Requirements Analysis of
Medical Open Source Software

Wei-Ming Liu
Master of Science in Computer Science
California Lutheran University
Feb. 18, 2009

Abstract. Several previous studies have argued that requirements of open source projects are often asserted by developers rather than derived from users. An explanation for this phenomenon is that developers of open source products are usually users. One may doubt whether this is still the case in software with focused user communities, such as medical software. The paper investigated the features of OpenEMR, a thriving open source electronic medical record product, to confirm that requirements are usually asserted even in medical projects. Many data in the online forums and archives were examined and traced to discover how the requirements were first proposed. The results show evidence to confirm that requirements are usually asserted, and also present several reasons to explain this phenomenon.

1 Introduction

In the last decade, open source software has reached an impressive achievement. Several open source products, such as Apache web server and Sendmail mail transfer agent, dominate product markets today. Many corporations, such as IBM and Sun Microsystems, have invested capital to open source projects. In many product categories, open source software has become a serious competitor to commercial software that dominates markets. For example, Linux is considered a major competitor to Windows. [1]

Open source software is usually of high quality and performance. Several studies have confirmed this. [2][3] However, open source software is usually developed by a group of unorganized volunteer developers who do not demand monetary rewards. Many developers have expressed enjoyment and learning as primary motivators. [4] From traditional views, this seems impossible since software engineering and development processes have long been a challenging and complicated issue. A number of studies regarding open source software development processes have been conducted, and argued that the development processes are radically different from those portrayed in software engineering textbooks. [5][6][7][8] It is important to identify these practices, which will benefit other open source projects, or even traditional software projects. [9]

From studies to date, several open source software processes have been investigated, which include requirements analysis, project management, and integration. [8][9][11] In traditional software engineering, requirements analysis is

important to the success of a software project. [10] A failure in requirements analysis may lead to extra cost, thus dooming the project. On the other hand, open source software is usually developed, at least partially if not wholly, by volunteer programmers who contribute their own efforts and resources without charge. This reduces the costs of software projects and makes possible the practices of other requirements acquisition processes.

A number of studies have shown that requirements in open source software are usually asserted by developers, through their personal experience or knowledge of user needs, rather than derived from users as in the requirement processes of traditional software projects. [8][11] Open source software requirements are often in the form of threaded messages or discussions. [8] Sometimes requirements are born from discussions where the original intentions were not requirement analysis. [8] An explanation for this phenomenon is that most developers of open source software are also users, who develop the products for their personal needs. [4][12]

There is doubt whether this requirement process can be applied to all open source projects. The previous studies were conducted on ubiquitous products, such as operating systems, web browsers, and computer games. The developers of these products are expected to be users. However, can this process be applied to products with focused users, such as medical, banking, or military applications? Are the developers of these products also users? If no, how could this process work? Or is there any other process used in these projects?

This paper presents results of a research attempting to answer these questions. It argues that requirements are usually asserted by developers even in medical open source projects. The research investigated OpenEMR project, which is an open source electronic medical record application. [13][14] The research corroborates its argument by discovering how often features of the product are asserted by developers, derived from users, or acquired from other sources.

2   Background

The OpenEMR project was originally developed by Synitech Inc. and version 1.0 was released in June of 2001 as MP Pro (MedicalPractice Professional). It was then redeveloped to comply with HIPAA standards (The Health Insurance Portability and Accountability Act). In June of 2002, it was reintroduced as OpenEMR version 1.3, which was then registered under GNU General Public License (GPL) in August of 2002. The code base was moved to SourceForge.net in March of 2005. Since its release, OpenEMR has undergone a lot of changes and had many features added. [15]

Today, OpenEMR is one of the most popular and active medical open source projects. The product has been downloaded for more than 39,000 times and translated into several languages. It was ranked by SourceForge.Net as 1,260 out of more than 230,000 projects, which include software in various applications. [16] Few medical projects have made this achievement. What leads to its popularity is probably the business model. OpenEMR is not a pure open source project where

volunteer developers scratch their own itch as Eric Raymond mentioned in "The Cathedral and the Bazaar". Several commercial firms provide service for OpenEMR, and some core developers are paid participants from these firms.

