MEDINFO 2001
V. Patel et al. (Eds)
Amsterdam: IOS Press

How the Open-Source Development Model Can Improve Medical Software

Thomas R. Yackel

Oregon Health Sciences University, Portland, Oregon, USA

Abstract

The current system of proprietary software development for healthcare applications is inadequate to meet the needs of providers, administrators, and patients. A recent advance in the field of Internet programming is the release of large-scale software projects as "open-source." The advantages of this method of development include higher quality, lower cost, and increased adherence to established standards. There are several hurdles to be overcome before the healthcare field can take full advantage of this development model. The implementation of open-source development of medical software could greatly improve clinical and research software and elevate the academic standards of the field of medical informatics.

Keywords:

Medical Informatics; Software Design; Quality Control

Introduction

After thirty years of development and countless sums spent on research and implementation, the state of medical software is inadequate to meet the needs of today's patients, providers, and healthcare administrators. Many medical software implementations are unsuccessful [1], [2], [3] and in many countries the penetration of computer systems into clinical care is low. [4], [5] Although few publish their failures, anecdotal reports shared at conference meetings confirm the dissatisfaction of most informaticians, clinicians, and administrators with currently available software products. [6] The lack of value provided by these systems reflects poorly on the field of medical informatics, which has largely ignored the issue. [7]

Efforts to study this problem are hampered by a lack of access to real-world systems by informaticists. Many informatics research programs are separate from the divisions that actually run the software and provide services to users. Due to proprietary agreements with software developers, many students lack experience with clinical information systems, since the most successful systems are not available for study outside the institutions where they were created.

These shortcomings have not lessened the demand for informatics applications that improve care and reduce cost. The Institute of Medicine report on medical errors suggests the use of clinical support systems to reduce medical mistakes. [8] In the US, private industry is taking steps to require decision support systems for those enrolled in their insurance plans to lessen errors and drive down costs. [9]

The expenses associated with medical computing systems have led to the development of software products distributed to clinicians free of charge. The business model for these projects may include providing aggregate data to drug companies or other interested third parties in order to finance development. [10] This places the users of these programs in the dubious ethical position of disseminating patient information instead of protecting it.

It is ironic that we have been unable to produce innovative, quality software for an industry that itself produces large amounts of high quality research and services. Through a system of public and private financing, open distribution of discoveries, and peer-review, the field of medicine has rapidly advanced and provided great value to society.

Unlike medical scientific discovery, our current system of medical software development has not kept pace with the needs of patients and healthcare providers. Clearly, new methods of delivering valuable software to the healthcare community are necessary. Is it possible that by treating medical software as an entity similar to medical knowledge, the field of medical informatics could advance as swiftly?

A small but growing community of medical software developers would enthusiastically answer, "YES!" Drawing from recent successes in a form of software development known as "open-source," some developers are beginning to release their medical software "discoveries" with freedoms that allow the successes to be peer-reviewed, replicated, and distributed at little or no cost. This essay will examine the open-source software development model and its application to medical computing, exploring the potential benefits for medical software users. It will also discuss the obstacles facing this method of medical software development, with an emphasis on practical solutions.

What is the Open-Source Software Development Model?

In this section, we explore the open-source software development model to gain an understanding of how and why it works.

Characteristics of an Open-Source Project

Most successful open-source projects share certain characteristics: a unique form of copyrighting, known as "copylefting," active programmers and contributors, and a modular design that allows parallel development. The project leader, who may be an individual or group, encourages these characteristics and provides a project vision that unifies the development team.

