundation organizations to play several support roles: (1) to hold project assets (e.g., software); (2) to protect the team from potential lawsuits; (3) to provide a mechanism to collect and manage fundraising efforts and to interact with outside organizations on the project's behalf; (4) to assist in conflict resolution between participating groups and individuals; (5) to work toward marketing their product; and (6) to protect and enforce property rights related to the code they create (O'Mahony, 2005).

Academic and/or scientific research is the last general category of organizations who are now participating in the open source development space. The motivations here are really a combination of the motivations of the other three categories just described. Some participate because of recent mandates by granting agencies to make software they develop available (Wayner, 1999; U.S. NSF Office of Cyberinfrastructure, 2007). Moreover, some of the technology groups supporting these institutions are now trying to cost share and avoid vendor "lock-in" by collaborating with other academic institutions on software they all need. The Sakai course management system is a prominent example (Sakai, 2008). Finally, it is likely that a significant body of more specialized software to support scientific research is being made available under open source licenses, under the same collaborative philosophy that academic research and publishing is grounded upon. An example of this is the Open Bioinformatics Foundation, a volunteer nonprofit organization with academic participation that tries to encourage collaborative open source software development in the field of Bioinformatics (http://www.open-bio.org).

This short summary of the open source ecosystem has tried to show that open source collaboration has evolved in the last five to ten years from what was originally seen as a mostly all-volunteer environment to one where there is a much more diverse community of interests participating. The graphic in Figure 1 shows this much more complex "ecosystem." From this perspective, open source collaborations have similarities to what we see emerging in the public sector "collaborative governance" environmental management literature (e.g., O'Leary et al., 2006; Koontz et al. 2004.)

Factors Thought to Influence Open Source Collaborations

As part of the research project we mentioned earlier, we have reviewed a sizable amount of theoretical and empirical literature in a variety of disciplines, searching for factors that might contribute to the success or abandonment of open source commons collaborations. We started with the obvious – the traditional information systems development literature – but moved to literature on distributed work and virtual teams, as well as literature on collective action and commons governance and management more specifically. Much of this latter work focuses on collaborations in natural resource commons or common property, but more recently scholars are studying collaborations in "digital commons," such as open access publishing, and open source and open content (van Laerhoven and Ostrom, 2007; Hess and Ostrom, 2007). One of the challenges we face, similar to the study of other commons, is that there are a large number of potentially influential variables (Agrawal, 2002). In this section we provide a very short and generalized overview of the variables we have identified through this process. We organize them into three clusters of attributes: physical, community and institutional (Ostrom, Gardner and Walker, 1994; Ostrom, 2005). They are graphically summarized in Figure 2.

Physical Attributes of Open Source Commons. This phrase refers to the set of variables related to the physical software being developed or some of the technological infrastructure needed to coordinate the team. Our review identified several variables or sets of variables that potentially affect the success or abandonment of open source: (1) software requirements, (2) modularity, (3) product utility, (4) competition, and (5) collaborative infrastructure.

Software requirements refer to the approaches taken for identifying what the software will or should do. It is thought that projects with clearly defined visions will do better than ones

without such visions. *Modularity* has to do with the design of the software, and whether it is easily broken down into separate, relatively standalone components. Within limits, a modular design is thought to make it easier for contributors to "carve off chunks" of the project that they can work on (Weinstock and Hissam, 2005). (Note that modularity has a relationship to the granularity topic we discussed previously.) *Product utility* describes the obvious; that a project will be more successful if the software being produced is something that people want or need (Ibid.). This connects back to the idea of user-centered innovation. *Competition* refers to whether the project is unique in what it is trying to do, or whether there are lots of other similar projects out there. Of course, significant competition would lead to potentially fewer available people or organizations wanting to join in to any particular project. Competition also is included to capture the situation where a rival technology comes along that greatly reduces people's interest in the product being developed. Finally, *collaborative infrastructure* describes the types of technologies used to help coordinate the collaborative team. There are a variety that could be used, including a code version control system, a bug tracking system, and a number of communication and documentation technologies (e.g., email lists, web-based forums, Internet Relay Chat, etc.). The particular configuration may be particularly important in reducing task granularity. For instance, establishing a norm for using a web-based forum for question and answer allows for help to be provided but also searchable documentation to be created over time.

Community Attributes of Open Source Commons. This label describes the set of variables related to the people or group who are engaged in collaborative development of the software, along with the financial and marketing aspects of the project. In our research, we identified the following as potentially influential for open source success: (1) user involvement;

(2) leadership; (3) social capital; (4) group homogeneity/heterogeneity; (5) group size; (6) project financing; and (7) marketing strategies.

User involvement is one of the long-standing variables known to influence the success or failure of traditional software development projects (Ewusi-Mensah, 2003). As we described above in the user-centric innovation section, it is also thought to be critical in open source settings (von Hippel and von Krough, 2003; von Hippel, 2005a; 2005b). Similarly, the challenging concept of *leadership* appears again and again in the literature as a factor that influences the success or failure of teams. It is known to be a factor in the performance of traditional face-to-face teams, as well as in the context of virtual teams (Tyran et al.,2003). Moreover, it is repeatedly mentioned as a key factor in open source studies (Weber, 2004). Components of leadership include how well the leader(s) are able to motivate others on the team (Healy and Schussman, 2003), as well as how well goals are clarified and articulated Katzenbach, J. and Smith, D. 1993). In the fields of political science and economics, the degree of *social capital* – usually characterized as "trust" between community members – is often discussed when describing a "healthy" or vibrant community (Putnam 2007; Costa and Kahn 2004). In other commons settings three factors contribute to the establishment and maintenance of trust in groups: reciprocal relationships (e.g., I help you, you help me), repeated interactions (Ostrom, *et al*. 1999), and regular face-to-face meetings (Maznevski and Chudoba, 2000; Nardi and Wittaker, 2002).

