Toward Research Transparency: Fostering Collaboration, Increasing Accountability, and Building a Knowledgebase Using Open Source Methods

As a discipline involved in the discovery, creation, and sharing of knowledge matures, more robust and often more complex methods are required to effectively practice that discipline. New variables need to be addressed, additional stakeholders need to be considered, and highly refined questions demand equally sophisticated answers. This is evident to institutional researchers as they work to provide nuanced, pragmatic information to academic and administrative units with significantly varied needs and expectations (Volkwein, 2008).

The Challenges of Institutional Research

In providing strong support to those addressing pressing issues for their universities, institutional researchers can face many organizational challenges, challenges they share with those in other knowledge fields. This paper explores three of these challenges in an effort to understand how they may be managed more efficiently.

The Challenge of Collaboration

The first challenge institutional researchers can face is collaborating with others effectively. Collaboration exists in matters of degree rather than as a binary function, but it is sometimes the case that institutional research (IR) projects require or benefit from specialized knowledge or diverse perspectives. Examples included inviting stakeholders to examine near final drafts of reports, working closely with those in other units to prepare reports, and integrating data from multiple sources to answer difficult questions. Though not usually thought of as such, a personnel change on a project also represents collaboration, though indirect, between the old personnel and the new. As such, it can be said that collaboration occurs whenever a project requires the involvement of two or more persons who may be separated by space or time.

Besides often being necessary, when properly implemented, collaboration can bring about a synergy that produces outcomes of better quality than a single individual would be capable of accomplishing on his or her own (Frank Fox & Faver, 1984; Landry, Traore, & Godin, 1996). Improperly implemented, it can result in excessive group decision making that wastes time and resources and nullifies the benefits of sharing perspectives, such as reduction in the workload. A large part of the challenge lies in organizing contributions to the project. It can be difficult to determine who should work on what portions of a project and track activity on each project component. If this is done poorly, project members may be exhausted by redundant activity and aggravated by lack of project awareness. All told, it takes significant effort to balance the potential benefits of collaboration with the risk of drowning in the organizational complexity that can result as new contributors become involved in a project.

The Challenge of Interruption

The second challenge institutional researchers sometimes face involves minimizing the impact of project interruptions. Demands on time and other resources can make it difficult to focus on a single project from start to finish. Sometimes a project needs to be delayed while waiting for input from others. Other times a delay is a natural part of the work cycle, as with yearly reporting. If the interruption is long enough, it may take time to reorient oneself with the project parameters before becoming productive again. This reorientation process may involve
reading through old documentation, meeting with others about topics previously discussed, and so on. As such, it can be said that the total cost of a break in a project equals the delay plus the time required to get back to a productive state. Given the limited time of institutional researchers, it makes sense to minimize interruptions. When that is not possible, it also makes sense to research methods to reduce the time it takes to reorient oneself with a project.

The Challenge of Iteration

The third challenge involves managing complex projects that require multiple iterations. This is especially relevant to those projects that involve complex data analysis. Such projects may necessitate managing a group of related files that need to be updated as new information becomes available. There is a danger of losing early file versions that may be useful and miscommunication can occur as a result of one or more persons involved in the project relying on outdated file versions (Wilson, 2006). Even somewhat stable processes like annual IPEDS reporting may change over time. A simple change in reporting agency definitions, the underlying data structure, or an institution’s organizational structure may necessitate one or more changes to the reporting system. In these cases, it behooves the institutional researcher to determine if better systems exist to manage this process.

Any one of these challenges alone, making effective use of collaboration, dealing with project delays, and managing project files can stress the IR practitioner, but often these challenges are encountered in tandem. More complicated projects require input from others and multiple sources of information, both of which contribute to more frequent or longer delays. If the trends in other knowledge fields (Owen-Smith, 2001; Wuchty, Jones, & Uzzi, 2007) and the growth in the sophistication of IR work are any indication (Volkwein, 2008), these challenges may become more pervasive in the future. They need to be addressed at some point because it is unrealistic to never work with others, focus on one project to the exclusion of others, and never introduce changes to established processes.

Benchmarking Open Source Software Development

It is useful to look for other communities, however defined, that experience these same issues to see what steps they take to manage them. The work of those who contribute to the development of open source software (OSS) can be characterized as similar to some IR work, though to a stronger degree. OSS work tends to be highly collaborative. While each project is guided by one or more developers, people are free to move in and out of the project and contribute as they see fit. OSS work also tends to be intermittent, as most contributors have other professional commitments, and iterative given that software typically evolves through a number of versions informed by constant feedback. Given some of the similarities between OSS and IR work and given the fact that many OSS projects have been successful, some very much so, it makes sense to determine how the approaches used there might be useful in IR work (Ambati & Kishore, 2004; Hardaway, 2005). This necessitates examining the typical organizational structure of OSS communities and the tools they use.

