A Study of Clinically Related Open Source Software Projects

Michael A. Hogarth, MD and Stuart Turner, DVM MS
Biomedical Informatics Research and Consulting Service
UC Davis School of Medicine, Davis, CA

ABSTRACT

Open source software development has recently gained significant interest due to several successful mainstream open source projects. This methodology has been proposed as being similarly viable and beneficial in the clinical application domain as well. However, the clinical software development venue differs significantly from the mainstream software venue. Existing clinical open source projects have not been well characterized nor formally studied so the 'fit' of open source in this domain is largely unknown. In order to better understand the open source movement in the clinical application domain, we undertook a study of existing open source clinical projects. In this study we sought to characterize and classify existing clinical open source projects and to determine metrics for their viability. This study revealed several findings which we believe could guide the healthcare community in its quest for successful open source clinical software projects.

INTRODUCTION

Open source software development has attracted significant interest over the past several years (1). Several highly visible mainstream open source projects have been successful including Linux, Apache, Firefox, Perl, Python, MySQL, PostgreSQL, as well as key Internet infrastructure components such as sendmail and BIND (2). Open source is not a new phenomenon and dates back to the earliest days of software development (3). Nevertheless, recent successes have demonstrated that open source development can deliver high quality mission-critical software.

The success of open source development in mainstream software has attracted interest in this methodology for the development and distribution of clinical applications (4, 5). Several advantages of open source have been suggested for the clinical application domain. Namely, that open source would reduce EMR ownership costs, that the risk of vendor disappearance is not present in open source projects, and that open source projects are more likely to adhere to standards for compatibility and data interchange (6). Although these assertions could indeed be correct, empirical data is lacking to validate them.

The clinical application domain differs from the mainstream software community. The success factors in play in the mainstream open source community may or may not translate directly to the clinical software community. For example, clinical applications require specialized healthcare knowledge by the development team, frequently requiring participation by individuals with clinical practice experience. The clinical domain also involves many complex workflows among multiple clinical information providers and stakeholders leading to software systems that are large and highly complex. As a result, open source may not be as successful in the clinical software venue as it has been elsewhere (7).

Although mainstream open source software development is a very active area of sociological and computer science research, to our knowledge open source in the clinical application domain has not been similarly studied. There are several important research questions that are as yet unanswered. Does the clinical software developer community 'milieu' have the right confluence of factors to make open source a viable development strategy? What are the current characteristics of developers of clinical open source applications and what motivates them to spend considerable effort in these projects? Are these the same motivating factors as have been found in the mainstream open source movement (8)? Are the clinical open source developers contributing without compensation or do they mirror the Linux community where up to 23% receive monetary compensation for their work on open source components (9). Although these questions remain largely unanswered, initiatives are currently underway to promote open source clinical application development (10). Understanding the differences and similarities between clinical systems open source development and mainstream open source development could lead to optimization of these investments.

To begin to characterize open source development within the clinical application domain, we undertook a study of current open source software projects that are clinically focused. Our goals in this initial study were modest and were to (1) identify known open source clinical applications, (2) collect basic information about these projects, and (3) apply a metric for determining the ‘viability’ of a particular open source project.

METHODS

Identification of study cohort:
Identifying existing open source clinical software projects was a non-trivial task requiring several ‘harvesting’ strategies. We primarily relied on existing open source project repositories such as SourceForge (11) and Freshmeat (12). Sourceforge is the largest repository of collaborative open source projects and at the time of this writing contains information on over 97,000 projects including many in the healthcare domain. In addition, we used several lists of open source clinical software applications. To maximize discovery of clinical open source projects, we also conducted Internet searches using the Google search engine with the keywords ‘medicine’, ’software’, and ‘open source’. Our primary sources of information are listed in Table 1.

Sourceforge	http://www.sourceforge.net
Freshmeat	http://www.freshmeat.net
Sourcewell	http://sourcewell.brelios.be
MedSource	http://www.medsource.com/open_ source_links.html
O’reilly’s Open Source	http://www.osdir.com
Linux Med News	http://www.linuxmednews.com
Yves List	http://homeusers.brutele.be/ ypaindaveine/opensource/ inventory.html
Open Source Healthcare Alliance	http://www.oshca.org
Spirit Project	http://www.euspirit.org/
Open Health	http://www.openhealth.org
Google	Query: +medicine +software +”open source”

Table 1: Medical Open Source Resources

Dates:
The harvesting of existing clinical open source applications was conducted from February 2003 – July 2003.