For a long time, *group heterogeneity* is thought to influence the ability for a team to act collectively (Sandler, 2004). However, this is a very general concept can be conceptualized and measured along several dimensions (Agrawal, 2002; Velded, 2000). Varughese and Ostrom (1998) sub-divide the concept into three categories: (1) socio-cultural heterogeneity; (2) interest heterogeneity; and (3) asset heterogeneity. Socio-cultural heterogeneity includes

attributes such as ethnicity, religion, gender, caste (Agrawal and Gibson, 1999), language, or other cultural distinctions. The general presumption is that groups with diverse socio-cultural backgrounds will have more difficulties working together because of a lack of understanding and, potentially, because of a lack of trust. Interest heterogeneity captures the motivations of people for wanting to participate in a commons. Volunteers, for example, participate in an open source project for different reasons than some paid programmers. It is an open question as to whether diverse or diverging interests in open source affect collaboration, although there is some literature that suggests some tensions when volunteer and business interests coincide. Lastly, asset heterogeneity captures the idea that some individuals may bring to a project capabilities or resources that others on the team might not have themselves. For example, concepts like wealth and power (in terms of political power) are two types of assets found in some group settings. Some studies related to natural resource commons have found that heterogeneity in assets negatively impacts a group's ability to self-organize (Blomquist, 1992; Issac and Walker, 1988).

Group size is another challenging variable that has a long history of debate over its influence in successful or failed commons and software development settings (see Schweik et al., 2008) for a summary. The general thought is that the larger the group the more challenging the coordination costs (Olson, 1965; Brooks, 1975). Yet others have found conflicting empirical results, and specifically in open source, the famous "Linus' Law" – "with more eyes, all bugs are shallow" (Raymond, 2001) – suggests that larger groups are actually helpful (this aligns with the crowdsourcing idea, earlier). Moreover, the relationship between group size and success is complex, not direct, and probably not linear. For instance, Olson and Olson (1997; cited in Deek and McHugh, 2008: 197) note that changes in group size tend

to simultaneously affect other variables, such as group homogeneity and leadership. In short, group size has long been thought to be influential, but its relationship is unclear.

The last two community attribute variables are *project financing* and *marketing strategies*. Several authors discussing open source emphasize financing as a key variable for project success (Weinstock and Hissam, 2005; Fogel, 2007). The argument essentially is that financing can ensure that someone is working on the project and provide some assurance that the project will move ahead. At the same time, funding from a particular source could lead to some tensions over future technical direction of the project in the case where there is a hybrid (e.g., volunteer and paid developer) team. Turning to marketing, surprisingly, there appears to be very little in the literature on this as a variable that affects open source success or abandonment. Yet there are indirect suggestions in the literature about the importance of getting the project known in the early days to gain a user community (e.g., market share) as well as more development support. For this reason, we include it in our list of potentially important variables (Figure 2).

Institutional Attributes of Open Source Commons. This category captures variables related to the governance and management systems used by the open source commons and the types of rules in place intended to guide the behavior of participants. We refer to this bundle of variables as the institutional design of the project. Institutions are known to be a key set of variables in natural resource commons settings. In this category, we build specifically on the work of Elinor Ostrom (1990, 2005) and her colleagues (Kiser and Ostrom, 1982, Ostrom, Gardner and Walker, 1994) who organize institutions into three levels: Operational, Collective Choice and Constitutional. Operational norms and rules oversee the day-to-day activities in a project. Collective choice rules define how changes to operational level rules occur and who has the authority to make such changes. Constitutional level rules specify who

is eligible to change Collective Choice rules and also define the procedures for making such changes. They also can be formalized rules that establish the boundaries or principles that the collaboration is grounded upon. The project's open source license is an obvious example of this type of constitutional level element. It is only very recently that researchers are beginning to conceptualize and investigate empirically institutional designs in open source settings (e.g., Schweik and Semenov, 2003; O'Mahony and Ferrarro, 2007; Marcus, 2007; and Schweik and English, 2007). But especially given the complexities emerging in the open source ecosystem (Figure 1) it is highly likely that institutional designs will be a factor in whether some projects succeed, and some projects become abandoned.

An Empirical Analysis of SourceForge.net Projects

We will now give an extremely condensed summary of empirical work we are just completing related to the variables denoted with an asterisk (*) in Figure 2. For those who are not familiar, Sourceforge.net (SF) is the largest open source software project hosting site "out there." It is a free (as in cost) platform that provides a place where programmers can create and manage an open source project, as well as providing a version control system for the storage and management of the code they are developing. We mentioned earlier that currently SF hosts over 130,000 projects.

Thanks in part to a project out of Syracuse Unversity (FLOSSMole, 2008), along with data "crawling" work we did on our own in the fall of 2006, we compiled a dataset containing of 107,747 SF projects (English and Schweik, 2007). Using this database, we first organized projects based on two longitudinal stages – "Initiation" and "Growth." Projects in the Initiation Stage have not yet produced a first release of code. Growth Stage projects have. Next, within these two longitudinal groups, we classified these projects as either successful collaborations

(meaning they were and continue to be actively worked on), abandoned or indeterminate collaborations. We then undertook a significant manual validation process to verify that the classification system indeed was accurate. We are *greatly* summarizing the work that was done here – interested readers should read English and Schweik (2007).