Copylefting

The philosophical justification for open-source is the belief that users have certain inalienable rights regarding the use of software. Above all, users should be free to access their data and the code that controls it. Richard Stallman, founder of the Free Software Foundation, [11] notes that "Free Software"[1] incorporates four specific freedoms:

- *The freedom to run the program, for any purpose (freedom 0).*
- *The freedom to study how the program works and adapt it to your needs (freedom 1).*
- *The freedom to redistribute copies so you can help your neighbor (freedom 2).*
- *The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3).* [12]

Stallman is careful to make the distinction that this freedom is akin to "free speech," not "free beer." In this context, "free software" is software with minimal restrictions for programmers and users. Each open-source project acknowledges these freedoms through the use of a particular type of license, known as a copyleft. The copyleft is a copyright to which specific distribution terms are added. These terms grant people the right to "use, modify, and redistribute the program's code *or any program derived from it* but only if the distribution terms are unchanged." [13] The distribution terms go on to stipulate that anyone who redistributes the software grants others the same freedoms to copy and change the code. Through the use of copylefting, a piece of code becomes joined to the legal instrument that enforces the freedoms enumerated above. Copylefting does not preclude individuals from charging fees for developing customizations or specific installations and implementations. These developments, however, are released back to the community at no cost for potential incorporation in the next distribution. A specific and popular application of copylefting is the General Public License or GPL. [14]

Programmers and Other Contributors

The programmers who have copylefted their software often develop works that initially satisfy a personal need. In his seminal work, *The Cathedral & the Bazaar,* Eric S. Raymond states, "Every good work of software starts by scratching a developer's personal itch."[15] In this regard, these software developers are distinguished from those writing software for the needs of another person. To a certain extent, the real value of the software created by these authors comes from completing a particular task with their code. After the software is written, releasing the work with a copyleft grants them certain benefits.

The first realized benefit is that many other programmers will scrutinize their source code. This form of peer review elicits bug fixes or other improvements that help the programmer who released the work. Simultaneously, if the software is of good quality, programmers enhance their standing in the community. [15] One can now easily see that many of the same reasons for publishing a medical discovery motivate the release of a work to the open-source software community, that is, to enhance one's reputation in peer-circles and to improve the ultimate product (be it knowledge or software).

Since most open-source projects involve programmers solving problems that relate directly to their work, they usually are domain experts in their field. For example, many of the contributors to the Apache web server product are web site administrators who adapted the program to meet their particular needs. Other contributors are part-time programmers who make their living through other means. These "hobbyist" coders are treated as co-developers and are involved in many of the important decisions regarding the software project.

Modular Design

A modular programming structure is also important, for it allows for easy division of labor, and object-oriented languages have helped promote quality open-source software.

Project Leader

The most crucial component of a successful open-source project is a dynamic and visionary project leader. The leader may be the original programmer or someone who has picked up a project that was previously abandoned. The "manager" in charge may actually be a group that meets via email. Regardless of the form, the project manager provides the leadership that encourages programmers and other contributors to remain with the program. This person or group will solicit suggestions for improvements and decide which to accept, rewrite, or reject. The project manager creates an environment that encourages cooperation and prevents "forking," or schisms that result in multiple development teams who create derivative projects.

Other Characteristics

Other characteristics of a successful open-source project include: the distribution of the programming work among numerous volunteers, the use of an inexpensive network of communication connecting the participants, and the ability to divide the project into distinct units. It is not surprising

[1] Stallman distinguishes "free software" and "open-source software" (see http://www.fsf.org/philosophy/free-software-for-freedom.html). For the purposes of our discussion, we will consider them synonymous.

that nearly all open-source projects make extensive use of email and the Internet.

Benefits of Open-Source Software in Medicine

There was a time when medical knowledge was considered proprietary, as medical software is considered today. In the proprietary medical era, local healers would go to great lengths to protect discoveries. Patients would seek out these providers based on their perceived reputations of quality. Eventually, however, it was recognized that society would derive greater benefits if medical knowledge was open and free to be shared. And despite this free sharing of information, physicians continue to obtain high salaries. Instead of relying on their proprietary knowledge to produce income, clinicians share their knowledge base and provide value by applying that information to individual circumstances. [16]

Today, medical discoveries are published and freely available. Occasional practitioners that promote secret and unique forms of treatment are labeled "quacks" by mainstream clinicians. [16]

Although quacks are few in the medical discovery arena today, we still live in an era of medical *software* quackery. Software is created and then hidden from view in order to protect "trade secrets." The result of this protection has not been the production of more and better medical software. Instead, we suffer from mediocre software and a stifled academic milieu surrounding informatics.

