

# Research Infrastructure for Empirical Science of F/OSS

Les Gasser                  Gabriel Ripoché                  Robert J. Sandusky  
Graduate School of Library and Information Science  
University of Illinois at Urbana-Champaign  
{gasser,gripoche,sandusky}@uiuc.edu

## Abstract

*F/OSS research faces a new and unusual situation: the traditional difficulties of gathering enough empirical data have been replaced by issues of dealing with enormous amounts of freely available data from many disparate sources (forums, code, bug reports, etc.) At present no means exist for assembling these data under common access points and frameworks for comparative, longitudinal, and collaborative research. Gathering and maintaining large F/OSS data collections reliably and making them usable present several research challenges. For example, current projects usually rely on “web scraping” or on direct access to raw data from groups that generate it, and both of these methods require unique effort for each new corpus, or even for updating existing corpora. In this paper we identify several common needs and critical factors in F/OSS empirical research, and suggest orientations and recommendations for the design of a shared research infrastructure.*

## 1. Introduction

A significant group of software researchers is beginning to investigate large software projects empirically, using freely available data from F/OSS projects. A body of recent work, along with targeted assessments of researchers in the field, point out the pressing need for community-wide data collections and research infrastructure to expand the depth and breadth of empirical F/OSS research, and several rough proposals have been made [3].

This paper attempts to justify and clarify the need for community-wide, sharable research infrastructure and collections of data. We review the general case for empirical research on software repositories, articulate some specific current barriers to this empirical research approach, and sketch several community-wide options with the potential to address some of the most critical barriers. First, we review the range of research and research questions that could benefit from a research infrastructure and data col-

lections. Second, we expose critical requirements of such a project. We then suggest a set of components that address these requirements, and put forth several specific recommendations.

## 2. Objects of Study and Research Questions

As an organizing framework, we identify four main *objects of study*—that is, things whose characteristics researchers are trying to describe and explain—in F/OSS-based empirical software research: *software artifacts*, *software processes*, *development communities*, and *participants’ knowledge*. In Table 1 we provide a rough map of some representative characteristics that have been investigated for each of these objects of study, and show some critical factors that researchers have begun linking to these characteristics as explanations. It is important to point out that these objects of study are by no means independent from one another. They should be considered as interdependent elements of F/OSS (e.g., knowledge and processes affect artifacts, communities affect processes, etc.) Also, each of the outcomes shown in Table 1 may play a role as a critical factor in the other categories.

## 3. Current Research Approaches

We have identified two major approaches in empirical research on the objects and factors in Table 1:

- Large-scale cross-analyses of project and artifact characteristics, such as code size and code change evolution, development group size, composition and organization, or development processes [4, 5, 7].
- Smaller-scale case studies of specific practices and processes, for concept/hypothesis development and exposing mechanism details [1, 10].

These two orientations are separated less by fundamental differences in objectives than by technical limitations in

Objects	Success Measures	Critical Driving Factors
Artifacts	Quality, reliability, usability, durability, fit	Size, complexity, software architecture (structure, substrates, infrastructure)
Processes	Time, cost, complexity, manageability, predictability	Size, distribution, collaboration, knowledge/information management, artifact structure
Communities	Ease of creation, sustainability, trust, social capital	Size, economic setting, organizational architecture, behaviors, incentive structures
Knowledge	Creation, use, need, management	Tools, conventions, norms, social structures, technical content

**Table 1. Characteristics of empirical F/OSS studies.**

existing tools and methods. For example, qualitative analyses are hard to implement on a large scale, and quantitative methods have to rely on uniform, easily processable data. We believe these distinctions are becoming increasingly blurred as researchers develop and use more sophisticated analysis and modeling tools [9], leading to finer gradations in empirical data needs.

#### 4. Essential Characteristics

Empirical studies of software artifacts, processes, communities and knowledge are constrained by several key requirements. They should:

1. *Reflect reality* from actual experience rather than assumed, artificially constructed phenomena.
2. Give *adequate coverage* of naturally-occurring phenomena.
3. Examine *representative levels of variance* in key dimensions and phenomena.
4. Demonstrate *adequate statistical significance*.
5. Provide results that are *comparable across projects*.
6. Provide results that can be *repeated, tested, evaluated, and extended* by others.

Taken together, these six requirements for software research drive several requirements on the infrastructure and data for that research. For example:

- To satisfy the needs for reality and coverage (1,2), data should be *empirical and natural*, from real projects.
- For coverage of phenomena, demonstration of variance, and statistical significance (2,3,4), data should be *available in collections of sufficient size*.
- To allow for comparability across projects, and to allow community-wide testing, evaluation, and extension of findings (5,6), data and findings should be *sharable, in common frameworks and representations*.