Sourceforge.net variables. With a measure of success and abandonment in hand, we turned to a process of matching SF data to the theoretical concepts shown in Figure 2. It is likely that SF will be around for some time. With groups like FLOSSMole regularly collecting temporal snapshots of the SF repository, we think it is useful to investigate whether SF data alone does well in explaining success or abandonment.

The data we utilize from SF for each project consists of five numerical variables and seven "groups" of categorical variables. The five numerical variables include: "Developers," "Tracker Reports," "Page Visits," "Forum Posts" and "Ranking Index." The seven "groups" of categorical variables include: "Intended Audience," "Operating System," "Programming Language," "User Interface," "Database Environment," "Project Topic," and "Project License." Short descriptions of each of these variables and the theoretical concept they are related to (in Figure 2) are provided in Table 2. One point that becomes immediately apparent is that SF data provides measures of some (but not all) physical and community attributes thought to be influential in open source projects, but is extremely lean in terms of data related to institutional attributes. The only institutional characteristic it captures is the project's open source license used.

Statistical Methods: Classification Tree Analysis. With a robust dependent variable in hand (success or abandonment), and the SF dataset providing some measures of factors that might be influential in leading to success or abandonment, we turned to the Classification

Tree approach for data analysis. We have built a number of different trees based on different samples of the data, but in this paper, we'll provide only one tree for discussion.

In general, classification techniques include cluster analysis, discriminate analysis, logistic regression, and classification and regression trees. The purpose of these approaches is to efficiently divide the sample data into groups based on one or more independent variables. For example, logistic regression, a commonly used technique, accomplishes classification by determining linear combinations of the independent variables that correlate with (or predict) dependent variable groupings (Hosmer and Lemeshow, 2000). Classification trees are a unique, nonparametric approach that has several advantages, including accommodation of both categorical and numerical variables, and the ability to model complex interactions (Breiman, 1984). We used classification trees (De'ath and Fabricius, 2000) to test the ability of the SF open source independent variable data to discriminate between projects that were successful and those that were abandoned after they generated a first release of their code – Growth Stage projects.

We initially set out to run a classification tree analysis on the entire dataset (n=107,747). Unfortunately, the computational requirements were too high. To circumvent this problem, we took multiple random subsets to develop trees for projects in the Growth Stage only. Our goal was to determine a representative sample size that would produce useful results, while still keeping below the computational threshold. It appeared that at n = 1000 or greater, the sample apparently included enough variability and enough replicates to produce instructive fairly accurate results in most cases. In the tree results we are about to discuss, we used a random sample of 1000 SF growth stage cases, with categorical variables being assigned a value of 0, 1, or a 2. A value of 0 indicates that the project administrator did not select that independent variable (for example, they do not use the java programming

language). A value of 1 indicates that the project administrator did choose that independent variable (e.g., they do use the java language), and a value of 2 indicates that the project administrator did not choose any subcategory of that independent variable group (e.g., they did not answer the Programming Language group entries at all).

Example of Classification Tree Results. We only have space to present one of the classification trees we generated using the SF dataset (Figure 3). As indicated by the "cc" percentages, greater than 80% of the projects in the first left and right nodes were correctly classified by dividing the projects by whether they had greater than or less than 6,352 page visits. Downloads and Forum Posts further separated successful projects in the right leaves. Moving down the tree on the right side, higher levels of Page Views and use of XWindows (one of the "User Interface" categories) were discriminators of success. Developers and number of downloads contributed to partitioning nodes that contained relatively few observations, and were partitioned with moderate success (cc=0.63 to 0.71). This model correctly classified 80% of the projects, with Kappa statistic = 0.524.

These statistics show that variables that one might expect to be associated with successful projects are indeed associated with success. Page Visits and Downloads are associated with the interest of users in the software and are a measure of *product utility* (Figure 2). Forum posts are one component of *collaborative infrastructure* (Figure 2) and indicate an active community where users and developers are communicating about the project. It also suggests a project trying to utilize technology to reduce task granularity by building a question and answer repository that is searchable. Finally, with the exception of the XWindows subcategory of the *User Interface* group of variables, categorical variables are conspicuously missing from the tree.

Discussion. In sum, our classification tree results suggest that greater software utility (reflected in higher numbers of downloads and page visits) and use of communication and infrastructure (forums, bug tracking system) discriminate between successful and abandoned open source projects in the majority of cases. We intentionally formulated our definition of success to include useful projects having a small number of users, but despite this definition, having a larger number of users discriminates between success and abandonment in the majority of cases. Also, successful collaborations tend to use the forums and bug tracking features of SF more than the abandoned ones.

We were surprised that our categorical variables (e.g., intended audience, operating system, programming language, database environment, project topics) did not stand out in this and other classification trees not presented. We interpret this to mean that open source has become a larger, more mainstream phenomenon. In our view, the "user-centric" and volunteer emphasis in past open source literature reflected, at least in part, programmers building software that they needed to support the continued buildup of open source technologies (e.g., the Linux operating system and related software, web and email processing, etc.). The fact that none of the categories related to these concepts stand out as important discriminators in our data suggests that people are collaborating in all kinds of open source projects, not just ones centered on these more traditional open source development efforts. Lastly, and perhaps not surprisingly, this analysis emphasizes the importance of community attributes over physical attributes in explaining success or abandonment of open source commons. Moreover, the role of institutional attributes remains to be seen given that the SF data contains very little related to this set of potentially explanatory factors.