Let us consider what the medical community actually requires of its software. Most would agree that important characteristics of medical software include reliability, usability, and maintainability. The software should easily interact with other programs to allow sharing of data. A program must somehow be "future-proof," so that should the vendor go out of business, it will still operate properly. The software should be easily customizable so that it fits into the myriad of care systems that currently exist. The data about patients should not be subject to mining by third parties, such as pharmaceutical and insurance companies, to finance the cost of the software. Finally, it would be ideal if informatics students and researchers could study the code and possibly improve it.

Many of the same qualities of good medical software are attributes of good software in general. In "Why Software Should Not Have Owners," Stallman provides a succinct analysis of society's requirements of software in general:

> *What exactly does society need? It needs information that is truly available to its citizens – for example, programs that people can read, fix, adapt, and improve, not just operate. But what software owners typically deliver is a black box that we can't study or change.* [17]

Most medical software today very much resembles the "black box" to which Stallman refers. Users are unable to freely access their data without paying a tax to the developer. The cost of updates may be prohibitive or it may be impossible for the purchaser to get the customizations they desire due to the monopoly imposed by the vendor. [15] This has resulted in unfortunate consequences, including the cancellation of large software projects. [3]

There are several ways in which the open-source software development model more adequately addresses the needs of the medical community. The most significant benefits are:

1. *Fewer bugs.* Open-source software tends to be more reliable because, "given enough eyeballs, all bugs are shallow." [15] Just as medical discoveries prove their value through peer review, open-source software is validated by thorough inspection by programmers outside the medical software's development team.
2. *Encouragement of standards.* Open standards are propagated by open-source, since all developers have equal access to the standard. [15]
3. *Reduced overall cost.* The overall cost tends to be lower because development is distributed among many users.
4. *Improved evaluation.* Measurements of quality applied to proprietary software are inexact because evaluators do not have access to the source code. With open-source software, however, outside reviewers can determine not only whether a particular piece of software works, but also the difficulty of the problem it attempts to solve, whether that problem was solved with a clever method, and whether the code can be useful for other programming tasks. [11] Since the products are open to be studied and replicated by informaticians, the status and importance of medical informatics as an academic endeavor is elevated and the overall quality of the available products improves.
5. *Infinitely supportable/customizable.* When the source code is available, the problem of "vendor-lock" disappears. As long as programmers can be hired, the software can be customized and supported.

Despite the many advantages of the open-source development model, there are challenges to overcome. First, there are few medical open-source projects that are beyond the prototype stage. This lack of large-scale open-source medical implementation introduces doubt as to open-source's feasibility in the medical arena.

Another problem with open-source projects is that implementation often requires a great deal of technical expertise. [11] In the world of medical software, however, some question whether any solution is truly "turnkey." A recent article in JAMIA suggests that, "all EMRs are 'in development,' at least in regard to their ability to serve an integrated health care system." [3] A commercial solution, however, clearly identifies the entity that must perform the development: the vendor.

Having the power to upgrade and customize a software product may not always be an advantage. Johnson has pointed out that a rarely acknowledged benefit of proprietary software is the displacement of information technology development outside the walls of a healthcare institution. This tends to reduce the political infighting that can occur with a large informatics project. [16]

Additionally, not all projects are appropriate for open-source development. In particular, programs that provide content may be more appropriately designed as proprietary systems. An example may be a drug database or any code that is unique to an organization. [16]

Despite these exceptions, one can argue that open-source development more closely approximates the needs of the medical community in the delivery of software. The benefits of "peer-reviewable" code on reducing software errors and the ability for individual institutions to easily customize an application to suit their particular needs would greatly enhance the likelihood that software would be adopted and used in medicine. A question that remains, however, is whether the incentives are great enough to encourage wide-scale development of quality medical software following the open-source model.

