SLOUCHING TOWARD OPEN INNOVATION: FREE AND OPEN SOURCE SOFTWARE (FOSS) FOR ELECTRONIC HEALTH INFORMATION

Greg R. Vetter[†]

TABLE OF CONTENTS

[†] Associate Professor of Law, University of Houston Law Center; Co-Director, Institute for Intellectual Property and Information Law; biography available at: www.law.uh.edu/faculty/gvetter. My background includes a Master's degree in Computer Science and nine years full-time work experience in the software industry. My thanks to Law Center students Daniel R. Peterson, Katherine A. Franco, Bo Tang and Domingo Llagostera for excellent research assistance. Mr. Peterson's contributions to the research and this paper are particularly noteworthy. For helpful comments and discussion, I thank Rich Saver, Marcilynn Burke, Sharona Hoffman, Andrea Matwyshyn, Andy Podgurski, Fred Trotter, and participants at the *Interdisciplinary Conference on: Beyond the Ideologies of Open-source and Proprietary Models of Innovation* held in April 2008 by the CENTER FOR RESEARCH ON INNOVATION, ENTREPRENEURSHIP & LAW at the WASHINGTON UNIVERSITY SCHOOL OF LAW. This article is forthcoming in a symposium issue from the conference in the Washington University Journal of Law and Policy.

ABSTRACT

The potential for Free and Open Source Software (FOSS) to enable open innovation in a particular software market depends on the characteristics of that market. From this premise, using a case study approach, this Article argues that some software markets have characteristics that inherently disfavor initiating or expanding the use of FOSS. The case study involves software to manage health information for hospitals or physician groups in the form of the electronic medical record, or EMR. Proprietary software venders supply most of the products for this software market. Recently, the U.S. government undertook experimental steps to promote a FOSS package for EMR, raising the question as to whether the EMR software market is amenable to FOSS. This Article describes various factors that might signal a FOSS-disfavoring market, including low technical aptitude among users, differences among users in their work flow and software interface needs, users with dispassionate computing agendas, and entrenched proprietary competitors in an area supporting minimal complementary goods or services. FOSS, however, might be able to overcome these impedances in a particular software market if its unique motivational mix is strong enough. This Article describes potential facilitators to support this possibility. One such facilitator, specifically for the EMR market, but perhaps generally for other markets, may be safe harbors for FOSS development within any relevant anti-collaboration and anti-tinkering laws. Licensing facilitators include emphasizing approaches such as dual licensing or promoting FOSS contributions by contractors engaged by users. This Article concludes by mentioning potential non-licensing facilitators to augment the FOSS motivational mix for markets that might disfavor it.

I. INTRODUCTION

Gartner, one of the most respected market research firms for information technology, recently called open source software the "biggest disruptor the software industry has ever seen and postulated it will eventually result in cheaper software and new business models."[1] The degree to which this prediction materializes depends on many influences, one of which is the subject of this Article. I argue that some software markets are more favorable for open source approaches than others. Using a case study of one particular software market, this Article develops a tentative framework of factors characterizing markets likely to disfavor contemporary approaches in free and open source software (FOSS).[2]

A software market is intimately intertwined with the licensing techniques employed in the market. This suggests that demand side responses may change based on new licensing techniques - an effect that is already a feature of the FOSS movement.[3] If identifiable characteristics describe FOSS-disfavoring markets, this perspective may lead to the development of new FOSS techniques to enable open innovation in those markets. The last part of this Article outlines directions to facilitate this process.

---

[1] *See* Peter Galli, *Open Source Is the Big Disruptor*, available at http://www.eweek.com/article2/0,1895,2186932,00.asp.

[2] The FOSS movement has spawned a variety of scholarship in the legal academy. *See generally* Yochai Benkler, *Coase's Penguin, or, Linux and The Nature of the Firm*, 112 Yale L.J. 369 (2002); David McGowan, *Legal Implications of Open-Source Software*, 2001 U. Ill. L. Rev. 241, 268, 274 (2001) [hereinafter McGowan, *Legal Implications*] (noting the volunteerism underlying open source software development); Greg R. Vetter, *The Collaborative Integrity of Open Source Software*, 2004 UTAH L. REV. 563 [hereinafter Vetter, *Collaborative Integrity*]. FOSS scholarship also includes an increasing number of books. For an early classic, see Open Sources: Voices from the Open Source Revolution (Chris DiBona et al. eds., 1999) [hereinafter OPEN SOURCES]. *See also* OPEN SOURCES 2.0: THE CONTINUING EVOLUTION (Chris DiBona et al. eds., 2006) [hereinafter OPEN SOURCES 2.0]; STEVEN WEBER, THE SUCCESS OF OPEN SOURCE (2004). A number of practicing lawyers have authored books on FOSS licensing, and these provide helpful background as well. *See e.g.* LAWRENCE ROSEN, OPEN SOURCE LICENSING: SOFTWARE FREEDOM AND INTELLECTUAL PROPERTY LAW 103-06, 126-28, 133-136 (2005) (discussing the way in which FOSS licensing developed and how it works).

[3] *See* Dirk Riehle, *The Economic Motivation of Open Source Software: Stakeholder Perspectives*, 40 No. 4 IEEE COMPUTER 25, 27 (2007), available at http://www.riehle.org/computer-science/research/2007/computer-2007.pdf (last visited June 24, 2008) (discussing how open source software licensing has effected market demand because the change from closed source software to open source software "reduces the lower price limit for possible deals and puts a new set of more price-sensitive customers within reach").

One example of a software model that has influenced licensing practices is an application service provider (ASP). An ASP typically provides a licensee with access and use to software over a network hosted by the provider. Through this ASP licensing scheme, customers can avoid (1) one time license payments, (2) hardware investments; (3) risk of outdated software; and (4) risks of being financially bound to a vendor. *See* Michael P. Widmer, *Application Service Providing, Copyright, and Licensing*, 25 J. MARSHALL J. COMPUTER & INFO. L. 79, 83 (2007). ASPs can also "aggregate software licensing fees with other services, which may effectively lower software costs." H. Lamar Curtis III & Andrew Ramzel, *Snake Up Your Firm's Productivity: These ASP's Offer Efficient Tech Solutions*, 19 No. 6 Legal Mgmt. 22, 24 (2000).

The FOSS licensing movement uses several techniques to emphasize source code transparency and, for many licenses, require subsequent development to occur under the same or a similar FOSS license. Sometimes the licenses include anti-royalty provisions for ongoing software use. And sometimes they require extension of the FOSS terms to closely intermixed software.[4] These licensing foundations influence the software development approach both organizationally and technologically. Thus, when a software market, such as that for operating systems, has a FOSS entrant, the strategic considerations and posture of the FOSS entrant are different compared to proprietary-licensed software products, which typically keep source code as a trade secret.

A software market, beyond the classic attributes one might use to define any market, will comprise some or all of: preexisting technologies; evolving hardware and software platforms; user requirements; business process demands; interoperability and availability needs; standards entanglement; and licensing methods.[5] The technological complexity involving each of these features will depend on the particular software market. The interactions among these features are significant. Moreover, all these structural features and their interrelationships evolve at breathtaking rates in the computing arts. Part II below will describe these features further.

Even if FOSS is the "biggest disruptor in the software industry," altering its efficacy in a market depends on understanding its unconventional motivational mix. While more research is needed in this area, much has been done to describe various motivational elements behind FOSS, such as reputation,[6] career concerns,[7] gift

---

[4] There are various issues of doctrine that are not well-settled with FOSS licensing. *See* McGowan, *Legal Implications*, *supra* note 2, at 289-302 (2001) (discussing doctrinal issues related to a variety of issues, including assent, privity, term, termination, and assignment); *see generally* Vetter, *Collaborative Integrity*, *supra* note 2, at 623-648 (discussing the influences among software licensing terms and software development). The primary basis of a FOSS license is typically copyright, although some FOSS licenses include provisions relating to patent law. Often FOSS licenses are classified into types. One type, attribution-only licenses (sometimes called BSD-style licenses) generally allow any use of the software, even in proprietary products without source code, so long as attribution is given. Another type, "copyleft licenses," has several requirements: (1) royalty-free software use; (2) available with source code; (3) distributable in modified or unmodified form; (4) with recipient users and redistributors granting a copyright license to other recipients for any added development; and (5) with all these conditions applying to future generations of the software upon redistribution with or without modification, including modifications that intermix other software.. Finally, unless a license is named, this Article does not intend to single out any specific license; FOSS licensing is taken as a system.