Conclusions

The primary goal of this paper was to make the point that what is occuring related to open source software collaboration has, potentially, important implications for public sector collaboration in the future. To make this argument, we provided an overview of key concepts– peer production, open source and open concent, user-centric innovation, crowdsourcing, and task granularity. With those articulated, we reflected on a failed attempt at implementing a peer production collaboration between local government officials, and presented a vision of what we think we should be striving for. We then turned to a summary of our current research project trying to understand factors that lead to continued collaboration (success) or abandonment in open source software "commons." We explained that open source is *not* just about volunteers, many projects involve participation from the private, public and nonprofit sectors. We then introduced a set of variables that are found in theoretical and/or empirical literature as potential influential factors, and then we described our efforts to investigate these relationships using a huge dataset of open source projects from Sourceforge.net. Our ultimate goal in this project is to identify some "design principles" that can potentially be "ported over" to more general "open content" collaborations, and more specifically, intra- and inter-governmental collaborations in the public sector, or collaborations across sectors.

Our empirical results suggest that some of the "physical attributes" of open source projects (e.g., programming language, type of software, database environment, etc.) are not significant factors in determining collaborative success. Larger numbers of Page Views and Downloads characterize success in the majority of projects, even when projects with small numbers of users are specifically included in the definition of success. We also find that projects who utilize collaborative infrastructure tend to be more successful than ones who don't. Finally, our analysis shows that the Sourceforge repository is missing some key data

related to (mainly) community and institutional attributes. The work we are currently undertaking – both case study and an online survey of open source developers – hopes to fill in these data gaps.

To conclude, we hope we have instilled in readers a recognition that open source collaboration is an important phenomenon that could be a model for how public sector organizations collaborate between themselves, or with other organizations in other sectors, or even with citizens themselves. In some of the examples of peer production, we've already started to see some initial explorations along these lines (e.g., NASA clickworkers). A significant question ahead will be whether open source-like collaborations will be explored and embraced in public sector settings, as other sectors are doing, or whether organizational and bureaucratic structures and a lack of incentives encouraging this kind of collaborative innovation will hold them back.

Table 1. **Examples of Task Granularity in Peer Production Applications**	
Extremely Coarse Grained	- Lead author in a Wikibook - Core developer in an open source software project
Coarse Grained	- Writing a chapter in a Wikibook - Leading a team of open content collaborators
Medium Grained	- Writing the first draft of a Wikipedia entry - Contributing a relatively small programming fix in an open source project - Making and posting a video to YouTube.com
Fine Grained	- Sign up to receive information of interest - Subscribing to an email list or RSS feed - Answering a question to someone else via an email distribution list or forum - Reporting a bug in some software - Posting an entry (e.g., a Digg story) - Submitting a story in SlashDot - Adding a sentence or reference to a Wikipedia page - Voting that you liked a posting in Digg - Save a URL via del.icio.us
Transparently Grained	- Google – Pagerank algorithm - Amazon.com – recommendations on what others have read - ReCapcha – typing in scanned words such that it contributes to digital books - Digg's recommendation system

Table 2 – *Selected* Variables in Sourceforge.net Metadata

SF Variable	Description	Theoretical concept it is thought to capture (Figure 2)
Developers	Total number of developers on the project	**Group size** – Community attribute
Tracker Reports	Total number of bug reports, feature requests, patches and support requests	**Collaborative infrastructure** – bug tracking system. Physical attribute.
Page Visits*	Total number of views of any of the project's SF website	**Product utility** – Physical attribute
Forum posts	Total number of Forum posts made to the project's public forums from 2005-10-06 through 2006-08-02	**Collaborative infrastructure** – Physical attribute.
Downloads*	Total number of downloads of the software package	**Product utility** – Physical attribute
Intended Audience	Categorical variable describing the type of person project targets (e.g., end users, advanced end users, business, computer professionals, other)	**User Involvement** (User centric Innovation) – Community Attribute
Operating System	Categorical variable describing the operating system(s) the software will run on.	**Product utility**, critical infrastructure – Physical attribute
Programming language	Categorical variable(s) describing the programming languages used.	**Product utility**, preferred technologies – Physical Attribute
User Interface	Categorical variable describing how the software interfaces with the user (e.g., command line, GUI, etc.)	**Product utility**, preferred technologies – Physical Attribute
Database Environment	Categorical variable for the database used in the project's software (if relevant)	**Product utility**, preferred technologies – Physical Attribute
Project Topic	Group of 19 categorical variables consists of the topics that the SF website uses to classify the projects (e.g., education, games, security, printing, etc.)	**Product utility**, critical infrastructure – Physical attribute
Project License	Categorical variable(s) describing the type of open source license(s) used.	**Constitutional rules** – Institutional Attribute