The Open Source Development Process

As a general description, OSS projects tend to be organized around a core group of one or more developers. This core group typically provides the management functions for the project including the review of external contributions to a project to determine if they are appropriate (Fogel, 2005). Anyone is free to submit a contribution for review and prolific contributors may be invited to join the development team. Frequent contributors to one or more projects are often held in high esteem by other contributors in these “gift economies” (Ljungberg, 2000;
Stromberg, 2004). The software is made available in various stages of development for free download.

Recent advances in information technology have enabled OSS developers and contributors to better coordinate and track these contributions. This is necessary given that some of these projects have hundreds of contributors. While not all projects make use of the same tools, the following tools are arguably the most popular.

**Open Source Tools**

Most OSS projects use a Web site to provide the overall mission and a description of the project, including any relevant documentation. This documentation tends to include a description of the features and functions of the software, instructions for compiling and/or installing the software, and details about how to submit contributions for review. OSS projects also utilize one or more electronic mailing lists to organize discussions and disseminate information among contributors. The mailing list is often where decisions are made about project activities.

Project activities themselves are monitored in a separate tool. An issue tracker, which functions somewhat like a to-do list, is used to track specific project actions. Each ticket in the system, which represents an issue or action, maintains a complete history of steps taken to fulfill that ticket. A version control system is used to manage the files associated with the project. A version control system stores a complete history of each file and all of its versions, along with a note accompanying each version that describes who changed the file, when the file was changed, and why the file was changed.

Often these tools work in tandem with each other. Changes in the version control system may be cross-referenced with tickets in the issue tracker. The tickets can in-turn may be cross referenced with the archived discussion on the mailing list. Frequently changes in the issue tracker or the version control system will automatically generate a message to the mailing list informing the community of the change. Many projects utilize collaborative development environments that integrate several or all of these tools into a single application designed to simplify the cross-referencing among the tools.

**Application to Other Fields**

How do these methods and tools contribute to project success and can they be used for work other than software development? Emerging research on these questions suggests that these methods and tools contribute to project success by facilitating good management efforts and that it makes sense to determine how to develop and use them to better support smaller projects in addition to larger collaborations (Ambati & Kishore, 2004; Jirotka, Procter, Rodden, & Bowker, 2006; Ljungberg, 2000; Robbins, 2002). Inadequate documentation and the frequent inflow and outflow of project members create larger problems as projects scale in complexity and the use of OSS tools in other environments may help alleviate some of this. To better understand why the researchers make these claims, it is useful to examine these methods and tools from a project management standpoint and speculate how IR work environments may be receptive to their use.

Perhaps most importantly, the use of these tools and methods fosters a high level of project awareness, a key factor in the completion of successful projects (Gutwin, Penner, & Schneider, 2004). All the tools work together to make it clear who is working on what, what has been accomplished, and what still needs to be done. This gives people the freedom to work a
more decentralized environment, which can be very beneficial in an environment with such clear expectations (Malone, 2004). People working in these systems do not require as much supervisor direction. They can simply log into the systems, see what is available to work on, and start working. There is less need to e-mail or contact other project members to see if a particular action has been completed or find out what actions have recently been proposed. All of that information is available in the environment provided by these tools. This makes increased collaboration easier to manage, especially for groups that are geographically dispersed, as all communication among project members is stored in a central location. It also can result in fewer meetings, a costly expenditure of time for many, since project members are effectively in constant, though measured, contact.

In IR work, collaboration tends to be smaller in scope. Institutional researchers work with others in their unit or rely on other units for more modest contributions, such as access to data or feedback on nearly finished products. As mentioned earlier, this might change as IR practitioners start to tackle some of the more complicated issues in higher education, like providing support for those working on the assessment of student learning outcomes for example. But in any case, OSS tools can be used to make collaboration more inviting by providing detailed information on what a project is about and clear procedures for getting involved. In a more mundane sense, the tools can be used to automate certain project functions. For example, instead of exchanging several e-mails with several contacts about data needed for an annual report, those contacts can simply upload the data to the project site and the site will send notification to the project leader when all the files are available.

**Building the Knowledgebase**

In OSS work, log files, electronic mailing list messages, wiki pages, and ticket comments, created as a function of the way in which the project members coordinate, become archaeological records of a project’s evolution. Information that is usually tacit in a project becomes accessible to other project members now and in the future. The tacit information contained in these artifacts of the development process combines with the available explicit documentation to provide a knowledgebase. This knowledgebase enables project leaders to make decisions that improve chances of project success, fosters the development of expertise among the project personnel, and makes it much easier for new contributors to acclimate to the project (Bendoly & Swink, 2007).