From Here to There: Implementing an Open-Source Development Model

Despite the arguments in favor of the open-source development of medical software and the successes realized by its implementation in other areas of information technology, there are serious hurdles that must be overcome for this model to be accepted by a large number of institutions and developers. The five principle challenges to this model are: 1) the proprietary mindset of most involved with developing, using, and studying medical software, 2) the public funding of proprietary software projects, 3) the issue of software copyrighting in academia, 4) a technologically complex domain that makes modularization of software projects difficult, and 5) a limited base of potential programmers. Let us further examine each of these challenges.

Proprietary Mindset

We have grown up in a culture where software is treated in much the same way as other media, such as print, video, and audio. It is a firmly entrenched belief that the only way to generate profits from software is from the initial licensing. Yet software behaves in way fundamentally different from other media. One demonstration of this is to note that the price of software is bound by the expected future value of vendor service. [15]

If one purchases a book, a movie, or an audio CD, the value of the item is independent of the publishing firm or even the artist who produces it. These items have value for their own sake. Should the author die, it is not uncommon for the value of these works to rise. But what happens to the value of a software product when its author dies or the software company declares bankruptcy? Except for categories of software that require no maintenance, such as software games, the value declines rapidly. This can be explained by the lack of future service that such an event foretells. What good is a proprietary hospital information system if the vendor goes out of business? Since it cannot be updated, it has very little value. This underscores the fact that the true value of software does not lie in the initial product, but in the ongoing implementation and maintenance of that product. It has also been noted by some authors that the majority of computer programmer time is spent not in initial coding, but on maintenance. [15] Unhindered access to source code allows self-support, regardless of the fate of the software vendor.

With this understanding, it is not hard to comprehend why those who subscribe to the open-source model release their source code for free. Widespread use of software *increases* its value by exposing bugs and thereby reducing maintenance costs. [15] Users who understand the software well enough will contribute these bug fixes while they get to use the software for free. Those who wish to have a third party implement the product are likely to hire the most knowledgeable programmers who, coincidentally, are those most involved in writing the current code.

Overcoming the bias of the software user community will take time. With the current lack of academic evaluation of medical software, many still equate quality with price. [18] As more open-source projects, from within and outside of medicine, become mainstream, users will start to understand the value of unhindered access to the source code of the programs that contain their data.

Public Funding of Proprietary Software Projects

If medical software is viewed as a form of medical knowledge, it will be necessary to publicly fund software development. In the United States, the National Library of Medicine and other groups already fund various activities in medical informatics, including numerous informatics training programs. Many of the software products developed in these training programs, however, are retained as proprietary. In order to recognize the greater societal good presented by open-source software, it should be a requirement that *any software development funded by the public be released as an open-source project* with the General Public License. Opening the source of such software would not only benefit the potential users of the product but also the training programs themselves, since it will provide more material for academic study and will legitimize medical informatics as an academic endeavor.

Software Copyrighting in Academia

A corollary to providing public funding for open-source software projects would be the revision of current agreements signed by academics so as to allow them to publish their work under the General Public License. Currently, special permission is required for works to be released from many academic programs using the GPL.

Technological Complexity of the Medical Domain

A major hurdle for any medical software project to overcome is the complexity of the medical domain itself. Project modularization is difficult due to the large number of interactions that occur between the various objects of a medical environment. Special coordinating networks must be developed to allow inclusion of a large number of programmers on the development team.

The Shallow Pool of Programming Talent

The most successful open-source projects are those where computer programmers solve a problem occurring in their day-to-day work. [19] Unfortunately, the medical field is notoriously skeptical of information technology and very few of its members are able to program computers. [20] In order for an open-source project to take root, there must be a base of qualified programmers who support it. The current barriers to using open-source programming tools must be lowered so that more physicians and others with domain knowledge of the healthcare field can participate in open-source software projects.