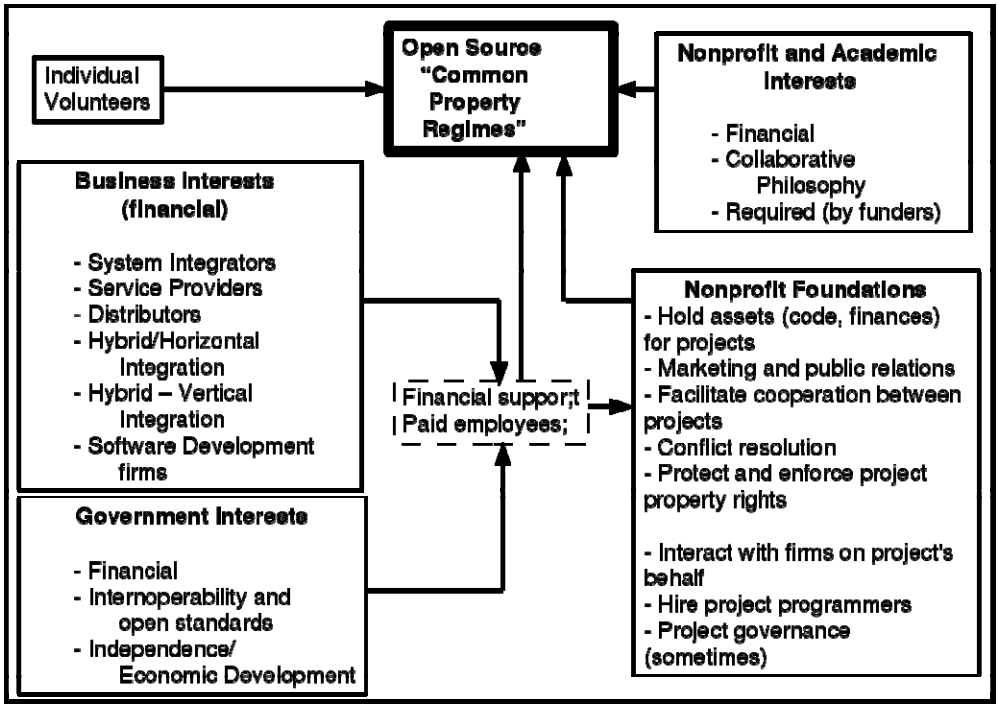

Figure 1.
A Broad-Scale View of the Open Source Ecosystem

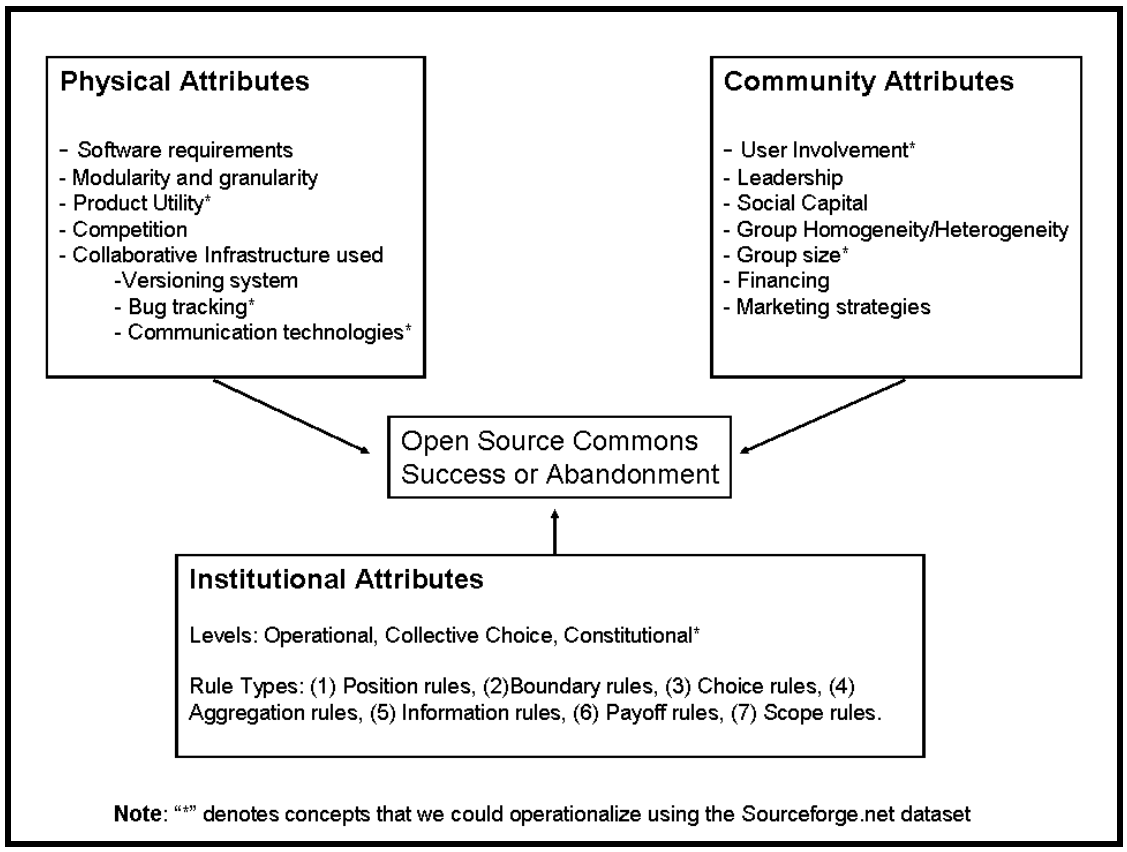

Figure 2.
Factors Thought to Influence the Success or Abandonment of Open Source Collaborations