The knowledgebase also contributes to the long-term success of a project by insulating it from the risks associated with key members, and their knowledge, from leaving the group. This is a common experience to those in any line of knowledge work, including IR, when inheriting the project of a previous employee or current employee who has moved on to something else. The latter situation is not as challenging since the co-worker is still available for consultation, but the former situation can prove very difficult (and it would be useful to not have to consult the co-worker frequently).

A well-developed knowledgebase provides the person who inherited the project not only the information on where the project is going, but also information on how the project has developed up to that point. This context makes it easier to identify ways to improve the project as it can be determined what alternate approaches have already been tested and to understand the context in which prior decisions were made. Consequently, project personnel may devote more time to value-added activities rather than revisiting the mistakes of the past.
**Enhancing Accountability**

The capability of these tools to generate a constantly evolving knowledgebase and track user contributions provides a strong measure of accountability, a necessary component of strong, scientific work. The tools make it clear who made which contributions, when they were made, and what had to be accomplished in making them. While accountability may not be quite the same issue in OSS communities, it is most certainly a concern in academic research and scholarly IR research.

Good scientific practice requires that researchers be able to make enough information available about their projects so others may replicate and extend their work. The extent to which different research disciplines currently make their methods and data available for these practices varies significantly (Schneider, 2004). Researchers in the physical and natural sciences seem to do this better than those in other disciplines. Specifically, it is not uncommon in education for researchers to be unable to provide enough project detail to replicate findings (Burkhardt & Schoenfeld, 2003; Mick, 2000; Wicherts, Borsboom, Kats, & Molenaar, 2006). It is not likely that these omissions are malicious; it is more probable that for most people, preparing materials for others to replicate research findings is time intensive (Wicherts et al., 2006).

By definition, OSS projects must be transparent enough for anyone who wants to make a contribution and offer criticisms and suggestions to do so. Projects typically provide read-only access to each of their tools and even to the final product without requiring any user authentication. The practices and tools used to do this can also be used to make research projects transparent, circumventing the need for researchers to spend time preparing separate packets of material for study replication and lending greater validity to the research process (Schneider, 2004). This level of transparency makes it possible to open up not just the end product of a research project, which can increase impact (Antelman, 2004), but the process as well. This may be useful for understanding how the project came together, like getting a look at how the literature base was defined, which is of obvious importance (Kennedy, 2007), or being able to identify errors that may have slipped past the notice of the project developers (Atman & McDonald, 2007). In any case, the available information makes it significantly easier to investigate the foundations for specific claims.

**Adopting Open Source Methods for Institutional Research**

While it is true that most OSS projects tend to make use of a Web site, electronic mailing list, issue tracker, and version control system, IR workers looking to ease into using the tools described here should feel free to adopt them on an as needed basis. A project may not be large enough to warrant using a Web site, but it may still be wise to use a version control system to manage a project’s electronic files. Likewise, sometimes a wiki is all that is needed to maintain a strong level of awareness among the project members.

Even when working on a project alone, it is important to consider whether or not developing a knowledgebase or providing accountability information might prove useful. As discussed earlier, a project might be handed down to someone with no prior exposure to it and a well-developed knowledgebase can help that person make the transition. If the project is one that is worked on intermittently, it is useful to be able to quickly get back into the project after the delay, a process that can be bolstered by the documentation and tracking information that these tools can generate and organize. In all cases, it is necessary to think critically about how to make best use of the tools given the goals of the project. A few examples may help clarify this point.
Fresearch

According to the project Web site, found at https://weblion.psu.edu/trac/fresearch, Freresearch is "…a line of research into the application of open source software development tools and techniques to academic and institutional research projects.” The project currently consists of two core developers. In addition to the Web site, the project utilizes an electronic mailing list for communications among developers as well as with the larger community. The Fresearch project allows for unauthenticated access to all project tools in the spirit of open source development.

Decisions about Fresearch project activities are discussed among the core group of developers. If possible, the discussion takes place on the project’s electronic mailing list. But, sometimes, the discussion occurs in a face-to-face meeting of the developers. The notes from these meetings are then made available on the Web site. Project activities are monitored using an issue tracker. These activities include any tasks that need to be completed to effect the decisions made and any defects or changes to existing project artifacts. Suggestions for enhancements that have been determined to have merit but require further development are also recorded in the issue tracker. As a research project, Fresearch has identified a number of reference materials such as journal articles, Web sites, and books. These materials are monitored in the issue tracker as well. Fresearch uses the version control system to maintain the source files for any research activities such as a project-wide bibliographic database and working files of any project publications.