3 Method

The research steps are based on the method used in a previous study of John Noll, "Requirements Acquisition in Open Source Development: Firefox 2.0", in which the features of Firefox 2.0 were investigated to discover the sources of its requirements. Since the research questions of these two studies are similar, the research method of the former can be adopted by the latter. However, the research method has been modified because the nature of medical open source projects is different from that of the Firefox project.

The method includes the following steps:

1. Identify the new features of OpenEMR between version 2.8.0 and 2.9.0.
2. Exclude the non-specific features.
3. Examine online forums, archives, log files, or databases, to find out when the features were first proposed, what roles the proposers played, and how the proposers got the ideas.
4. Categorize the requirements as asserted by developers, derived from users, or acquired from any other sources.

The release notes of OpenEMR between versions 2.8.0 and 2.9.0 list more than one hundred new features. [17][18][19][20][21] However, several statements are not specific enough to be identified as features implemented by a single contribution, such as "Many other bug fixes and minor improvements", "Performance optimizations", or "Improved support for units in services and products". These non-specific features were excluded from the list. Then, each remaining feature was investigated to find out the sources. Finally, the sources of the following features were found:

1. Claims generation integrated into OpenEMR – no more FreeB.
2. Active Directory and LDAP support.
3. Patient pictures may be displayed in the demographics summary.
4. New customizable printable super-bill.
5. SMS Reminder feature.
6. UTF8 encoding support.
7. Fixes for security vulnerabilities.
8. New optional frame-based workflow.
9. Added PHP5 compatibility.
10. Added experimental X12 835 remittance processing for primary claims.
11. Added second date to the patient encounter form for date of onset or hospital admission.

12. Support for deleting immunizations.
13. Option to use date of service as invoice date.

4   Results

The sources of requirements are summarized in Table 1. The first column lists all the features where sources were found. The second column shows what roles the proposers played in the context. Their identities were discovered by signatures or through the messages they left. The third column shows how the requirements were proposed. There are three categories of requirement sources:

1. Asserted by developers from knowledge of user needs.
2. Asserted by developers from personal experience as users.
3. Derived from users who are not developers, by submitting bug reports, proposals or requests.

Most of the requirements were asserted by developers (eleven out of thirteen), either from knowledge of user needs, or from personal experience as a user. The requirements asserted by developers are not radically distinct from those derived from users in the sense that developers are usually users. This phenomenon is also true in medical open source projects, since several requirements are asserted by developers from personal experience as users. Since the sample size of the study is small, a further survey is needed to determine what percent of the medical open source developers are user-developers. However, one can still conclude from the results that this phenomenon is, at least, not uncommon.

Some of these user-developers are medical specialists. Unlike other specialties, programming skill is relatively easier to pick up. Although it takes a lot of training to become a software professional, one can learn programming simply by self-study because there is no need of fundamental knowledge, such as mathematics, which may take more effort to be good at. Besides, as shown in several studies, programming activity is regarded as enjoyment for many. [4] Some medical specialists may have this attribute. Therefore, it is reasonable that many medical specialists have voluntarily participated in open source projects, which in turn benefit themselves.

Along with medical specialists, a number of IT professionals working in medical-related organizations have participated in OpenEMR project. Their jobs are to provide software-related service to medical specialists, such as maintaining medical databases or solving problems of software. One may doubt if they can be regarded as users since they do not need any information stored in the database. However, a user of a software product is not necessarily a person who needs to use the data. For example, a programmer who uses a DBMS to create applications for his customers is a user of the DBMS, but may not need the data of customers. Sometimes medical specialists are not the people who directly manipulate software. [26] In this sense,