Conclusion

The demand for robust and less costly medical software will increase as administrative requirements, medical knowledge, and quality of care directives expand. Our current method of producing medical software requires revolutionary change in order to meet the needs of today's patients, providers, and healthcare administrators. The open-source software development model has the potential to deliver inexpensive software of higher quality than can be provided by proprietary vendors. It is time to lay the necessary foundation that will encourage the use and development of open-source software in medicine.

References

[1] Williams LS. Microchips versus stethoscopes: Calgary hospital MDs face off over controversial computer system. *Can Med Assoc J* 1992: 147(10): pp. 1534-47.

[2] Dean M. Unhealthy computer systems [news]. *Lancet* 1993: 341(8855): pp. 1269-70.

[3] Goddard BL. Termination of a contract to implement an enterprise electronic medical record system. *J Am Med Inform Assoc* 2000: 7(6): pp. 564-8.

[4] Lenhart JG, Honess K, Covington D, Johnson KE. An analysis of trends, perceptions, and use patterns of electronic medical records among US family practice residency programs. *Family Medicine* 2000: 32(2): pp. 109-14.

[5] Iakovidis, I. From Electronic Medical Record to Personal Health Records: Present situation and trends in European Union in the area of Electronic Healthcare Records. *Medinfo*. 1998: 9 Pt 1: suppl 18-22.

[6] Lorenzi NM, Riley RT, Blyth AJC, Southon G, Dixon BJ. Antecedents of the people and organizational aspects of medical informatics: review of the literature. *J Am Med Inform Assoc* 1997: 4(2): pp. 79-93.

[7] Braude RM. People and organizational issues in health informatics. *J Am Med Inform Assoc* 1997: 4(2): pp. 150-1.

[8] Kohn LT, Corrigan J, Donaldson MS. Institute of Medicine (U.S.) Committee on Quality of Health Care in America. *To err is human: building a safer health system.* Last updated: [1999-11-24] Washington, DC: Nat Acad Press, 1999.

[9] Martinez B. Business consortium to launch effort seeking higher standards at hospitals. *The Wall Street Journal* 2000 Nov 15: Sect. A-3, A-12.

[10] Riben, M. Personal communication. [Nov 12, 2000].

[11] Lerner J, Tirole J. The simple economics of open source. [web page] http://papers.ssrn.com/paper.taf?abstract_id=224008. [Accessed Nov 17, 2000].

[12] What is free software? [web page] http:// www.fsf.org/philosophy/free-sw.html. [Accessed Nov 12, 2000].

[13] What is copyleft? [web page] Nov 28, 2000; http://www.gnu.org/copyleft/copyleft.html. [Accessed Nov 15, 2000].

[14] GNU Lesser general public license. [web page] http://www.fsf.org/copyleft/lgpl.txt. [Accessed Nov 28, 2000].

[15] Raymond ES. *The cathedral and the bazaar : musings on Linux and open source by an accidental revolutionary* 1st ed.. Cambridge, Mass.: O'Reilly, 1999.

[16] Johnson DL. Open-source medical information management. [web page] 1999; http://lorenzo.uwstout.edu /QQMIM/medicalfreesource. html. [Accessed Nov 15, 2000].

[17] Stallman R. Why software should not have owners. [web page] Oct 5, 2000; http://www.gnu.org/philosophy/why-free. [Accessed Nov 15, 2000].

[18] Head M. Barriers to open source use in medicine persist.[web page] Nov 10, 2000; http://www.Linuxmednews.com /stories.php?story=00/11/10/7975019. [Accessed Nov 14, 2000].

[19] Carnall D. Healthy outlook. *LinuxUser* 2000:42-4.

[20] Gage JS. Does open source have a future in medicine? *MD Computing* 1999: 16(5): pp. 52-3.

Address for Correspondence

Thomas R. Yackel, MD, MPH
Oregon Health Sciences University
3818 SW Sam Jackson Park Road, Mail Code BICC
Portland, OR USA 97201
Email: yackelt@ohsu.edu