[5] Several of these characteristics combine in a software market to establish common patterns for the life cycle timing of versions or iterations of the software, and to establish the typical extent to which vendor or third-party support is necessary for installation and customization of the software.

[6] Eric S. Raymond discusses several aspects of reputation-enhancing behavior, contrasting reputational gains for the prospects of economic reward with reputation gains for social status within the open-source "hacker" gift culture. Eric S. Raymond, *Homesteading the Noosphere*, § 2 (ver. 3.0, 2000), *available at* http://www.catb.org/~esr/writings/homesteading/homesteading/ (last visited June 19, 2008).

economies,[8] and complementary economics.[9] The pronouncements arising from this research, however, are complicated by continuous change within information technology. This evolution is no longer exogenous to FOSS. Nevertheless, the motivation to supply FOSS to a particular software market helps determine whether the market is or will remain FOSS-disfavoring, and whether new approaches might change that inclination.

The unique nature of FOSS development complicates the motivations compared to the profit motive of a typical proprietary software supplier.[10] Proprietary software has a supplying company and paying users. In contrast, FOSS has a community. Some community members are mere users, some are users who additionally contribute to testing and/or development, and some are primary developers with great influence over the technological direction of the software. The users may or may not pay for the software. Companies are sometimes initiators and coordinators of FOSS products, but do so under different monetizing business models from those typically used in proprietary software products. While user feedback is important for successful proprietary software, user involvement in the community has heightened importance for FOSS. Thus, the motivational mix in FOSS includes the degree to which the users in the software market prefer to engage in the FOSS experience.

FOSS originated from highly technological software markets. There are two distinct ideologies within the greater movement.[11] One emphasizes the label "free software," representing self-determination and social solidarity with computing. The other emphasizes the label "open source" as a better software development approach arising from transparent source code. Users may identify with both, one, or neither of these strands within the movement. Users in many markets see computing as an instrumental asset toward greater organizational productivity and effectiveness. This utilitarian outlook may leave little room for the ideological drivers within FOSS.

If some characteristic features of a software market hint that it is FOSS-disfavoring, and if FOSS motivation for that market is estimable in some way, this

---

[7] *See* Josh Learner & Jean Tirole, *The Simple Economic of Open Source* 14-15 (HBS Finance Working Paper No. 00-059, 2000), available at http://ssn.com/abstract=224008 (discussing the "career concern incentive" that many open source programmers value, such as future job offers, shares in open-source software companies, or "access to the venture capital market").

[8] *See* Raymond, *supra* note 6, § 6 (emphasizing that social status is governed by what one gives away as opposed to what one controls).

[9] *See* Joel West, *How Open is Open Enough? Melding Proprietary and Open Source Platform Strategies*, 32 RES. POL'Y 1259, 1259-66 (2003) (discussing how the emergence of standardized platforms which allow for substitution of "complementary assets" has been a driving force for the evolution of the computer industry).

[10] Sandeep Krishnamurthy, *An Analysis of Open Source Business Models*, *in* OPEN SOURCES 2.0, *supra* note 2, at 280–82.

[11] *See* Greg R. Vetter, *Exit & Voice in Free & Open Source Software Licensing: Moderating the Rein over Software Users*, 85 OREGON L. REV. 183, 205 (2006) (noting that the line between the two ideologies is not a bright line) [hereinafter Vetter, *Exit & Voice in FOSS*].

provides a static sense of the potential for FOSS penetration into that market. In computing, however, rapid change is guaranteed. The dynamic picture may tumble forward with surprise turns. Many interests, including governments, investors, and companies of all types, are betting that the tumbling evolution of information technology includes growth in FOSS. Such growth is not assured - even if it is generally anticipated.

The context in which this Article will examine these issues is a business-to-business software market within health care where the U.S. government has recently supported efforts to promote a FOSS product called WorldVistA.[12] This is a rare example. The U.S. government has been comparatively passive with FOSS in comparison to many other countries that explicitly mandate or favor it.[13] WorldVistA is one of a small number of FOSS products with a presence in the market for storing and managing health information electronically for use by health care providers.[14] Providers include doctors and hospitals, and thus their arrangements vary from sole practitioners to large multi-site organizations. Various acronyms label the market, but I will use

---

[12] Thomas Goetz, *Physician, Upgrade Thyself*, N.Y. TIMES, May 30, 2007 ("The effort to promote WorldVistA is supported by a grant from the Centers for Medicare and Medicaid Services, the [federal] agency that sets the prices for Medicare and Medicaid payments."); Forrester Research, *Open Source Software: A Primer for Health Care Leaders*, 8-10 (Mar. 2006), http://www.chcf.org/documents/healthit/OpenSourcePrimer.pdf. *See generally* Sharona Hoffman and Andy Podgurski, *Finding a Cure: The Case for Regulation and Oversight of Electronic Health Record Systems*, HARVARD JOURNAL OF LAW AND TECHNOLOGY - (forthcoming), at 6-8, 32-33, *available at* http://ssrn.com/abstract=1122426 (describing EMR systems and WorldVistA). [hereinafter Hoffman & Podgurski, *Finding a Cure*].

[13] Jyh-An Lee, *New Perspectives on Public Goods Production: Policy Implications of Open Source Software*, 9 VAND. J. ENT. & TECH. L. 45, 55-64 (2006); David S. Evans, *Politics and Programming: Government Preferences for Promoting Open Source Software*, *in* GOVERNMENT POLICY TOWARD OPEN SOURCE SOFTWARE 34, 34-35 (Robert W. Hahn ed., 2002), http://aei-brookings.org/admin/pdffiles/phpJ6.pdf (last visited June 15, 2008) [hereinafter Evans, *Preferring OSS*].

[14] E-mail from Fred Trotter to author (Apr. 11, 2008, 14:52 CST) (on file with author) [hereinafter Trotter email]. Other FOSS EMR software products include: FreeMed, http://www.freemed.org; GnuMed, http://wiki.gnumed.de/bin/view/Gnumed; OpenMRS, http://openmrs.org/wiki/OpenMRS; and ClearHealth, http://www.clear-health.com. *Id.* Another vendor is DSS, http://www.DSSinc.com, whose products are based on the same U.S. Department of Veterans Affairs public domain software that underlies WorldVistA. The American Academy of Family Practice also keeps a list of FOSS EMR software. *See* AAFP, Center for Health Information Technology, Open Source Medical Projects, http://www.centerforhit.org/x337.xml; Sujansky & Associates, *Open-Source EHR Systems for Ambulatory Care: A Market Assessment*, Executive Summary at 1-3 (California HealthCare Foundation, Jan. 2008), http://www.chcf.org/topics/view.cfm?itemid=133551.

Electronic Medical Record, or EMR.[15] There are hundreds of EMR software suppliers licensing proprietary software. Given the variance among health care providers in size, type, medical specialty, and jurisdiction, the EMR market has multiple submarkets. At present, however, it is primarily a non-retail market.[16] It exists within a highly regulated industry, and these regulatory forces influence the software requirements.[17]

Building on Part II's software market characterization, Part III will describe the EMR market specifically and begin to develop the factors that indicate some likelihood of a FOSS-disfavoring market. WorldVistA has virtually no penetration in the physician office segment of the EMR market, and only a few nascent installations in the institutional setting. This Article's claim, however, is not that this product's minimal penetration at this time demonstrates that the market is FOSS-disfavoring. The claim is rather that the structural characteristics may represent a perfect storm of factors for a FOSS-disfavoring market even while new proprietary software installations continue in that market. Against this storm, the question arises whether some FOSS motivational mix is sufficiently potent to overcome the resistance arising from the structural characteristics in the EMR market. The related question is whether new FOSS licensing efforts would facilitate the process.

Any new approaches should consider FOSS motivations, which this Article overviews in Part IV below. From this, Part VI presents some tentative factors characterizing a FOSS-disfavoring market that generalize from the EMR software market. Beyond the commonly noted suppositions that FOSS is less successful in markets where users have less technical acumen and where there are minimal complementary effects for other products or services, the framework includes the degree to which software-supported human workflow differs among users. It also raises user interface issues generally in light of the typical need for business process automation software to govern user permissions and capabilities.