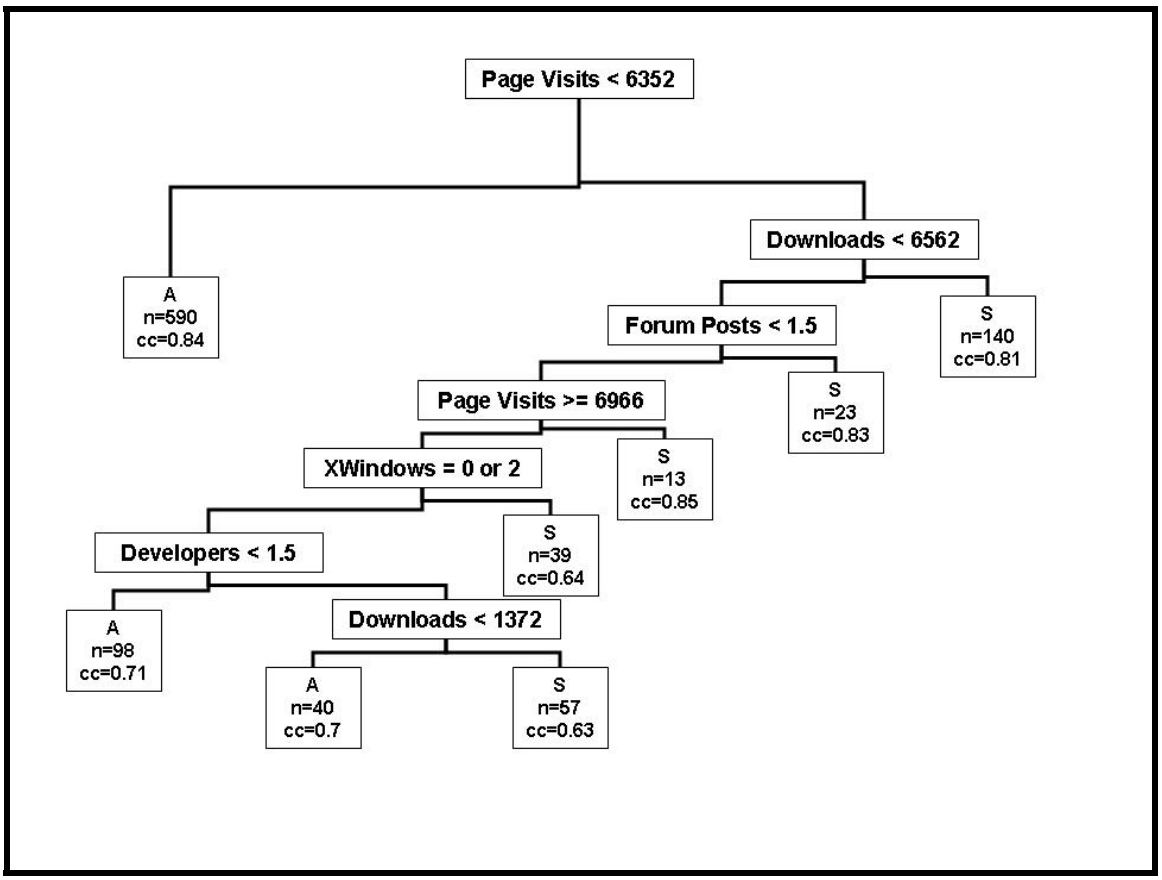

Figure 3.
Example of Classification Tree Results Using 1000 Randomly Sampled SF Growth Stage Projects

Acknowledgements

Support for this study was provided by a grant from the U.S. National Science Foundation (NSFIIS 0447623). However, the findings, recommendations, and opinions expressed are those of the authors and do not necessarily reflect the views of the funding agency. Thanks go to Megan Conklin, Kevin Crowston and the FLOSSmole project team (http://ossmole .sourceforge.net/) for making their Sourceforge data available, and for their assistance with their data.

References

Agranoff, R. 2008. “Conductive Public Organizations in Networks: Collaborative Management and Civic Engagement.” In Kaifeng Yang and Erik Bergrud (eds.) *Civic Engagement in a Network Society*. Information Age Publishing: Charlotte, North Carolina.

Agrawal, A. 2002. “Common Resources and Institutional Stability.” In Elinor Ostrom, Thomas Dietz, Nives Dolsak, Paul C. Stern, Susan Stronich, and Elke U. Weber (editors) *The Drama of the Commons*. National Academy Press: Washington, D.C. pp. 41-85.

Aigrain, Philippe. 2005. “Libre Software Policies at the European Level.” In Joseph Feller, Brian Fitzgerald, Scott A. Hissam, and Karim R. Lakhani (editors) *Perspectives on Free and Open Source Software*. MIT Press: London, England. pp. 447-459.

Bardach, E. 1998. *Getting Agencies to Work Together: The Practice and Theory of Managerial Craftmanship*. Washington, D.C.: Brookings Institution Press.

Benkler, Yochai. 2005. “Coase's Penguin, or Linux and the Nature of the Firm.” In Risab Aiyer Ghosh (editor) Code: Collaborative Ownership and the Digital Economy. MIT Press: Cambridge, MA.

Benkler, Yochai, 2006. *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. Yale University Press: New Haven, CT.

Brabham, D. 2008. “Crowdsourcing as a Model for Problem Solving: An Introduction and Cases.” *Convergence*. 14(1): 75-90.

Brabham, D. 2008. “Moving the Crowd at iStockphoto: The Composition of the crowd and Motivations for Participation in a Crowdsourcing Application.” *FirstMonday*. 13(6). http://www.uic.edu/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/2159/1969. Accessed July 18, 2008.

Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.G. 1984. *Classification and Regression Trees*. Belmont, California: Wadsworth International Group.

Brooks, F. P. Jr. [1975] 1995. *The Mythical Man-Month: Essays on Software Engineering*. Anniversary Edition. Reading, MA: Addison-Wesley.

CiviCRM, 2008. “About CiviCRM.” Available at http://civicrm.org/aboutcivicrm. Accessed March 11, 2008.

Costa, Dora L. and Matthew E. Kahn. 2003. “Engagement and Community Heterogeneity: An Economist's Perspective.” *Perspectives on Politics*, 1(1): 103-111.

De'ath, Glenn and Fabricius, Katharina E. 2000. “Classification and Regression Trees: A Powerful Yet Simple Technique for Ecological Data Analysis.” *Ecology*. 81(11): 3178-3192.