Inclusion criteria:
Two fundamental criteria for inclusion in our study were: (1) the software project had to be *clinically relevant* and (2) the software had to be *open source*. A project was clinically relevant if the software had an unambiguous intent to support the clinical care of patients. By this we mean the software had as its primary intent the generation, management, storage, or manipulation of information used to perform clinical care on a patient. This is in contrast to information related to the patient but not used for direct clinical care, such as billing or financial information. If an application was principally designed to support biomedical research it was excluded.

We used the following three criteria to determine whether the software project was “open source”:

- There were no license restrictions on the redistribution of the software for sale or gratis
- There were no license restrictions on the ability to modify the software and create derivative works
- The source code was readily available

Although a prevailing criterion in mainstream open source software is the use of an open source license approved by the Open Source Initiative (13), we found several open source clinical projects that adhered to the basic open source licensing principles but did not include one of the OSI licenses in the distribution. As a result, if the project provided source code and lacked an exclusionary license with regards to redistribution and ability to modify, it was included in the study.

Although ready availability of source code is not commonly an explicit criterion (since it is implied), we found several clinical application projects that purported to be ‘open source’, yet did not make source code available. If projects did not make source code readily available through some mechanism, we excluded the project from the study.

Data collection:
We collected a basic set of attributes for all projects in the study. These attributes included license type, operating system targets, programming language used, and software maturity level. For those projects listed in Sourceforge, additional attributes were collected including the dates the project was first listed, dates of subsequent releases, and total number of downloads over the lifetime of the project.

Vitality Score:
For projects hosted on Sourceforge and where the information was available, we used the attributes of product releases, total age of the project, and time since the last release to calculate a vitality score (14, 15). The vitality score is designed to provide a measure of the relative development activity of a project and is calculated as follows:

$$\text{Vitality Score} = \frac{\text{Number of Product Releases X Age (days)}}{\text{Time Since Last Release (days)}}$$

In projects with only one release, the score will always be 1.0 despite its age.

Application Classification:

We defined fifteen functional classes specific to the medical domain based on the predominant functionality of the projects: *Clinical Information System / Electronic Medical Records System; Messaging; Continuing Medical Education; Data Acquisition, Collection & Reporting; Database Management; Decision Support;; Imaging; Issue Tracking; Laboratory Information System; Medical Process or System Automation; Pharmacy; Practice Management; Public Health; Quality Management; Scheduling & Workflow; Standards, Terminologies & Vocabularies, Telemedicine, Security; Community, Chat or Networking;* and *Infrastructure*.

RESULTS

Data was collected on a total of 218 projects. Pruning of projects that had insufficient information or those found to not match the inclusion criteria yielded a set of 179 open source projects for further study.

Development Status:
Software development status was categorized using the status explicitly stated by the developer. In 61 projects (34.1%), this was not stated or was unavailable. The majority of these were projects which had announced themselves and established a distribution site, but had not undergone release of any versions. For some, distribution of code was ad hoc and lacked sufficient information to effectively match the stage of code release to any status.

Status	Frequency	Percent of total
Not determined	61	34.1
1 – Planning stage	24	13.4
2 – Pre-alpha	13	7.3
3 – Alpha	20	11.1
4 – Beta	35	19.6
5 – Production/Stable	23	12.8
6 – Mature	3	1.7
Total	179	100

Table 2: Development Stage/Status

Project age:
Only 71 projects could have their project age reliably determined. The majority of the projects in the study were in the beta stage of development (19.6%). Only three projects (1.7%) described their systems as being *mature*.

Application Type:
Clinical Information Systems/Electronic Medical Records Systems were the dominant application category and represented 25.1% of all projects surveyed (Table 3). Software related to clinical imaging was nearly as prevalent and represented 22.9%. Decision support systems were present but in relatively low frequency at 8.4%.

Application Type	Number	Percent of Total
Clinical Information System or Electronic Medical Record System	45	25.1
Data Acquisition	16	8.9
Database Management	8	4.5
Decision Support	15	8.4
Imaging System	41	22.9
Messaging/HL-7	11	6.2
Practice Management	5	2.8
Scheduling, Workflow	5	2.8
Terminology related software	9	5.0
Other	24	13.4
Total	179	100

Table 3: Application Type

Programming Languages:
When combined, the C-type languages were clearly the preferred programming language among developers of clinically related open source software and C++ was the most widely adopted among these. The second most commonly used language was Java, followed by PHP, then Delphi/Kylix, Perl, and Python respectively. Many applications noted the use of more than one programming language.