---

[15] Another common acronym is EHR, for Electronic Health Record. Institute of Medicine, Key Capabilities of an Electronic Health Record System 1-2 (2003), http://www.nap.edu/catalog/10781.html [hereinafter EHR Key Capabilities]. Another, older acronym is CPR, for computerized patient record. Joan R. Duke & George H. Bowers, *Scope and Sites of Electronic Health Record Systems*, 89 *in* ASPECTS OF ELECTRONIC HEALTH RECORD SYSTEMS (Harold P. Lehmann et al., eds., Springer, 2d ed. 2006) [hereinafter ASPECTS OF EHR]; MARGRET K. AMATAYAKUL, ELECTRONIC HEALTH RECORDS: A PRACTICAL GUIDE FOR PROFESSIONALS AND ORGANIZATIONS 6 (American Health Information Management Association 2d ed. 2004).

[16] Initiatives by several information technology companies may add a "retail" element to software for electronic health information. *See* Intuit, Quicken Health Care Management Products, http://quicken.intuit.com/healthcare-management (describing Intuit's Quicken Medical Expense Manager software product); Google Health, www.google.com/health; Microsoft HealthVault, http://www.healthvault.com.

[17] *See* Arnold J. Rosoff, *On Being A Physician in the Electronic Age: Peering into the Mist at Point-&-Click Medicine*, 46 ST. LOUIS U. L.J. 111, 119-125 (2002) (discussing regulatory regimes that might bear on software used in providing health care).

Suggested directions to facilitate open innovation in FOSS-disfavoring markets begin with a specific recommendation for the health care industry, where certain anti-collaboration laws might chill FOSS involvement. The recommendation would generalize to other regulated industries with similar collaboration governing mechanisms. After this, Part VII presents other suggestions organized between licensing approaches versus other facilitators. Licensing includes a need to develop stronger licensing traditions around dual licensing and other forms of asymmetric copyleft licensing. Suggestions under the other facilitators' category include emphasizing a recent movement for service markets within FOSS.

The suggestions and tentative framework of factors emphasize this Article's overarching theme: contemporary FOSS approaches to open innovation may not necessarily fit every software market. This Article proceeds from a baseline intuition that the FOSS movement brings beneficial influences to the greater information technology ecosystem. To the extent one embraces this intuition, course-correcting FOSS's application in disfavoring markets will allow its influences to continue to thrive.

II. SOFTWARE MARKETS

There is some degree of fluidity and arbitrariness in describing what falls within a particular software market, or in deciding how to describe the market scope. The purpose behind defining the market influences the delineation. With this in mind, this Part reviews key technological characteristics which could be used to differentiate one class of software applications from another, and thus could be helpful in differentiating different software markets.

This Article does not pause to anchor the technological characteristics to a particular general framework that might be used to define a market. It is self-evident that many such frameworks exist, but my judgment is that the discussion can proceed without choosing a single framework or reviewing the range of available frameworks. Economic definitions for market scope will be important later in this Article's argument, particularly notions of complementary goods and services and market interactions among these. The concepts for non-commoditized markets are also important because most software markets have differentiated products. For example, one company's inventory tracking software product for a dental products distributorship might be very different in features and functionality than the other software product suppliers to that niche market. Customer switching costs are typically high in software markets and the buyer/seller relationship is often a long-term entanglement. These principles apply regardless whether the customer is a retail consumer or a business, although these factors are often explicitly considered in the procurement process by a business.

A software market transaction might also involve services along with purchased or licensed software. For ease of discussion, this Part II, and Part III below, will simply describe all software transactions as purchases, and will characterize software markets in a proprietary software sense, putting aside new perspectives on software markets arising

from the FOSS movement.[18] Suppliers regularly include services with software supplied to business customers allowing cross-subsidization internal to the supplier. Often, as a result, securing new customers, particularly business customers for high-dollar enterprise software, is a strategic activity undertaken by professional salespersons. Retail software products are also sometimes bundled with services, such as: technical support; rights to new versions of the software, or updates for continued product viability, such as antivirus software updates. Bundled services for retail software, whether in use by consumers or businesses, are more straightforward than the complex support and services arrangements associated with enterprise software. Many retail software products are purchased by both consumers and companies, but above certain price levels, and for various types of functionality, businesses are the only customers in the market.

Of course, the competition among proprietary software suppliers occurs with respect to value in relation to price. Value is measured against the desired software features and functionality. Customers evaluate price for original procurement, but business customers may also study the life-cycle cost for software ownership. In typical sales situations in which business customers procure software, the value analysis also incorporates technological requirements influenced by business needs. The business customer, through its information technology department, will go beyond asking whether the software product will run on its computers and determining the initial price. These technological requirements are the subject of Section II.A below, and provide a basis for differentiating software markets.

Finally, in the case of most business customers, software markets often have a "build versus buy" dimension.[19] Even when software or information is not a primary organizational output, many enterprises develop substantial internal information technology capabilities. For these organizations, third-party software products compete for the best value proposition, but these organizations also might compare procured solutions to internal development. The build versus buy decision isn't so stark, however, because, even if a business has no internal software development capability, it can engage a contractor to develop software from scratch or select a software product that is close to the desired feature set and engage the product supplier to customize the product for the

---

[18] Interview by Todd R. Weiss with Eben Moglen, COMPUTERWORLD, Jan. 21, 2008 (discussing the impact on proprietary software companies under "the idea of making a program or other piece of work freely distributable, as opposed to restricting its use via a copyright").

[19] The decision whether to procure software from a vendor or develop it in-house depends on various factors, but a dominant factor is the need for precisely fitting functionality versus its availability, calibrated against the value of a precise fit in light of the failure risks of software development. *See generally* Henry Chesbrough, New Puzzles and New Findings, 17-19 *in* OPEN INNOVATION: RESEARCHING A NEW PARADIGM (Henry Chesbrough, Wim Vanhaverbeke & Joel West eds., Oxford Univ. Press 2006) [hereinafter OPEN INNOVATION]. In health care information technology, the "buy" option seems dominant as most institutional providers and physician offices license EMR software from others rather than creating it from scratch. *See* AMATAYAKUL, *supra* note 15, at 253-56.

customer. Numerous complex considerations inform these latter two avenues, but they are viable alternatives in many cases.

A. *Technological Market Characteristics*

To operate, software needs hardware and other software. Computing technology is layered. Hardware, in the form of processing chips and memory, is the foundation. Layers of software are built on the hardware. The upper layers typically provide the user interface. An example is a user working with a spreadsheet. Computing work travels down from the upper layers. The spreadsheet task spends some time with the processor, which may momentarily switch away from the task many times to work on other tasks before completing.[20] The finished work then travels back up the layers, eventually to show the spreadsheet user a result.

The insight from this model is that not all software runs in all environments. This is especially true for compiled object code, the preferred distribution method for proprietary software. Taking a common example, some software products run on Microsoft's Windows XP operating system, but do not run on any of the GNU/Linux operating system distributions.[21] In either case, the operating system provides numerous layers of software between the hardware and the software product.

Thus, a characteristic for any software market is the platform(s) on which the products in the market operate. The term platform might refer either to hardware or other lower-level layers of necessary software, but most commonly refers to the operating system. For example, two products dominate the software market for household financial organization: Intuit's Quicken product, and Microsoft's Money product. Neither are available natively for GNU/Linux.[22] The "natively" qualifier refers to technology that allows software designed and compiled for one operating system to operate on another operating system. The product runs on the nonnative operating system under emulation software interposed between the product and the nonnative operating system. These dependencies influence purchasing decisions in a software market. Purchasers prefer

---

[20] This account applies a typical model for computer processing. *See* RANDAL E. BRYANT & DAVID R. O'HALLARON, COMPUTER SYSTEMS: A PROGRAMMER'S PERSPECTIVE __–__ (2003).

[21] For example, the Quicken Medical Expense Manager software product only runs on Windows operating systems. *See* System Requirements & FAQs, *supra* note 16.

The GNU/Linux operating system is sometimes referred to as Linux. An operating system, however, is not a single large software work, but is rather an aggregation of many software components. The central component is the kernel, which is properly called Linux. Distributions of a Linux kernel-based operating system include other critical components. Most distributions include a set of essential software tools from the GNU project, a separate open source software effort. Richard Stallman, Linux and the GNU Project, http://www.gnu.org/gnu/linux-and-gnu.html. Thus, some use the name "GNU/Linux" for such a distribution. *Id.* The GNU acronym is a self-referential label meaning "GNU's Not UNIX," with Unix being a predecessor computer operating system. *See* The GNU Operating System, http://www.gnu.org.