Table 1: The sources of new features in OpenEMR

Feature	Role of proposer	Category
Claims generation integrated into OpenEMR. No more FreeB	Developers from supporting firms [22][23]	Asserted from knowledge of user needs
Active Directory and LDAP support	IT professionals in medical organizations [24]	Asserted from personal experience as users
Patient pictures may be displayed in demographics	Medical specialists [25]	Asserted from personal experience as users
New customizable printable super-bill	IT professionals in medical-related organizations [26]	Asserted from personal experience as users
SMS Reminder feature	Developers from supporting firms [27]	Asserted from knowledge of user needs
UTF8 encoding support	-- * [28]	Derived from users by submitting a proposal
Fixes for security vulnerabilities	Developers from supporting firms [29]	Asserted from knowledge of user needs
New optional frame-based layout for improved workflow	Developers from supporting firms [30]	Asserted from knowledge of user needs
Added PHP5 compatibility	Developers from supporting firms [31]	Asserted from knowledge of user needs
Added X12 835 for primary claims	IT professionals in medical organizations [32]	Asserted from knowledge of user needs
Added second date for onset or hospital admission	Developers from supporting firms [33]	Asserted from knowledge of user needs
Support for deleting immunizations	Medical Specialists [34]	Derived from users by submitting a proposal
Option to use date of service as invoice date	Developers from supporting firms [35]	Asserted from knowledge of user needs

* The proposer can only be identified as a login account

the IT professionals can also be regarded as users. This user group is often ignored when it comes to the software products with focused user communities, since one may tend to assume that all users of these products are those who have related expertise, in this case, medical specialists. [36] Compared with medical specialists, most of these IT professionals have better programming skill and software knowledge. They are capable of working as major contributors or core developers. They also work as a source of requirements, which are asserted through their personal experience or knowledge of user needs.

Paid developers from commercial firms that support medical open source projects and provide software-related service are another important source of requirements. They acquire knowledge of user needs from their customers, who are usually medical institutions. It is possible that some traditional requirements analysis or validation processes have been done in these firms before the requirements were asserted on the project forums or mailing lists. However, these validation processes are not as necessary as in closed source projects because the extra costs caused by excessive requirements are less for the supporting firms. In traditional projects, costs increase when requirements are assigned to developers, but here many requirements are simply proposed without being assigned.

Commercial firms are important sources of requirements in the project under study. This phenomenon is common in medical open source projects. [37] In fact, medical institutions that use open source software without supply may encounter some difficulties. First, open source software is often under licenses that include disclaimer of warranty. [38] It is, therefore, risky to use open source software in life-critical applications. Furthermore, unlike users of ubiquitous tools, who can easily find solutions for their software problems online, users of medical software rely more on consulting service. These problems can be reduced with the supply of commercial firms. Thus, a successful open source project is expected to be supported by several firms, which in turn work as sources of requirements.

The goal of the research is to determine how requirements are acquired in medical open source projects. Ideally, the features under study should be randomly sampled from the population, because random sampling reduces bias. [39] In practice, there are some limitations in the research. Sometimes information in open source projects is only partially available. A part of communication among developers is conducted in closed media such as email, thus not available for investigation. As a result, the research only shows the features where the sources are able to be found, and simply ignores others. Despite this bias in sampling, the research provides a lot of information and leads to a significant conclusion.

An interesting issue was found while this research was being conducted. Sometimes the same requirements were asserted by developers more than once. They were ignored after first proposed, but implemented soon after secondly proposed, probably by a significant figure, in an urgent request, or with a specific statement. [27][40] Obviously, there are some factors that determine whether a requirement is to be taken seriously or simply ignored. The understanding of these factors will benefit most open source projects, and thus worth future research.

5 Conclusion

Like other well-known open source projects, requirements of medical products are often asserted by developers. This phenomenon can be explained by the fact that many developers are also users, which may be surprising to many. User-developers include medical specialists who are good at programming, and the IT professionals who work in medical-related organizations. Besides these user-developers, paid participants from commercial firms are an important source of requirements. They acquire knowledge of user needs while receiving feedback from their customers.

On the other hand, the research results imply that it should be the same case in other open source projects with focused user community, such as banking or military applications. What happened in medical projects is likely to have happened in these projects. However, further studies are still needed to find out the answer because there may be some unexpected factors that lead to different results.