Operating Systems:
Thirteen different operating systems (OS) were identified and 30% of the projects listed more than one type of OS. Because of the heterogenous nature of the various Linux and Unix operating system distributions, these were collectively classified under POSIX (Portable Operating System Interface) and represented the majority of target operating systems at 33%. Next was *Operating System Independent* at 28% followed by *Windows* (32%), *Sun/Solaris* (4%), and *MacOS* (3%) .

Software Change Management Infrastructure:
A majority of the projects surveyed were found in Sourceforge, which provides tools to manage the software development lifecycle (SDLC). A smaller number of projects used an ad-hoc method of source file distribution (9.5%). A significant number of projects (24.6%) had not released any files, or did not use a version control system, or did not have their files readily available online and versioned. This suggests that a significant fraction of open source projects in the clinical

domain fail to reliably adhere to minimal software engineering development standards.

Downloads:
Information on the total number of downloads during the project lifetime was obtained when available. This information was available for 118 (85%) projects, with these all being harvested from Sourceforge. Of these, the project with the highest number of downloads was *OpenEMed* with 10,408 over its 38 month lifetime. However, the project with the highest monthly download rate was the *Hospital DBMS Project* with an average of 412 downloads/month, or twice the rate of *OpenEMed*.

Project Name	Downloads	Application Type	Age in Months	Status
OpenEMed	10,408	Clinical Information System	38	Beta
Eviewbow DICOM Java Product	7,322	Imaging	40	Beta
Tk_familypractice	7199	Clinical Information System	42	Beta
ImLib3D	6,820	Imaging	26	Beta
OpenEEG	5,594	Data Acquisition, Collection & Reporting	22	Beta
dcm4che	5,429	Imaging	21	Alpha
Hospital DBMS Project	4,947	Database Management	12	Beta
HAPI	4,678	Messaging/Hl7	20	Alpha
Meditux	3,987	Data Acquisition, Collection, Reporting	34	Production/ Stable
Res Medicinae	2,903	Clinical Information System	24	Pre-Alpha

Table 3: Top Downloads (Sourceforge listed projects)

Vitality Score:
The vitality score for projects in Sourceforge was calculated. The average vitality score was 121 with a maximum of 1,992. When accounting for numbers of developers, the highest mean vitality score was for projects listing four active developers. This suggests that a critical mass of four developers may be an important metric regarding the viability and sustainability of a project.

Evolution Metrics:
Releases were determined by reviewing the chronological listing of source and/or binary code releases, whether the release was for a core program (or kernel) or for a dependent module. The mean time from announcement of the project to first code release was 128 days (median = 24 days) with a maximum of 1,009 days for one project (Figure 3). The longer a project dwelled in the planning or pre-alpha stages, the less likely it would evolve toward the later development stages and actual release of source code.

Growth of new open source projects:
The growth of new open source projects in the clinical application domain shows a linear rise beginning five years ago and continuing at a constant rate of approximately 25% per year (Figure 4).

DISCUSSION

Open source software development is an intriguing model for software development and distribution. Given the development cycle of clinical information systems can be as long as five years from conceptual planning to first version, some systems may be fairly outdated before they see “first light” in a clinical venue. In addition, the domain knowledge required to build such a system may add considerably to the development costs. The open source approach, with its philosophy of collective effort and rapid iteration could significantly shorten this lifecycle while reducing development costs. Recent studies of successful mainstream open source projects suggest that a vibrant open source community requires a confluence of factors, some of which may not necessarily exist or translate directly to the prevailing clinical software application development market in the U.S. As a result, current initiatives to spur the development of open source clinical applications may not necessarily result in higher numbers of robust open source clinical systems.

This study is a first attempt to characterize the current state of clinically related open source software and to potentially gain insights into how those interested in nurturing this methodology could optimally deploy their resources.