[22] *See* Microsoft Money System Requirements, http://www.microsoft.com/money/freetrial_essentials.mspx#systemRequirements.

natively supported applications, but obtaining the desired software functionality sometimes dominates the buying decision and may lead to selection of nonnative software.

Software markets often exhibit two effects arising from the same inertia, known as cumulative functionality development. The first effect is functionality expansion with attendant backward compatibility pressures for new versions. Pejoratively, this is described as feature bloat[23]. The second effect is magnifying user lock-in.

The inertia spawning these effects arises due to the continually dropping cost of hardware and communications bandwidth. The processors and the network can handle much more software each year and still provide improved performance. Thus, cost factors for software development tend to channel software product suppliers into a "kitchen sink" mentality. In other words, over time, it seems beneficial to keep adding capabilities to the product, particularly because the ever-increasing computing power minimizes the need for optimization costs or removal costs.

This functionality expansion is helpful in the sales process. Retail customers shop for functionality, and business customers often extensively evaluate software based on feature strength. The more features, the better, because the customer likes to know that they could use the software in some particular way in the future, even if they don't initially plan to do so. Although pathways and dependencies within the software product determine the degree to which the following is true, adding features is sometimes less costly for the software supplier than removing features. Removing features may break other parts of the product, resulting in costly recoding and retesting that could have been avoided. The difficulty with removing a feature is that even if most users no longer use it, the users that still do will be dissatisfied if they upgrade to the new version. A similar effect is the need for backward compatibility. Not only must a feature remain in the product when a new version is released, it must continue to provide the core benefit even if expanded. In most software markets, backward compatibility for user data and software functionality is an important customer concern. That being said, it isn't always provided. In effect, users want backward compatibility so they can continue their locked-in status. This is not because they prefer that status and the leverage it gives the software supplier, but because it avoids retraining costs.

A major change in a business customer's processes may bring the ultimate juncture for a software user: a product switch. Switching products involves the costs of software evaluation and selection, a new implementation, and resulting user retraining. A common example is when two companies merge or there is a buyout. Assuming each company uses a different third-party accounting package, one software supplier is going to have a larger customer, and one software supplier is going to lose a customer. Another example is a business expansion to offer a new product or service for which the

---

[23] Harry McCracken, *How to Build Better Software: It's Simple*, PC WORLD, Vol. 23 Issue 2, 17 (2005).

company's current software product neither provides nor promises functionality. Retail customers might also switch products, but their switching costs don't ripple across the workflow of an entire organization.

Beyond a software product's need to run on particular platform(s) and respond to platform evolution as it augments its capability over time, it may need attention in the related areas of standards and interoperability.[24] Both areas facilitate beneficial extension of the software's inputs, outputs, or capabilities. Although standards have various purposes in the greater economy, within information technology they primarily serve to facilitate interoperability, enable code reuse, and reduce technologist and user training costs. For this Article, standards will be taken in its broadest sense, including both de jure standards such as XML,[25] and de facto standards, such as Microsoft's Excel product for spreadsheet calculating. Similar to the need for a specific set of features and functionality, software procurement professionals sometimes evaluate competitive products based on the standards supported by the software. For example, a company buyer evaluating inventory tracking software might specify that the package be able to export reports directly into an Excel spreadsheet.

The next technical consideration is availability. In the consumer context, this might mean software that doesn't regularly invoke a need to reboot the computer, or software that is resistant to disabling malware. In the business context, the same meaning might apply as the starting point. For an enterprise, the expectations for uptime and availability of the software can extend to situations where, truly, failure is not an option. These mission-critical information technology systems use specialized redundancy and other high-availability technologies applicable to enterprise computing.

Availability needs are sometimes an element of the user requirements. Among a potentially large number of parameters, user requirements might specify the ease of use necessary for the software, response times for operations, whether any specialized computing devices, such as mobile computers or handheld devices, are necessary, and whether different users can have different capabilities within the system. User interface issues related to ease of use can influence software procurement processes to specify products that run on the most widely used operating systems. Such ubiquitous presence minimizes user training and the prior familiarity generates positive ease of use impressions.

For business customers in a software market, both availability and user requirements link to the process or processes to which the business will apply the software. If the business process is to computerize inventory tracking with low-skilled workers in a factory operating the software, the ease of use will need to be high, as will

---

[24] *See* Greg R. Vetter, *Open Source Licensing & Scattering Opportunism in Software Standards*, 48 B.C. L. REV. 111 (2007) [hereinafter Vetter, *Scattering Opportunism*].

[25] World Wide Web Consortium (W3C), Extensible Markup Language (XML), http://www.w3.org/XML.

availability. If the business process is a specialized statistical database to support a forecasting group at a public utility in preparing long-range capacity plans, the user interface can present complexity, and in a short-term sense the software is not mission-critical. While a business process in the abstract sometimes isn't a technological characteristic for a software market, it drives determination of the required technological characteristics. Sometimes the business process is to insert automation between two automated systems with human linkage, in which case the new process is inherently technological. The vast range of what could be called a business process dissuades reviewing a long string of examples. For any particular process to be automated via computing, user needs and system availability will be part of the calculus.

Most of the technological characteristics of a software market reviewed in this Section are touched upon below in a progressive narrative offered as an example of the "business process automation" software market. A market scope defined as "business process automation" has very broad scope, yet has a core set of characteristics that map well to the market studied later in this Article - electronic medical record software.[26]

B. *Business Process Automation*

Information technology has reconstructed how businesses implement their processes. Companies have computerized virtually every conceivable activity to some degree. The result is gains in productivity, greater reliability and quality for outputs, and long-term cost reductions.[27] Software that enables these results can be grouped into a broad market classification called business process automation software.[28]

This market has institutional buyers and sellers of all sizes, but virtually all of the software products are unavailable at retail. They would be of no use to the typical household consumer. The sellers offer software products sometimes bundled with services or customization of the software. Tiny companies may sell to the largest companies, and the opposite may occur as well. The products enhance productivity through more accurate information handling. They replace human activity with computed results. In some fields, such as manufacturing, the software may direct machinery. In other areas, such as accounting, finance or insurance, the software reduces recordkeeping costs. Labor cost reduction is often part of the value proposition for business process automation software, although the computational assets often induce a

---

[26] Mariel L. Bernstein et al., *Five Constants of Information Technology Adoption in Healthcare*, HOSPITAL TOPICS, Vol. 85 Issue 1, 18-19 (2007).

[27] *See generally* Randy Weston, *Bristol-Meyers CEO demands massive supply chain fix*, COMPUTERWORLD, Nov. 17, 1997, at 47. (describing Bristol-Meyers' effort to re-engineer its global supply chain, resulting in "a more efficient production and distribution process that the company expects will save $150 million per year"). Ken Cottrill, *Winning SIMON Says*, TRAFFIC WORLD, Jul. 27, 1998, at 38. (discussing Shell Inventory Management Order Network, nicknamed SIMON, that "has made inventory management more efficient and helped the company capture new business").

[28] *See* AMATAYAKUL, *supra* note 15, at 112-115.

new labor cost in technologists to support and maintain the computers, network and software.

To illustrate, what follows is a stylized, progressive example for computerized inventory control, one of many business processes subject to automation during the growth of computing. The example begins in the past, at least in the early 1980s, or perhaps earlier. A small manufacturer, LittleBuyer, replaces a manual inventory tracking method with "off-the-shelf" networked computers, and software purchased from BigSeller. The inventory is for quantities of parts used in LittleBuyer's manufacturing process. The inventory information also includes attribute information LittleBuyer collects by testing the parts upon receipt, such as weight measured on an ultra-precise scale. The human-implemented business process of tracking inventory is automated by the software from BigSeller. LittleBuyer can thus allow one of four inventory-tracking employees to retire without replacement.[29]

Assume that LittleBuyer installed the original system before commoditized and standardized barcode technology was available. This factor, along with the physical facility layout and the type of manufacturing, would dictate work flow, perhaps both for humans and the manufacturing process. Later, when low-cost barcode technology arrives with portable, handheld scanners, LittleBuyer can redesign the workflow for productivity gains. The employees no longer have to go to specific computer locations to enter inventory information. The employees may now collect the inventory information using the handheld scanning device as they move around the facility by scanning barcodes placed on the parts in receiving. The scanner's software stores the information for batch transfer to the computers when the barcode is connected to the network linking the inventory control computers.[30] The employees also find the user interface on the handheld scanner much more convenient than the hierarchical screens and menus on the inventory tracking computers. This now allows LittleBuyer to reduce the employees handling inventory tracking from three to one.