6 References

[1] Eric S. Raymond. Halloween Documents. October 1998. A web page of Eric S. Raymond. Accessed February 16, 2009.
http://catb.org/~esr/halloween/index.html

[2] Netcraft. Most requested sites over the last 30 days. February 15, 2009. A web page of a study conducted by Netcraft shows that most of the top 50 most requested sites with long uptimes ran Apache. Accessed February 16, 2009.
http://uptime.netcraft.com/up/today/requested.html

[3] Eric S. Raymond. The Cathedral and the Bazaar, section Release Early, Release Often. May 27, 1997. A web page of Eric S. Raymond. Accessed February 16, 2009. http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/

[4] Karim R. Lakhani and Robert G. Wolf. *Perspectives on Free and Open Source Software*, chapter Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. 2005. MIT Press.

[5] Audris Mockus, Roy T. Fielding, and James D. Herbsleb. *Perspectives on Free and Open Source Software*, chapter Two Case Studies of Open Source Software Development: Apache and Mozilla. 2005. MIT Press.

[6] Niels Jørgensen. *Perspectives on Free and Open Source Software*, chapter Incremental and Decentralized Integration in FreeBSD. 2005. MIT Press.

[7] Walt Scacchi, Joseph Feller, Brian Fitzgerald, Scott Hissam, and Karim Lakhani. Guest Editorial Understanding Free/Open Source Software Development Processes. Software Process Improvement and Practice. 2006. Published online in Wiley InterScience (www.interscience.wiley.com)

[8] Walt Scacchi. Free and Open Source Development Practices in the Game Community. January/February 2004 IEEE software. Published by the IEEE Computer Society.

[9] Daniel M. German. *Perspectives on Free and Open Source Software*, chapter Software Engineering Practices in the GNOME project, third paragraph. 2005. MIT Press.

[10] Leonard Tripp et al. Guide to the Software Engineering Body of Knowledge. Chapter 2: Software Requirements. 2004. Published by the IEEE Computer Society. Accessed February 16, 2009. http://www.swebok.org/index.html

[11] John Noll. Requirements Acquisition in Open Source Software Development: FireFox 2.0. July 11, 2008. IFIP International Federation for Information Processing. Volume 275/2008. Published by Springer Boston.

[12] Eric S. Raymond. The Cathedral and the Bazaar, section The Mail Must Get Through. May 27, 1997. A web page of Eric S. Raymond. Accessed February 16, 2009. http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/

[13] OpenEMR. March, 2005. An online codebase of OpenEMR. Accessed February 16, 2009. http://sourceforge.net/projects/openemr/

[14] OpenEMR. June, 2005. A web page of Open Source Medical Software. Accessed February 16, 2009. http://www.oemr.org/

[15] Sunset Systems. 2005. A web page of Sunset Systems. Accessed February 16, 2009. http://www.sunsetsystems.com/

[16] What is SourceForge.net. February, 2009. A web page of SourceForge.net. Accessed February 16, 2009. http://apps.sourceforge.net/trac/sitedocs/wiki/What%20is%20SourceForge.net?

[17] Rod Roark. OpenEMR 2.9.0 Released. August 7, 2008. News of OpenEMR project. Accessed February 16, 2009. http://sourceforge.net/forum/forum.php?forum_id=854538

[18] Rod Roark. OpenEMR 2.8.3 and FreeB 0.13 Released. August 22, 2007. News of OpenEMR project. Accessed February 16, 2009. http://sourceforge.net/forum/forum.php?forum_id=728212

[19] Rod Roark. OpenEMR 2.8.2 Release Note. January 14, 2007. Release notes of OpenEMR project. Accessed February 16, 2009. http://sourceforge.net/project/shownotes.php?group_id=60081&release_id=478454

[20] Rod Roark. RE: OpenEMR 2.8.1 January 17, 2006. A message on OpenEMR forums. Accessed February 16, 2009. http://sourceforge.net/forum/message.php?msg_id=3527902

[21] Rod Roark OpenEMR 2.8.0 Released. November 11, 2005. A message on Linux Medical News forums. Accessed February 16, 2009. http://linuxmednews.com/1131747101/index_html