#### 5. Available Empirical Data

F/OSS researchers have access to very large quantities and varieties of data, as most of the activity of F/OSS groups is carried on through persistent electronic media whose contents are open and freely available. The variety of data is manifested in several ways.

First, data vary in *content*, with types such as communications (threaded discussions, chats, digests), documentation (user and developer documentation, HOWTOs, FAQs, tutorials), and development data (source code, bug reports, design documents).

Second, data originates from different *media* sources, such as communication systems, version control systems, issue tracking systems, and content management systems.

Third, data can be found from various *locations*, such as community websites, software repositories and indexes, and individual project sites.

Most F/OSS project data is available as *byproducts* of development, maintenance, and system-use activities in F/OSS communities. Very little data is directly available in forms specifically intended for research use. This byproduct origin has several implications for the needs expressed above.

#### 6. Issues with Empirical Data

Many steps often have to be performed to identify, gather, and prepare data before it can be used for research. Data identification and preparation are important aspects of the research process and help guarantee that the six essential characteristics described above are met. The following steps are common barriers that most empirical F/OSS researcher will have to address:

##### Discovery and Selection

Because so much data is available, and because such a diversity exists in data formats and media, finding and selecting pertinent, usable data to study can be difficult. This is a general Resource Description/Discovery (RDD) and information retrieval issue, appearing here in the context of

scientific data. Appropriate information organization and metadata principles should ideally be employed in the original sources, but this is rare in F/OSS (and other software) data repositories, in part because of the byproduct nature of F/OSS research data.

### **Access and Gathering**

By access we mean the actually obtaining useful data once it has been discovered and selected. Access difficulties include managing administrative access to data, actually procuring data (e.g., overcoming bandwidth constraints) and dealing with difficulties transforming data in a useful format (such as a repository snapshot or via web scraping).

### **Cleaning and Normalization**

Because of the diversity of research questions, styles, methods, and tools, and the diversity of data sources and media available, researchers face several types of difficulty with raw data: original data formats may not match research needs; data of different types, from different sources or projects, may not be integrable in its original forms; and data formats or media may not match those required by qualitative or quantitative data analysis tools. In these cases, research data has to be normalized before it can be used. Data normalization activities may include data format changes, integration of representation schemas, transformations of basic measurement units, and even pre-computation and derivation of higher-order data values from base data. Normalization issues appear at the level of individual data items and at the level data collections.

### **Linked Aggregation**

Normalized data is critical for cross-source comparison and mining over data “joins”. However, some F/OSS-based research projects are exploring structural links and inferential relationships between data of very different characters, such as linking social network patterns to code structure patterns, or linking bug report relationships to forms of social order [10]. Linked data aggregation demands invention of new representational concepts specific to the kinds of data links desired for projects, and transformations of base data into forms compatible with those links.

### **Evolution**

Real projects continually evolve, both in content and in format: web sites are redesigned, tools are modified, etc. Research projects may have to track, adapt to, and reflect these changes. This can cause problems at many of the previous levels, as access rights can be modified, formats can change and links can be created or removed. In addition, trajectories of evolution themselves are actually an important object of study for some empirical software researchers. The central issue for this paper is how to adhere to the essential characteristics given above (such as the needs for

testable, repeatable, and comparable results) while reacting to and/or managing this evolution.

## **7. Addressing These Issues**

The main objective of a research infrastructure is to address community-wide resource issues in community-specific way [13]. For F/OSS research, the objective is to improve the collective productivity of software research by lowering the access cost and effort for data that will address the critical questions of software development research. In this section we offer some possible approaches to such an infrastructure, by first briefly describing each “component”, and then considering its benefits and drawbacks.

### **Representation Standards**

One of the broadest approaches to common infrastructure is the use of representation standards [13]. Such standards would move some issues of cross-source data normalization forward in the process that produces F/OSS projects’ information. For example, standard internal formats for objects such as bug reports could eliminate many internal differences between Bugzilla, Scarab, Gnats, etc., fostering simpler cross-analysis of data from these various repositories. Such representation standards would also facilitate exchange of data and/or processing tools within the F/OSS research community. For example, as part of our investigation of F/OSS bug reporting/resolution processes [9], we developed a general XML schematization of bug reports, derived from (but more general than) the Bugzilla internal database schema, and designed as a normalization target and translation medium for multiple types of bug reports from different systems [8]. Issues include the difficulty of developing, promulgating, maintaining, and enforcing such standards.