Assume further that BigSeller develops a new software component that LittleBuyer purchases and adds to the original software. The new component has smart algorithms that scan the database of current and historical inventory information and

---

[29] *See generally* Margaret Sheridan & Janice Matsumoto, *No Pain All Gain*, RESTAURANTS & INSTITUTIONS, Feb. 15, 1999, at 57. (after installing a computerized inventory control system, "labor and payroll are reduced by an estimated $25,000 yearly").

[30] Barcode technology such as described here is commonly used in retail and manufacturing settings. *See generally* Anonymous, *C-Store Chain Rings Up Savings Using With Handheld Stock Ordering System*, DAIRY FOODS, Jan. 2005, at 88.

estimates the optimal quantity of parts to order for a given time period.[31] This allows the parts procurement employee group to be reduced from two to one. In total, since the original installation of the system, LittleBuyer has reduced labor costs by four full-time-equivalents. However, it eventually has to hire a full-time computer technologist to support the inventory control computers and software.[32]

Next, BigSeller issues a new major version of its software. Since LittleBuyer has always kept a software maintenance contract with BigSeller, it receives the new version without cost. Over the years, LittleBuyer's technologist became adept at installing new versions of BigSeller's software. The software users prefer the new version because it replaces the hierarchical screens and menus with a windowed interface that operates similar to all the other computers that have appeared in their lives. This upgrade, however, offers a new module that wasn't available before and isn't included under the maintenance contract. Regardless, LittleBuyer decides to buy it. The new module is an Application Programming Interface, or API. It provides several hundred commands that allows LittleBuyer's technologist to write custom software exchanging data with the inventory software, and commanding that software to perform tasks.[33] In other words, via the API, LittleBuyer can add more automation in addition to the automation inherent in the original BigSeller software. BigSeller never discloses its source code, so it has to provide the API for customers who wanted to do things differently from the pathways available in the regular product.

For some LittleBuyer customers, product weight must be minimized. LittleBuyer uses the API to write code for special reports and handling about parts in inventory meeting the low weight requirements. Over time, these customers are increasingly satisfied because LittleBuyer's parts are more frequently within tolerance for weight. As a result, LittleBuyer's sales in this segment expands dramatically because most of the weight-conscious customers use LittleBuyer's parts in satellites, which was a growing market around the time LittleBuyer purchased the API.

Finally, the Internet arises and LittleBuyer writes code with the API to automatically send procurement requests to its parts suppliers through the Internet. The procurement job is now obsolete, so LittleBuyer lets that employee retire without

---

[31] *See generally* Kanti Bansal et al., *Brief Application Description. Neural Networks Based Forecasting Techniques for Inventory Control Applications*, Data Mining and Knowledge Discovery, Jan. 1998, at 97. (describing using "neural network based data mining and knowledge discovery techniques to solve the problems of inventory in a large medical distribution company," resulting a prototype that "was successful in reducing the total level of inventory by 50% in the organization, while maintaining the same level of probability that a particular customer's demand will be satisfied").

[32] *See generally* Randy Weston, *Bristol-Meyers CEO demands massive supply chain fix*, Computerworld, Nov. 17, 1997, at 47. (noting that after implementing the business process automation software, the implementation team remained in place to "maintain the software and roll out upgrades").

[33] *See generally* Tim McElligott, *Interfáce: (the noun)*, Telephony, Mar. 28, 2005, at 28. (talking about using APIs to build software programs for their specific systems).

replacement. The single technologist is still able to support the system, even with the custom software she has written, because around the time it connected to the Internet, LittleBuyer also replaced all the computers with new models that are more standardized and easier to support and maintain because they have internal automation and software tools. Moreover, the new computers have internally redundant hard drives and an operating system with automatic and transparent data replication to a networked offsite location.[34] This increases the robustness of the inventory tracking system to make sure it is rarely unavailable to support manufacturing.

This progressive narrative could be repeated with striking parallelism for almost every information-handling business process imaginable. Barcode technology wouldn't always be involved, nor would custom programming always be part of the narrative. The roles of the little company and the big company could viably swap. There might not be a software product supplier that fills the market niche. There might be suppliers, but companies might develop the software themselves for considerations of institutional competence.[35] The common theme across all comparable narratives is that paper-based business processes and their attendant human-labor implementation have disappeared as cost and effectiveness pressures force companies to automate. The workers who remain at companies after automated processes are implemented will often have new roles and activities that involve greater use of computing.

Moreover, the automation and reautomation of business processes won't stop anytime soon. For example, computerized voice and email have changed business communication processes in the last few decades, and the next generation of speech recognition technology will bring another wave of change.[36] Continued growth in Internet bandwidth and connectivity will provide new automation opportunities, as will the convergence of mobile computing and cell phones.

One point in the narrative needs additional emphasis: the importance of LittleBuyer's network connecting the computers. Before, and early in, the era spanning the narrative, many businesses had automated some processes, but often only in an isolated manner. These "islands of automation" made particular parts of the business more effective, but information sharing with other processes was often via paper.[37] For

---

[34] Marty Ward, *Protect Your Data*, Computer Technology Review, Sep./Oct. 2006, at 9.

[35] Wesley H. Higaki, *Applying an improved economic model to software buy-versus-build decisions*, HEWLETT-PACKARD JOURNAL, Vol. 46 Issue 4, 61 (1995).

[36] *See generally* Anonymous, *VERINT'S INTELLIFIND*, Call Center Magazine, Jul. 2005, at 10. (describing a analytical tool that "uses speech recognition, audio indexing and categorization technology to create a searchable audio-interaction database for uncovering trends, opportunities, and the ways in which business processes and products are perceived by the marketplace"); Albert Pang, *Re-Engineering Benefits VARs' Telephony Efforts*, Computer Reseller News, Aug. 15, 1994, at 55. (predicting products using speech recognition technology "will change the way corporate America uses voice mail, help desks, and telecommunications switches").

[37] A. Harris, *A Holistic approach to control*, COMPUTING & CONTROL ENGINEERING, Vol. 18 Issue 2, 32 (2007).

example, a payroll computer might take all inputs manually, and only output paper such as paychecks and reports. A machine in a factory might be controlled by specialized computers, but only share data about the manufacturing operation via printed reports. Particularly in the manufacturing sector, eliminating islands of automation was a long-standing problem because machinery suppliers might attach computers to devices without communication capabilities, or with communication capabilities that did not match the other equipment-controlling computers elsewhere in the facility.

The final point of the narrative is to note positive spillover effects from business process automation: computer literacy and personal use of computing. The personal computer accelerated business process automation because a company could affordably provision an employee with her own computing device. Many of the employee's software tasks at work were inapplicable to personal use. But tasks in the personal productivity category, such as making documents with word processing software, or calculating with spreadsheets, or diagramming with drawing packages, were applicable to personal pursuits. A common phenomenon that drove sales for home use early in the personal computer era was the desire to have similar personal productivity software applications available at home.[38]

This narrative provides a concrete example of one software application, computerized inventory control, that could also comprise a software market. The story of LittleBuyer's progression to greater automation touches upon the technological market characteristics discussed in the previous section, in light of the general features used to define a software market.

The scope applied to the description of a particular software market depends on the discussion purposes for which the market description is rendered. Sometimes, that scope follows the business processes automated by software in that market, such as computerized inventory control, or, more broadly, manufacturing resource planning.[39] Sometimes the software market scope maps to its platform, such as "Windows applications," meaning all software running on Microsoft's Windows family of operating systems. This is a very large scope. Similarly, the scope might map to other commercial or industrial segments, such as "accounting software." This Article recognizes the fluidity of market definitions, while proposing that the possibility of such fluidity still allows for meaningful delineations.

The delineation for software markets in health care will focus on the electronic medical record (EMR).[40] The next Part will describe the EMR software market while

---

[38] PAUL E. CERUZZI, A HISTORY OF MODERN COMPUTING 262-63, 272-80 (1998).

[39] The manufacturing resource planning class of software sometimes goes by the more broad, and more broadly applicable, label of "enterprise resource planning" or ERP. *See* Thomas J. Hall, *ERP gone bad: a case study*, MANUFACTURING BUS. TECH., Vol. 26 Issue 4, 16 (2008) (describing a troubled ERP software implementation desired by a manufacturer to help optimize use of new production equipment).