“Strategic Indicators”

“Strategic Indicators” is the title of an annual publication produced by an IR-related office of a major research university. (The project cannot be viewed by the public for security purposes.) The publication reports on a number of data elements that are intended to monitor the university’s progress toward achieving its goals. As a project, the publication is the result of a coordinated effort involving many offices from various locations in the organizational structure. The production of this publication has been managed using a collaborative development environment since fall 2005.

The “Strategic Indicators” project is an example of how OSS tools may be adopted for non-OSS work. The source files necessary for producing the “Strategic Indicators” document are maintained in a version control system. All tasks necessary for the production of the publication are recorded in an issue tracker. Individual tickets in the issue tracker typically relate to a single activity. Notes relating to progress on completing the activity are recorded with each ticket as changes or comments. Defects identified in the document and suggestions for future improvements are also recorded in the issue tracker.

But the “Strategic Indicators” does not utilize a mailing list. The development is typically coordinated by one core developer who communicates with the various contributors using traditional methods like e-mail, telephone conversations, and face-to-face meetings. Most of the contributors to the project are not even aware of the existence of the collaborative development environment. Still, the tools prove useful even in the absence of direct collaboration.

Dr. Project

Dr. Project, found at https://www.drproject.org, is a collaborative development environment application, which itself is developed using its own software as a collaborative development environment. This particular application has been developed to facilitate learning
to utilize collaborative development environments among students, particularly in the computer science curriculum.

Dr. Project incorporates each of the common OSS tools into a single, integrated Web-based application. Dr. Project began as a modified version of another open source collaborative development environment application and is actively being developed as an open source project itself. Dr. Project exemplifies how OSS tools may be modified to serve needs outside the realm of software development.

**Challenges in Adopting OSS Tools**

In reviewing the examples presented above and asserting that it is worth considering whether these OSS development methods and tools may be useful in developing and completing IR projects, it is important to realize that these systems offer challenges of their own. With the possible exception of the mailing list, which functions similarly to e-mail, the tools described in this paper demand an investment of time to learn how to use proficiently. The Web sites for these projects, most of which are wiki-enabled, require that the user learn how to enter and format text, attach files, set the administrative parameters, and so on. Issue trackers require similar efforts to learn how to use effectively. Probably the most challenging tool to learn how to use is a version control system, which represents a very large change in the way people conceptualize saving and storing their files. Taken together, these changes are at least as large as the move from inter-office memos to e-mail messaging.

Regardless, it could be argued that learning how to use these tools is not much more challenging than learning how to use an e-mail client, internet browser, or word processing program for the first time. Many of the tools offer strong documentation and large online communities eager to answer questions. (Many of those community members are involved in the development of the tools to some extent, so they tend to be very knowledgeable about the software.) An exception is that some of the tools may require installation on a server, which is probably best handled by an information technology department. Of course, many of the learning difficulties may be reduced or avoided by using commercial versions of these tools and the customer support that comes along with them, as mentioned earlier.

A possible misconception that may prove challenging in adopting these tools is that they reduce the need for planning. This is definitely not the case. Projects well-supported with technology can fail if the necessary organization is not in place (Lawrence, 2006). If anything, a decision to use any of the tools, which requires that the users consider what role it is going to play in completing their project, may lead to more upfront organizational planning than would exist if the tools were not being used. The tools are designed to facilitate the execution of a well-planned project, not to compensate for poor planning. Given the investment of time learning how to use these tools and deciding how to use them requires, users have to decide for themselves if the potential benefits are worth the effort (Borgman, 2006).

**Future Research**

Although the use of these tools is considered standard for most OSS projects and the authors have realized their benefits in their own IR work, more empirical work needs to be done to better establish the extent of these benefits. How much time do these tools save in the long run? Do they improve the quality of output, or both? If so, to what degree do they do so? What tools are best suited to the various types of projects found in IR?
More work also needs to be done to modify the tools and methods for using them to take advantage of what is known about how users interface with these kinds of tools and what they expect from them (McAfee, 2006). Although some commercial versions of the tools discussed in this paper are a bit more user-friendly (though that almost always comes with a cost in flexibility), they were not designed with IR work specifically in mind. This results in additional planning when users have to figure out how to modify the existing systems to account for a missing feature that they need.

The main thrust of this paper has been to try to understand how the tools and methods currently used in most OSS projects contribute to project success and to consider how they can be used to benefit IR work. It was explained that these methods and tools facilitate and support effective project management practice, such as increasing project awareness, making it easier for people to contribute in ways they deem appropriate, creating a reusable knowledgebase, and rendering the project transparent for accountability purposes. If these methods and tools can be used to effectively manage highly complex software development projects with many contributors, it seems reasonable to determine if they can be used, in whole or part, to support relevant IR work. Of course, this requires considering how the IR context differs from the OSS context, a significant job for those interested in conducting further research on the topic.
References