Deek, Fadi, P. and McHugh, James A.M. 2008. *Open Source Technology and Policy*. Cambridge University Press: New York, NY.

DP. 2008. Digital Proofreaders. http://www.pgdp.net/c/. Accessed July 17, 2008.

English, R. and Schweik, C.M. 2007. “Identifying Success and Abandonment of Free/Libre and Open Source (FLOSS) Commons: A Preliminary Classification of Sourceforge.net projects.”*Upgrade: The European Journal for the Informatics Professional*. Vol. VIII, Issue no. 6 (December).

Ewusi-Mensah, Kweku. 2003. *Software Development Failures*. MIT Press: Cambridge, MA.

FLOSSMole, 2008. http://sourceforge.net/projects/ossmole/. Accessed July 22, 2008.

Fogel, Karl. 2007. *Producing Open Source Software: How to Run a Successful Free Software Project*. O'Reilly Media: Sebastopol, CA. Available at http://producingoss.com/.

Friess, S. 2007. “Online Fossett Searchers Ask, Was It Worth It?” Wired. November, 6. http://www.wired.com/techbiz/it/news/2007/11/fossett_search. Accessed July 17, 2008.

Ghosh, R. A. (2005). Understanding Free Software Developers: Findings from the FLOSS study. In J. Feller, B. Fitzgerald, S. Hissam & K. R. Lakhani (Eds.), *Perspectives on Free and Open Source Software*. Cambridge, MA: MIT Press. pp. 23-46.

Google, 2008. Technology Overview. http://www.google.com/corporate/tech.html. Accessed July 11, 2008.

Haas, P. 1992. Peter M. Haas, "Introduction: Epistemic Communities and International Policy Coordination" *International Organization*, 46:1: 1-35.

Hamel, Michael and Schweik, Charles M. under review. “Open-Source Collaboration: Two Cases in the U.S. Public Sector.” *The Information Society Journal*.

Hess, Charlotte and Ostrom, Elinor (editors). 2007. *Understanding Knowledge as a Commons: From Theory to Practice*. Cambridge, Mass: MIT Press.

Hosmer, David W. and Lemeshow, Stanley. 2000. *Applied Logistic Regression*. New York, NY: John Wiley and Sons.

Howe, J. 2006. “The Rise of Crowdsourcing.” *Wired*. June. Available at http://www.wired.com/wired/archive/14.06/crowds_pr.html. Accessed July 17, 2008.

Katzenbach, J. and Smith, D. 1993. “The Discipline of Teams.” *Harvard Business Review*. 71: 111-120.

Kiser, L.L. and Ostrom, E. 1982. “The Three Worlds of Action: A Meta-theoretical Synthesis of Institutional Approaches.” In E. Ostrom (ed.) *Strategies of Political Inquiry*. Beverly Hills, CA: Sage. Pp. 179-222.

Koontz, T.M., Steelman, T. A., Carmin, J., Korfmacher, K. S., Moseley, C., and Thomas, C. W. 2004. *Collaborative Environmental Management: What Roles for Government?* Washington D.C.: Resources for the Future Press.

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

Krishamurthy, S. 2005. “An Analysis of Open Source Business Models.” In J. Feller, B. Fitzgerald, S. Hissam & K. R. Lakhani (Eds.), *Perspectives on Free and Open Source Software*. Cambridge, MA: MIT Press. pp. 279-296.

Lakhani, K. R., & Wolf, R. G. 2005. “Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. In J. Feller, B. Fitzgerald, S. Hissam & K. R. Lakhani (Eds.), *Perspectives on free and open source software*. Cambridge, MA: MIT Press. pp. 3-22.

Lewis, James. 2007. “Government Open Source Policies (Version 4).” Available at http://www.csis.org/media/csis/pubs/070820_open_source_policies.pdf. Accessed March 6, 2008.

Lukensmeyer, C. J. and Torres, L.H. 2008. Citizensourcing: Citizen Participation in a Networked Nation. In Kaifeng Yang and Erik Bergrud (eds.) *Civic Engagement in a Network Society*. Information Age Publishing: Charlotte, North Carolina. pp. 207-233..

Maxwell, Elliot. 2006. “Open Standards, Open Source, and Open Innovation: Harnessing the Benefits of Openness.” *Innovations*. 1(3): 119-176.

Maznevski, M. L. and Chudoba, K. M. 2000. Bridging Space Over Time: Global Virtual Team Dynamics and Effectiveness. Organization Science, 11(5), 473-492.

Muffatto, Moreno. 2006. *Open Source: A Multidisciplinary Approach*. Imperial College Press: London.

Nardi, B. A. and Wittaker, S. 2002. The Place of Face to Face Communication in Distributed Work. In P. Hinds and S. Kiesler (Eds.), *Distributed Work*. Cambridge, Ma: MIT Press.

Nasa, 2001. Clickworkers (original site). http://clickworkers.arc.nasa.gov/top. Accessed July 17, 2008.

Nasa, 2008. Clickworkers HiRise. http://clickworkers.arc.nasa.gov/hirise. Accessed July 17, 2008.

NOSI, 2008. Nonprofit Use of FOSS Survey, 2008. Available at http://www.nosi.net/system/files/NOSISurveyReport08.pdf. Accessed March 11, 2008.

O'Leary, R., Gerard, C. and Bingham, L.B. 2006. Introduction to the Symposium on Collaborative Public Management. *Public Administration Review*. 66(s1): 6-9.

Olson, Mancur. 1965. *The Logic of Collective Action*. Cambridge, Mass.: Harvard University Press.