[40] One signal that a market is operating is the appearance of information sources for the market. *See* emrupdate: unbiased independent EMR discussion and resources, http://www.emrupdate.com (2008).

referencing the factors to be developed in a later Part that indicate its potential as a FOSS-disfavoring market.

III. FOSS-DISFAVORING MARKETS: THE ELECTRONIC MEDICAL RECORD (EMR)

Among all major segments of the U.S. economy, health care has lagged in realizing benefits from information technology.[41] While segments such as manufacturing, finance and retail have automated using information technology, health care disproportionately relies on paper flowing through and stored within organizations to handle mission-critical information.[42] Lost are opportunities to provide easy access to multiple users of the information. Lost are opportunities to improve reliability and quality and reduce the cost of healthcare. Lost are the trees providing this paper. This Part will situate the Electronic Medical Record (EMR) within information technology in health care, and then discuss the particulars of the EMR software market.[43]

A. *Information Technology in Health Care*

Computerized information handling in health care has enjoyed the greatest success where either the information needs are somewhat standardized, such as in scheduling and accounting, or where the health care provider has sufficient size to invest in the technology and recover efficiencies of scale. That being said, the conventional wisdom is that handling health care information is pervasively under-automated and overly-costly as a result.[44] This is clearly a national policy concern when one considers that health care is 16% of gross national product and that governmental entities finance a substantial portion of the care that is provided.[45]

---

[41] Brian Lord, Open and Closed Medicine, EHR Scope, Vol. 4, 155-56 (2007), http://www.ehrscope.com/downloads/ehr_scope_fall07_web.pdf.

[42] John Morrissey, A Day in the Life of a Medical Record: Lifting the veil on the security of today's paper-based environment, The National Alliance for Health Information Technology, 1-4, appendix (2006), http://www.nahit.org/cms/index.php?option=com_content&task=view&id=258&Itemid=211.

[43] Bernstein, *supra* note 26, at 18 ("The healthcare industry . . . has established a dependence on IT for maintaining patient records, scheduling, billing and accounting, materials management, and the management of clinical and business operations.").

[44] GAO, Health Information Technology: HHS Is Pursuing Efforts to Advance Nationwide Implementation, but Has Not Yet Completed a National Strategy, GAO-499T, (February 14, 2008). ("Highlights of GAO-08-499T, a testimony before the Committee on the Budget, U.S. Senate"); Robert Wood Johnson Foundation et al., Health Information Technology in the United States: The Information Base for Progress 1:2 (2006) *available at* http://www.rwjf.org/files/publications/other/EHRReport0609.pdf [hereinafter RWJ, HIT in the U.S.]. *But see* Jaan Sidorov, *It Ain't Necessarily So: The Electronic Health Record and the Unlikely Prospect of Reducing Health Care Costs*, HEALTH AFFAIRS, Vol. 25, No. 4, at 1079 (2006).

[45] CMS, National Health Expenditure Accounts 2006 Highlights, http://www.cms.hhs.gov/NationalHealthExpendData/downloads/highlights.pdf (last visited June 17, 2008).

Thus, insufficient use of information technology in health care is cited as an opportunity to dampen rising health care costs as well as reduce errors in care.[46] This opportunity derives from more effective information sharing within and among providers, and goes beyond merely eliminating paper and fax. Information embodied in paper is a rival resource. Only one person can have the paper-based medical record at a time. Labor costs to handle the paper-based medical record, sometimes called the medical chart, are nontrivial. Physical copies must be generated to share it with other providers in a paper-based system. Additionally, information on the paper is not addressable for computer processing. A computer can not scan a wall full of shelves containing paper medical charts to flag those patients whose age and lab results suggest a recommendation for a bone density scan to screen for osteoporosis. If the same set of medical charts are in an EMR, such a scan is likely trivial.[47] Thus, harvesting inference, trends, and situations for alert, is much more effective in an EMR with addressable information fields.

This Article, and this Section's discussion of information technology in health care, will mostly put aside information technology embedded in devices.[48] This is more prevalent in the institutional setting than the physician office setting.[49] Institutions such as hospitals have the high-dollar equipment that often relies on very advanced computing to deliver its benefits. If FOSS were to be used in these devices, that might raise certain issues,[50] but those issues are not this Article's focus. These embedded computers might

---

[46] *See* EHR Key Capabilities, *supra* note 15, at 2-3; T.-Y. Leong, K. Kaiser, S. Miksch, *Free and Open Source Enabling Technologies for Patient-Centric, Guideline-Based Clinical Decision Support: A Survey*, IMIA Yearbook of Medical Informatics 74 (2007) [hereinafter *FOSS for CDA*], available at http://www.donau-uni.ac.at/imperia/md/content/department/ike/ike_publications/2007/contributionsinbooks/leong_2007_open-source-guidelines.pdf.

[47] John R. Christiansen, I Seem to be a Spime: Why Nobody Wants EHRs and PHRs, http://informationlawtheoryandpractice.blogspot.com/2008/04/i-seem-to-be-spime-why-nobody-wants.html ("An EMR, especially in larger organizations, is not a simple electronic "flat file" transformation of the paper record into something like a Word or Excel document, but is a system made up of various applications and databases which store and process patient data.").

[48] *See* U.S. Food and Drug Administration, FDA News: FDA Announces Initiative to Facilitate the Development and Availability of Medical Devices: New Guidelines for Use of Bayesian Statistics in Clinical Trials Issued as Part of Initiative, May 22, 2006, http://www.fda.gov/bbs/topics/NEWS/2006/NEW01377.html; Elaine Remmlinger et al., *Grand Challenges of Information Technology in Medicine*, ASPECTS OF EHR, *supra* note 15, at 433 (noting that unlike medical devices and other technology, information systems and the supporting network are not subject to FDA regulation).

[49] John Pulley, *Picking up the check for EMRs*, Government Health IT, Nov. 26, 2007, available at http://www.govhealthit.com/blogs/ghitnotebook/350133-1.html ("Lay of the land EMR systems come in two basic flavors: big systems for large acute-care settings, such as hospitals and medical centers, and products for the ambulatory or outpatient care market, primarily smaller doctors' offices and group practices."). *See also* Duke & Bowers, *in* ASPECTS OF EHR, *supra* note 15, at 94-95; AMATAYAKUL, *supra* note 15, at 29.

[50] I. Valdes, *FDA Validation a Threat to Free/OSS?*, LinuxMedNews, Apr. 15, 2002, http://linuxmednews.com/1018893577/index_html.

provide information that is channeled, either electronically or via manual entry, into an EMR. Thus, as a series of input devices, embedded computers and information technology in hospital equipment might resemble the "islands of automation" discussed in Section II.B above if they are not or cannot be interfaced with general purpose computers to feed data into EMR software.[51]

Of a similar - yet greater - consequence are external sources of information for the EMR, such as computerized laboratory, medical imaging, or pathology results.[52] These provide some of the information a provider must store in its EMR software, or in its paper chart.[53] Interfacing, and standards for interfacing, among EMR software and these systems are becoming increasingly important as interest in EMR software grows within the health care industry.[54] The companies supplying laboratory testing services to health care providers have automated much of their operations. Their size, and the scale of their operations, mandate automation for effective operations.

Providers, particularly physician groups, have traditionally only felt an operational mandate to use computerized business processes in two areas: patient scheduling; and medical billing as a specialized accounting activity.[55] Like many technologies, software has better chances for successful implementation if complexity can be reduced, partitioned, or left with humans. While some business processes have inherently high complexity, scheduling appointments, in health care or in other sectors, is of a manageable complexity and has been successfully implemented in software in these contexts. Virtually all health care providers use software to schedule appointments. This business process was amenable to automation for several reasons. Scheduling has manageable complexity and software products were available for the task. The information inputs are relatively standardized, such as time, patient demographics, and provider names. Computer costs are minimal, often requiring only one computer even in

---

[51] Mark Tuthill, *Automating Anatomic Pathology*, 29 HEALTH MGMT. TECH. 18, Mar. 1, 2008, *available at* 2008 WLNR 4838457 (discussing interfacing medical devices to information technology systems).

[52] Elizabeth A. Boyer et al., *System Integration*, 89-91 *in* IMPLEMENTING AN ELECTRONIC HEALTH RECORD SYSTEM (James M. Walker et al., eds., Springer 2005) [hereinafter IMPLEMENTING AN EHR]

[53] Rosoff, *supra* note 17, at 131-32.