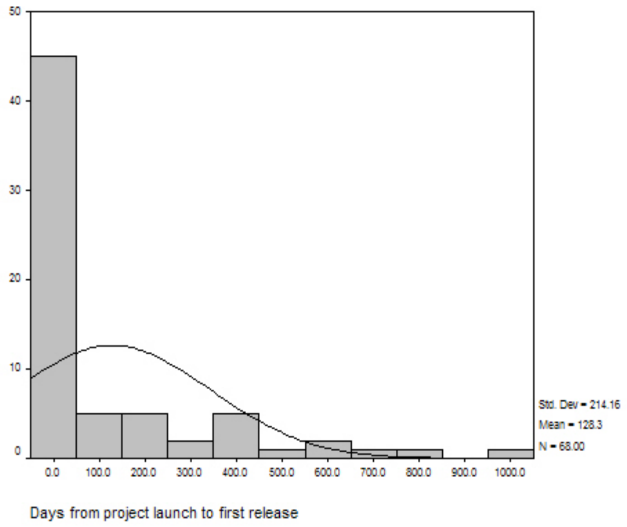

Figure 3: Project evolution

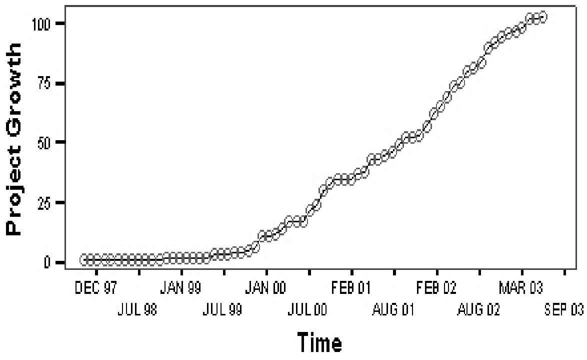

Figure 4: Growth of clinical open source

The study revealed the following preliminary observations:

1. A significant fraction of clinical open source projects "die on the vine" without producing a deliverable
2. The largest segments of prevailing open source clinical applications are clinical information systems and imaging systems. Together they account for nearly half of all open source clinical software discovered in this study.
3. If a project delays releasing a deliverable more than 100 days beyond the first announcement, it is unlikely to ever produce one.
4. Having four developers seems to correlate with a more viable project (highest vitality score/index)

There remain many unanswered research questions regarding clinical open source systems. Do the benefits of open source in the mainstream community translate equally to the clinical software community? What are the motivations of clinical open source developers and are they equivalent to those found in the mainstream open source communities? Are there viable business models for clinical open source software that would result in the success seen in mainstream open source? These are important research questions that should be investigated as the healthcare informatics community looks to open source development as a potential solution for some of the current challenges in developing and deploying clinical information systems.

We hope this study encourages further research into the socio-technical aspects of open source clinical system development such that investments in time, funding, and effort in this area can be ideally targeted to maximize their effects.

REFERENCES

1. Weber S. The Success of Open Source. Cambridge, Massachusetts: Harvard University Press; 2004.

2. O'Reilly T. Lessons From Open-Source Software Development. Communications Of The ACM 1999;42(4):33-37.

3. Feller J, Fitzgerald B. A History of Open Source Software. In: Understanding open source software development. London: Addison-Wesley; 2002.

4. McDonald CJ, Schadow G, Barnes M, Dexter P, Overhage JM, Mamlin B, et al. Open Source software in medical informatics--why, how and what. Int J Med Inform 2003;69(2-3):175-84.

5. Carnall D. Healthy Outlook. Linux User 2000 July 2000:42-44.

6. Kantor GS, Wilson WD, Midgley A. Open-source software and the primary care EMR. J Am Med Inform Assoc 2003;10(6):616; author reply 617.

7. Stahl MT. Open-source software: not quite endsville. Drug Discov Today 2005;10(3):219-22.

8. Bonaccorsi A, Rossi C. Why open source software can succeed. Research Policy 2003(32):1243-1258.

9. Hertel G, Niedner S, Herrmann S. Motivation of software developers in open source projects: an Internet-based survey of contributors to the Linux kernel. Research Policy 2003(32):1159-1177.

10. Beckley E, Versel N. Alliance for affordability; Family-friendly group brings IT costs within reach. Modern Physician 2003 Dec 1.

11. SourceForge. http://www.sourceforge.net

12. Freshmeat. http://www.freshmeat.net

13. OSI. http://www.opensource.org

14. Crowston K, Annabi H, Howison J. Defining Open Source Software Project Success. In: 24th International Confererence on Information Systems; 2003 December 2003; Seattle, WA; 2003.

15. Freshmeat. Vitality Score FAQ. http://freshmeat.net/faq/view/27