Olson, G. and Olson, J. 1997. “Making Sense of the Findings: Common Vocabulary Leads to the Synthesis Necessary for Theory Building.” In K. Finn, A. Sellen and S. Wilbur (eds.) *Video-Mediated Communication*. Lawrence Erlbaum Associates: Mahwah, NJ.

Ostrom, Elinor. 2005. *Understanding Institutional Diversity*. Princeton, N.J.: Princeton University Press.

Ostrom, E. Gardner, R. and Walker, J. 1994. *Rules, Games, and Common-Pool Resources*, Ann Arbor: University of Michigan Press.

Ostrom, E., Burger, J., Field, C.B., Norgaard, R.B., and Policansky, D. 1999. “Revisiting the Commons: Local Lessons, Global Challenges.” *Science*. 284: 278-282.

Peizer, Jonathan. 2003. Realizing the Potential of Open Source in the Nonprofit Sector. Open Society Institute. Available at http://www.soros.org/initiatives/information/articles_publications/articles/realizing_20030903. Accessed March 10, 2008.

Putnam, Robert D. 2007. “E Pluribus Unum: Diversity and Community in the Twenty-first Century. The 2006 Johan Skytte Prize Lecture.” *Scandinavian Political Studies*, 30(2): 137-174.

Raymond, Eric. 2001. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol, CA: O'Reilly.

ReCaptcha, 2008. What is reCAPTCHA? http://recaptcha.net/learnmore.html. Accessed July 17, 2008.

Sakai, 2008. “The Sakai Partners Program.” Available at http://www.irrodl.org/index.php/irrodl/article/view/496/950. Accessed March 16, 2008.

Sandler, Todd. 2004. *Global Collective Action*. Cambridge: Cambridge University Press.

Schweik, C.M. and English, R. 2007. “Conceptualizing the Institutional Designs of Free/Libre and Open Source Software Projects.” *First Monday* 12(2). Available at http://www.firstmonday.org/issues/issue12_2/schweik/index.html.

Schweik,C., English R., Kitsing, M. and Haire, S. 2008. “Brooks’ versus Linus’ Law: An Empirical Test of Open Source Projects” in Soon Ae Chun, Marijn Janssen and Ramon GilGarcia (eds) *The Proceedings of 9th International Digital Government Research Conference*, Montreal, Canada, May 1821, pp. 423424.

Schweik, C., Evans, T and Grove, J.M. 2005. Open Source and Open Content: a Framework for Global Collaboration in Social-Ecological Research. *Ecology and Society* 10 (1): 33. [online] URL: http://www.ecologyandsociety.org/vol10/iss1/art33/. 25 pp.

Schweik, C.M. and Semenov, A., 2003. The Institutional Design of "Open Source" Programming: Implications for Addressing Complex Public Policy and Management Problems. *First Monday* 8(1). http://www.firstmonday.org/issues/issue8_1/schweik/.

Simon, Kimberly D. 2005. “The Value of Open Standards and Open-Source Software in Government Environments.” *IBM Systems Journal* 44(2): 227-238.

Stebbins, R. A. 2001. “Serious Leisure.” *Society*. 38(4): 53-57.

Surowiecki, J. 2004? The Wisdom of Crowds. Publisher?

Tyran, Kristi. L., Tyran, Craig. K., & Shepherd, Morgan. 2003. “Exploring Emerging Leadership in Virtual Teams.” In C. B. Gibson, & S. G. Cohen (Eds.), *Virtual Teams that Work: Creating Conditions for Virtual Team Effectiveness* (1st ed.) (pp. 436). San Francisco: Jossey-Bass.

U.S. NSF Office of Cyberinfrastructure. 2007. “Software Development for Cyberinfrastructure.” Available at http://www.nsf.gov/pubs/2007/nsf07503/nsf07503.htm. Accessed May 14, 2008.

van Laerhoven, F. and Ostrom, E. 2007. “Traditions and Trends in the Study of the Commons.” *International Journal of the Commons*. 1(1): 3-28.

Varughese, G. and Ostrom, E. 2001. “The Contested Role of Heterogeneity in Collective Action: Some Evidence from Community Forestry in Nepal. *World Development*. 29(5): 747-765.

Velded, T. 2000. “Village Politics: Heterogeneity, Leadership and Collective Action.” *Journal of Development Studies*. 36(5): 105-134.

von Hippel, Eric and von Krogh, Georg. 2003. “Open Source Software and the 'Private-Collective' Innovation Model: Issues for Organization Science.” *Organization Science*. 14(2). March-April. pp. 209-223.

von Hippel, Eric. 2005a. *Democratizing Innovation*. Cambridge, MA: MIT Press. Available at http://web.mit.edu/evhippel/www/democ1.htm.

von Hippel, Eric. 2005b. “Open Source Software Projects as User Innovation Networks.” In Joseph Feller, Brian Fitzgerald, Scott A. Hissam, and Karim R. Lakhani (editors)

Perspectives on Free and Open Source Software. MIT Press: London, England. pp. 267-278.

Wayner, Peter. 1999. “Germany Awards Grant for Encryption.” *New York Times*, November 19.

Weber, S. 2004. *The Success of Open Source*. Cambridge, MA: Harvard University Press.

Weinstock, C. B., & Hissam, S. A. 2005. “Making Lightning Strike Twice.” In J. Feller, B. Fitzgerald, S. A. Hissam and K. R. Lakhani (Eds.), *Perspectives on Free and Open Source Software*. Cambridge, Ma: The MIT Press. pp. 143-159.