[54] Health Level Seven, Inc., What is HL7?, http://www.hl7.org/about/ (last visited June 17, 2008) (discussing its strategy to develop "coherent, extendible standards that permit structured, encoded health care information of the type required to support patient care, to be exchanged between computer applications while preserving meaning").

[55] *See* emrupdate.com, Discussion Forum for Billing Software & Services, http://www.emrupdate.com/forums/5.aspx (2008) (providing discussion venue for users of medical practice management software, typically including scheduling and billing functionality).

a multi-physician office. Finally, scheduling software plays an important precursor role for the billing function of medical accounting.[56]

A software implemented patient scheduling system helps operational effectiveness in real-time management of the clinical day, in evaluating the past, and in securing payment to the provider. The software typically keeps a history of the visits scheduled, thus allowing for reporting such as: which provider saw the most patients in a given time frame? Which saw the least? Who had the most cancellations? Where are most of the patients from? This important information is supplemented by the critical role that the scheduling software's "visit list" plays in medical billing. Completed visits are the basis for providers to request reimbursement from third-party payers, typically health insurance companies. These medical billing transactions are increasingly computer-supported.[57] That support originally was computer software to print paper forms to submit claims for payment. Mailing the paper forms is waning in favor of electronic transmittal using standardized transactional formats known loosely as electronic data interchange (EDI). The Internet facilitated greater use of EDI for medical billing transactions and the evolution in this area shows the potential for information technology to increase effectiveness within health care.

Finally, no discussion of information technology in health care would be complete without mentioning privacy and data security. As these topics increase in general importance, their poignancy in health care heightens.[58] This Article will not cover either topic except to acknowledge their critical role in the context of information technology use. Particularly, as the pervasiveness of computing increases, and its modes of use expand, general privacy discussions are increasingly involved with information technology. Important federal regulations regarding health care appeared in the mid-1990s to regulate disclosure of identifiable health care information. It is known as

---

[56] JEFFERY P. DAIGREPONT, AUTOMATING THE MEDICAL RECORD, 51-55 (2d ed. American Medical Association 2004). *See* EHR Key Capabilities, *supra* note 15, at 253. The Institute of Medicine describes the necessity of practice management functionality as follows:

> Electronic scheduling systems for hospital admissions, inpatient and outpatient procedures, and visits not only increase the efficiency of health care organizations, but also provide better, more timely service to patients (Everett, 2002; Hancock and Walter, 1986; Woods, 2001). Use of communication and content standards is equally important in the billing and claims management area-close coupling of authorization and prior approvals can, in some cases, eliminate delays and confusion. Additionally, immediate validation of insurance eligibility should add value for both providers and patients through improved access to services, more timely payments and less paperwork.

Id.

[57] W. E. Hammond, Patient management systems: the early years, Proceedings of ACM conference on History of medical informatics, 153-54, ACM O-89791-248-9/87/001 l/O153 (ACM Dec. 1987).

[58] *See generally* Sharona Hoffman & Andy Podgurski, *In Sickness, Health & Cyberspace: Protecting the Security of Electronic Private Health Information* 48 B.C. L. REV. 331 (2007); Sharona Hoffman & Andy Podgurski *Securing the HIPAA Security Rule*, 10 J. INTERNET L. 1 (Feb. 2007).

the HIPAA Privacy Rule.[59] For example, it requires health care providers to secure contractual promises to keep data confidential from certain third-parties with whom a provider may need to share the information for operational purposes. In 2005, a companion regulation issued called the HIPAA Security Rule governing modes of security for health care data.[60] The security rule, for example, suggests that data should be encrypted when stored in computers.[61]

This Section's discussion shows that health care is not barren of effective information technology. It merely lags behind other sectors of the economy as to the level of operational automation.[62] The lag is apparent because the automation that has been implemented is minimal in the core data used by health care providers. This core data is the medical record. It is a heterogeneous information set of various data types that varies based on numerous factors, such as medical specialty or institutional setting. Its complexity is a challenge to its automation.

B. *The Medical Record*

The medical record is pervasive in health care. During treatment it is a focal point for work activity. After treatment, it waits for the next visit by the patient. Even if the patient never returns to a provider after an initial encounter, regulatory considerations govern retention and use of the information in the medical record.[63] Its importance and longevity relates to the health care providers who generate much of its content.[64]

1. *Information Repository for Health Care Providers*

Health care providers range from sole-practitioner physicians to large facilities such as multi-site hospitals. This difference in institutional setting accentuates the differing information needs by the various medical specialties. The result is a heterogeneous information environment. The phrases "electronic medical record" and "electronic health record" indicate some commonly expected data elements, such as: patient demographics; common health indicators; physician orders, such as prescriptions; medications and allergies; a record of communications with the patient; laboratory or

---

[59] The HIPPA Privacy Rule was enacted by the U.S. Department of Health and Human Services (HHS); the acronym stands for: Health Insurance Portability and Accountability Act (HIPAA). 45 C.F.R. §§160.101-.534 (2007).

[60] 45 C.F.R. §§160.302-.318 (2007). *See* Darren Lacey, *Privacy and Security*, *in* ASPECTS OF EHR, *supra* note 15, at 295-307.

[61] 45 C.F.R. §§160.306, .310, .312 (2007); Lacy, *Privacy and Security*, *in* ASPECTS OF EHR, *supra* note 15, at 301-305.

[62] DAIGREPONT, *supra* note 56, at 2; The White House, *Transforming Health Care: The President's Health Information Technology Plan*, http://www.whitehouse.gov/infocus/technology/economic_policy200404/chap3.html [hereinafter *Transforming Health Care*].

[63] DAIGREPONT, *supra* note 56, at 49; AMATAYAKUL, *supra* note 15, at 26-27.

[64] RWJ, HIT in the U.S., *supra* note 44, at 3:28.

pathology results; and a history of all these for past visits to the provider holding the medical record.[65] But beyond that archetype, the information contained in the medical record can vary greatly.

The institutional setting may have specialized needs for the medical record.[66] One consideration is related to the hospital layout and facilities. For example, location tracking is a necessity for some patients in some institutions. Another example is information related to multi-day stays in the hospital, a visit mode which doesn't typically happen in a physician office. The institutional medical record may also have greater need to allow for a variety of health care providers to contribute to its content as opposed to a small physician clinic. If the hospital is specialized, or focuses on certain types of care, this will also impact its medical record needs.

The variance among institutional providers is undoubtedly surpassed by the variance observable among physician offices, leading to greater potential heterogeneity for the medical record. First, there is variance by medical specialty. An orthopedic physician has different medical information needs than a dermatologist. Second, facilities will differ. Third, physician preferences will differ based on experience, training, taste and personality. For example, some physicians will want a head-shot picture of the patient in the medical record to trigger familiarity.

The last point, physician preferences, may be dominant in the non-institutional setting. Health care is a service business provided in a hands-on manner by skilled workers. While many physicians practice in groups, the profession generally enables a physician to practice alone if she desires. This allows the physician to establish a practice environment tailored completely to her preferences. The clinical work flow, office business processes, and medical record content can reflect the physician's goals and emphasis for the practice. For example, some physicians prefer to handwrite parts of the medical record onto forms that they develop. Others might prefer to deliver the same information to the medical record using dictation that is later transcribed. These preferences can reach to the mundane; a physician might want particular colors of paper for different parts of the chart, or might want a specific system of tabs for the file folder holding the paper medical chart.

Specific provider preferences, whether institutional or with a physician, relate to modalities in practicing medicine. Some aspects of these modalities spring from guidance given by the medical specialty societies.[67] Others spring from the provider's experience and training. Physical facilities also impact the modalities, as does the care experience a provider desires to provide a patient. All these influences manifest

---

[65] DAIGREPONT, *supra* note 56, at 29-33.

[66] *See* AMATAYAKUL, *supra* note 15, at 131-138, 188.