[22] Rod Roark. [1375545] Simplify Billing. December 17, 2005. A message on OpenEMR feature requests tracker. Accessed February 17, 2009. http://sourceforge.net/tracker/index.php?func=detail&aid=1375545&group_id=60081&atid=493004

[23] yes2EMR. FreeB installation. July 17, 2005. A message on OpenEMR forums. Accessed February 17, 2009. http://sourceforge.net/forum/message.php?msg_id=3252225

[24] Jason Morrill. Authentication info – ldap. November 7, 2007. A message on OpenEMR forums. Accessed February 17, 2009.
http://sourceforge.net/forum/message.php?msg_id=4611706

[25] Matthew Roller. RE: OpenEMR Wish list, item 18. March 29, 2005. A message on OpenEMR forums. Accessed February 17, 2009.
http://sourceforge.net/forum/message.php?msg_id=3071404

[26] Jake Messinger. Printable blank Encounter form. September 6, 2007. A message on OpenEMR forums. Accessed February 17, 2009.
http://sourceforge.net/forum/message.php?msg_id=4505548

[27] David Kennerson. OpenEMR and the outside world. September 19, 2007. A message on OpenEMR forums. Accessed February 17, 2009.
http://sourceforge.net/forum/message.php?msg_id=4525856

[28] vanas. OpenEMR and UTF-8. March 18, 2007. A message on OpenEMR forums. Accessed February 17, 2009.
http://sourceforge.net/forum/message.php?msg_id=4212705

[29] wpennington. Logging of Actions. – Inadequate Logging. March 22, 2005. A message on OpenEMR forums. Accessed February 17, 2009.
http://sourceforge.net/forum/message.php?msg_id=3061437

[30] Rod Roark. Proposed layout changes November 28, 2006. A message on OpenEMR forums. Accessed February 17, 2009.
http://sourceforge.net/forum/message.php?msg_id=4033139

[31] Rod Roark. MySQL 5 and PHP 5 –anyone tri. March 23, 2006. A message on OpenEMR forums. Accessed February 17, 2009.
http://sourceforge.net/forum/message.php?msg_id=3647136

[32] Balu Raman. EOBs 835's. June 16, 2006. A message on OpenEMR forums. Accessed February 17, 2009.
http://sourceforge.net/forum/message.php?msg_id=3783191

[33] Rod Roark. RE: X12 claims invalid for hospital inpatient. March 19, 2005. A message on OpenEMR forums. Accessed February 17, 2009.
http://sourceforge.net/forum/message.php?msg_id=3055870

[34] Juan. RE: Added revised immunization and shot recor. June 5, 2005. A message on OpenEMR forums. Accessed February 17, 2009.
http://sourceforge.net/forum/message.php?msg_id=3186008

[35] Andres Paglayan. sql-ledger posting dates. December 12, 2005. A message on OpenEMR forums. Accessed February 17, 2009.
http://sourceforge.net/forum/message.php?msg_id=3471366

[36] Robert L. Glass. *Perspective on Free and Open Source Software*, chapter Standing in Front of the Open Source Steamroller, paragraph *Political/Sociological Model*. 2005. MIT Press.

[37] care2x, Hospital Information System, Support, Consultancies. 2008. A web page of care2x project. Accessed February 17, 2009. http://www.care2x.org/

[38] Richard Stallman et al. GNU General Public License Version 3, section Disclaimer of Warranty. June 29, 2007. A web page of the Free Software Foundation. Accessed February 16, 2009. http://www.gnu.org/licenses/gpl.html

[39] Valerie J. Easton & John H. McColl. Statistics Glossary v1.1, chapter Sampling. September 1997. Published by STEPS. Accessed February 17, 2009. http://www.stats.gla.ac.uk/steps/glossary/sampling.html#randsamp

[40] Matthew Roller. [1213156] Adding options for appointment reminders. June 2, 2005. A message on OpenEMR feature requests tracker. Accessed February 17, 2009. http://sourceforge.net/tracker/index.php?func=detail&aid=1213156&group_id= 60081&atid=493004