### **Metadata**

The use of metadata permits researchers to identify relevant characteristics of specific data collections. Metadata can serve numerous roles in the organization and access of scientific data and documents, including roles in location, identification, security/access control, preservation, and collocation [12]. Standardization of metadata and addition of metadata to F/OSS information repositories, especially at the point of creation, would let the research community identify much more easily the data used in each study, understand and compare data formats, and would also simplify the selection process, by making visible critical selection information. Fortunately, some metadata creation can be automated; unfortunately, representation standards are also an issue for metadata.

### **Tools**

Tools could potentially be developed to address each of the issues reviewed in the previous section. Some such tools

already partially exist in a generic form or are developed as needed by research groups. Tools such as web-scrapers that gather data, entity extractors that mine for specific entities like people and dates, or cross-references that link multiple information sources of a single project are commonly developed from scratch in each research effort. These tools are part of the basic toolbox of almost every empirical F/OSS researcher and could easily be provided as such. In fact, several nascent efforts are already underway to produce such tools (e.g. [6]).

Another contribution of a research infrastructure could be to place research data access and manipulation tools upstream, directly within software development tools used by the F/OSS community (e.g., CVS, Subversion, Bugzilla), instead of requiring sometimes-tedious and potentially risky post processing. For example, in most cases, F/OSS tools rely on databases for data storage and manipulation. These databases contain valuable information that is often lost during the translation to a web-visible front-end. (Usually the front-ends rely on web interfaces that display information in a user-friendly fashion but drop important structure in the process). Access to the underlying database can be much more valuable (and in many cases easier) than the current techniques of web-scraping that must recreate such missing relations post-hoc, and may not be successful.

### Centralized Data Repositories (CDRs)

Gathering specific snapshots of raw data and making them available to the research community from a controlled “cleanroom” location could provide benchmark data for comparative analyses and measurement of progress—a type of infrastructure that has proven invaluable in other disciplines. It would ensure that data parameters stay constant across studies, and through evolutionary stages of projects. Moreover, it might be easier in many cases to get a snapshot from such a repository than to go through all the steps of collecting the data directly from an F/OSS community. The CDR approach can have advantages of control, organization, and data persistence. However, this approach also raises the issues of *data selection* and *maintenance*. As with any managed information collection, CDRs would need *selection policies* to detail which materials from projects, tools and communities would be chosen for inclusion, and why [2]. The F/OSS community is already too large to attempt building practical evolving archives of *all* F/OSS projects (if such a notion were even meaningful). Selection necessarily induces bias, but careful selection would foster research on a shared body of data, possibly leading to more reliable findings. Second, *preservation policies* need development as F/OSS data is evolving quickly and collections will have to be maintained.

### Federated Access

Federating access is another approach to facilitating information sharing without making many redundant copies of original data, while maintaining local control over data access and organization. A central federation repository collects only metadata, and uses it to provide common-framework access to a variety of underlying sources. Federation has the advantages of distributed sharing, such as trading off lightweight central representations and sophisticated search infrastructure, against local data maintenance, information preservation, and access control.

### Processed Research Collections

Putting all the previous components together would lead to a set of normalized, processed and integrated collections of F/OSS data made available to the research community through either federated or centralized mechanisms.

### Integrated Data-to-Literature Environments

Finally, an advanced contemporary approach would be an attempt to connect both data sources and research literature in a seamless and interlocking web, so that research findings can be traced back to sources, and so that basic source data can be linked directly to inferences made from it. Such arrangements provide powerful intrinsic means of discovering connections among research themes and ideas, as they are linked through both citation, through common or related uses of underlying data, and through associations among concepts. Similar efforts are underway in many other sciences (e.g. [11, 13]). Networks of literature and data created in this way, with automated support, can reduce cognitive complexity, establish collocation of concepts and findings, and establish/maintain social organization within and across F/OSS projects.

## 8. Recommendations

In accord with the rationales outlined above and the strong sense of the F/OSS community [3], we recommend that F/OSS researchers begin collective efforts to create sharable infrastructure for collaborative empirical research. This infrastructure should be assembled incrementally, with activity in many of the areas defined below:

### Refine Knowledge

This paper has provided a sketch of some ideas toward robust and useful research infrastructure. The ideas and motivations here need more development, and collaborative efforts are encouraged.

### Exploit Experience

Many standards for sharable scientific data exist for other communities, as do many repositories of data conforming to those standards. We should do further research on what

other communities have done to organize research data. For example, many collections of social science data are maintained around the world<sup>1</sup>. We should use the experiences of these projects as a basis for the F/OSS research infrastructure. The success of these archives in the social science community is also a partial answer to questions of “why bother?”

### Instrument Existing Tools

We should work with existing F/OSS community development tool projects to design plugins for instrumenting widely used F/OSS tools (such as Bugzilla, CVS/Subversion, etc.) to make the content of those tools available via APIs in standardized formats, administratively controllable by original tool/data owners. Such an effort could also benefit the community of F/OSS developers itself; this sort of instrumentation could help interfacing multiple tools, projects, and communities, and might increase willingness to participate.

### Develop Data Standards

Standards for metadata and representation will help glue together data and tools such as finding aids and normalization tools. In collaboration with F/OSS tool developers, we should work toward standardizing formats and content of repositories of many kinds.

### Create Federation Middleware

Federated approaches to data archives will have much lower initial costs and will foster community building while maintaining local control over base data and sharing. Foundations for such middleware exists (e.g. in Digital Library frameworks such as Fedora).

### Develop Consensus on Data Selection Policies

We need much more consensus on what kinds of data provide the most utility for the widest variety of empirical F/OSS research projects. Developing this consensus will also help to congeal the community of empirical software researchers.

### Create Prototypes

As a proof of concept, we should mock up a complete F/OSS research infrastructure model embodying as many of the desired characteristics as feasible. Such a partial implementation might use, for example, a complete cross section of sharable information from a single project, including chat, news, CVS, bug reporting, and so on. We have already instigated some local efforts in a few of these areas, such as generalized bug report schemas, semi-automated extraction of social processes, preliminary data taxonomies, automated analysis tools, and others have also begun efforts in these directions [4, 6, 8, 9].

<sup>1</sup>See for example [http://www.iue.it/LIB/EResources/E-data/online\\_archive.shtml](http://www.iue.it/LIB/EResources/E-data/online_archive.shtml) for a list of such collections.

In the end, efforts in these directions will pay off in the form of deeper collaborations in the empirical software research community, wider awareness of important research issues and means of addressing them, and ultimately in more systematic, grounded knowledge and theory-driven practice in software development.

## References

- [1] M. S. Elliott and W. Scacchi. Free software development: Cooperation and conflict in a virtual organizational culture. In S. Koch, editor, *Free/Open Source Software Development*. Idea Publishing, 2004.
- [2] G. E. Evans. *Developing library and information center collections*. Libraries Unlimited, Englewood, CO, 4th edition, 2000.
- [3] L. Gasser and W. Scacchi. Continuous design of free/open source software: Workshop report and research agenda, October 2003. <http://www.isrl.uiuc.edu/~gasser/papers/CD-OSS-prelim-report.pdf>.
- [4] D. German and A. Mockus. Automating the measurement of open source projects. In *Proceedings of the 3rd Workshop on OSS Engineering*, Portland, OR, May 2003.
- [5] S. Koch and G. Schneider. Results from software engineering research into open source development projects using public data. *Diskussionspapiere zum Tätigkeitsfeld Informationsverarbeitung und Informationswirtschaft*, H.R. Hansen und W.H. Janko (Hrsg.), Nr. 22, Wirtschaftsuniversität Wien, 2000.
- [6] Libre Software Engineering tool repository. <http://barba.dat.escet.urjc.es/index.php?menu=Tools>.
- [7] A. Mockus, R. T. Fielding, and J. Herbsleb. Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology*, 11(3):1–38, July 2002.
- [8] G. Ripoche and L. Gasser. Possible bugzilla modification to create a Web-API for direct XML serialization of bug reports. SQA Project Memo UIUC-2003-20, 2003.
- [9] G. Ripoche and L. Gasser. Scalable automatic extraction of process models for understanding F/OSS bug repair. In *Proceedings of the International Conference on Software & Systems Engineering and their Applications (ICSSEA'03)*, Paris, France, December 2003.
- [10] R. J. Sandusky, L. Gasser, and G. Ripoche. Bug report networks: Varieties, strategies, and impacts in an OSS development community. In *Proceedings of the ICSE/MSR Workshop*, Edinburgh, Scotland, UK, 25 May 2004.
- [11] L. Shoman, E. Grossman, K. Powell, C. Jamison, and B. Schatz. The Worm Community System, release 2.0 (WCSr2). *Methods in Cell Biology*, 48:607–625, 1995.
- [12] T. R. Smith. The meta-information environment of digital libraries. *D-Lib Magazine*, July/August 1996.
- [13] S. L. Star and K. Ruhleder. Steps toward an ecology of infrastructure: Design and access for large information spaces. *Information Systems Research*, 7(1):111–134, 1996.