[67] Hoffman & Podgurski, *Finding a Cure*, *supra* note 12, at 45-46 (discussing the potential to incorporate clinical practice guidelines published by various medical societies into EMR software).

themselves in a clinical work flow that expresses desired health care approaches and related business processes, while still meeting the demands of a busy clinical schedule.[68]

2. *Relation to Human Work Flow in the Clinical Setting*

The medical record's structure and access features must support optimal use of the health care provider's time.[69] This is true for physicians, mid-level providers such as physician assistants or nurse practitioners, and for the nursing staff. Optimizing physician time is the most important among these three, but most health care organizations have a profit pressure and thus must consider all operational costs. These are often dominated by labor costs, adding to the emphasis on optimal work flow and a medical record to support that.[70]

Among the various inputs to the medical record, the physician's role is central, regardless of whether the medical chart is for a hospital or for an office. Different physicians will make the medical record at different points in time. Some will complete most or all of it during the patient visit or perhaps immediately thereafter. Others will complete their parts of the medical record later, perhaps by completing self-developed forms, or by dictating information about the visit.

Some physicians may vary when they complete their parts of the record based on the day's events. An emergency surgery might require a physician to complete the medical charts in the evening or the next day even if she would normally complete them the same day as the patient visit. In the hospital setting, the record is made for a visit that might stretch over many days. A paper medical chart inherently offers this temporal flexibility for all these approaches.[71] The work flow organized around a paper medical chart allows providers such as physicians, mid-levels, and nurses to generate or gather information that medical records personnel file into the paper chart. The mobility and readability of the paper chart supports a variety of work flow configurations, and flexibly allows reconfiguration of clinical work flow without computing or software expertise, reconfiguration, or reprogramming.

As a container to hold and organize information, the paper medical record has granularity at the document level, where a document is one or more pages. Addressability is usually accomplished by the use of tabs or similar mechanisms to particular documents comprising part of the medical record. The documents might come from a variety of sources. For example, some documents might be lab results faxed from a third-party laboratory. Alternatively, the lab results might be available on a secure web

---

[68] Ellie E. Henry, *Optimizing Primary Care Practices*, *in* IMPLEMENTING AN EHR, *supra* note 52, at 131.

[69] AMATAYAKUL, *supra* note 15, at 30.

[70] Terry Siek, *Superior Scheduling*, Health Mgmt. Tech., Vol. 29, at 24, *available at* 2008 WLNR 1488820.

[71] *See* Morrissey, *supra* note 42, at 1-2.

site and printed to paper for filing in the paper medical chart. In either case, the lab results document(s) build up under one particular tab in the paper medical record.[72] Their addressability, for search by human vision, is by the category-labeled tab and then by date. The human looking for a specific lab value can read and process the results by scanning for that laboratory test and reading its value. Similarly, other documents in the medical record may be completed by hand, or computer generated as in the case of transcribed dictation, where word processing software is used.

The granularity and addressability of information contained on paper is inherent to how paper works. In other words, its design principle is to carry no meta-data (data describing the attributes of other information) with the information printed on the page. When the discussion turns to automating the medical record in Section C below, the question of meta-data and its granularity and addressability will be paramount.

3. *Regulation of the Medical Record*

A complete primer on the regulatory forces bearing on the medical record is beyond this Article's scope, so this subsection will highlight those regulatory forces that are of the greatest prominence for the market characteristics influencing automation of the medical record. One common influence imposed by from these forces is the need to keep the medical record confidential. Confidentiality consists of both limiting information exposure within the organization to those who need to see it and having appropriate processes in place for implementation in the event that confidentiality is breached.[73]

For providers who accept Medicare patients - which includes most providers - the medical record, under federal law, must be sufficient to support the requested claim for payment.[74] The details of these requirements are not important, but they are one of several reasons why providers regularly engage third-party consultants to spot check a random sampling of the medical charts in an auditing process.

Chart audits also facilitate compliance with a provider's medical malpractice carrier's standards, and this illustrates how the tort system acts as a regulatory force on the medical record. For professional liability arising from medical malpractice and other situations where the victim is a patient receiving treatment from a provider, the discoverability of the medical record makes it an important source of evidence for the tort

---

[72] DAIGREPONT, *supra* note 56, at 27-31.

[73] Darren Lacey, *Privacy and Security*, *in* ASPECTS OF EHR, *supra* note 15, at 286-91.

[74] J.D. Epstein et al., *Medicare*, 69-70, 80-81 *in* FUNDAMENTALS OF HEALTH LAW (American Health Lawyers Association, 3rd ed. 2004) [hereinafter FUNDAMENTALS] (discussing requirements for physician certification as a provider under Medicare); Department of Health and Human Services, Centers for Medicare & Medicaid Services, Medicare Enrollment Application, Clinics/Group Practices and Certain Other Suppliers, CMS-855B, at 15, http://www.cms.hhs.gov/CMSforms/downloads/cms855b.pdf (requiring disclosure of location where provider stores its medical records).

system. As a result, self-interested tampering with the medical record is a risk in the medical malpractice setting.

State law also bears on the medical record. Most states require providers to retain the medical record for some number of years after the last patient visit. Many states have implemented privacy, data security, or physical security protections that require providers to exercise care and caution in handling and storing the medical record. Some states regulate other minor aspects of the medical record, such as the price a provider can charge for supplying copies of the medical record to a third-party.[75]

At the federal level, the HIPAA privacy and security rules mandate various provisions that tend to emphasize more careful handling of the medical record. The rules cover protected health information[76] generally, but most of this is stored by a provider in the medical chart. In other words, most providers have formal or informal document retention policies where all paper other than what is stored in the medical chart is destroyed. The HIPAA provisions are the most well-known and prominent regulatory forces influencing the medical records. The HIPAA security rule is of particular note because it applies to medical records whether they are stored in paper or electronic form.[77]

In the world of paper medical charts, all of these forces along with the clinical importance of the chart have led to systematic paper-based filing, storage and retrieval systems. While these systems seem quaint in an electronic age, their efficacy should not be underestimated. Aside from the relative cost factors, automating these systems has advantages and disadvantages from a business process perspective.[78]

C. *EMR Software*

Characterizing the EMR software market starts with situating the term "electronic medical record" among some other phrases.[79] The broadest term is "electronic health information." More narrow is "electronic health record" (EHR). Further, this Article distinguishes EMR as yet narrower than EHR under the logic that the EHR definition includes billing and medical accounting information. EMR, by contrast,

---

[75] *See e.g.* K.S.A. § 65-4971(b) (2007), http://kansasstatutes.lesterama.org/Chapter_65/Article_49/65-4971.html (establishing the maximum fees that Kansas medical care providers can charge for reproduction of medical records).

[76] 45 C.F.R. §§160.103 (defining "[i]ndividually identifiable health information [as] information that is a subset of health information . . . [and r]elates to the past, present, or future physical or mental health or condition of an individual; the provision of health care to an individual; or the past, present, or future payment for the provision of health care to an individual; and . . . [t]hat identifies the individual . . .").

[77] Lacey, *Privacy and Security*, *in* ASPECTS OF EHR, *supra* note 15, at 302.

[78] *See* Morrissey, *supra* note 42, at 7 ("Both the current paper and the envisioned electronic methods of keeping and using medical records have their downsides.").

[79] AMATAYAKUL, *supra* note 15, at 4-7.

focuses on the clinical work of a health care provider centered on the medical chart.[80] The word "medical" emphasizes the work flow of the medical professionals delivering care.[81]

This taxonomy correlates to the fact that market penetration of practice management software packages is higher than EMR software.[82] The practice management software packages provide, among other things, patient scheduling and demographics, and partial or full medical billing support. There has always been a strong motivation to automate the monetary recovery processes of even a small medical practice, in part because the automation project itself was substantially easier than automating the entire medical record.

Multiple goals underlie the rationale for most business process automation, such as: cost reductions via paper elimination; better information access; enhancing the value provided by human activity; and computerizing rote work. Computerizing a process typically entails redesigning the process, to at least some degree, in order to emphasize the beneficial aspects of computerizing, and deemphasize its disadvantages compared to the precursor technology.[83] For example, until very recently, much more information could be readably presented on a single piece of paper than on a computer monitor. This simple reality has tremendous implications for automating a business process. The implications go beyond "break the information into logical groups for display on the small screen."[84] Implications range from how many computers are needed and where they are located, to which humans do what tasks, and to whether certain tasks continue to be carried out by human activity. Most beneficial goals for business process automation are long-term because the initial cost and short-term disruption to implement change in an organization can be exasperating and difficult.

Vendors in the EMR software market must pay attention to these realities, and soften the blow as much as possible.[85] Nonetheless, installation of an EMR system is

---

[80] Harold P. Lehmann et at., *Introduction*, *in* ASPECTS OF EHR, *supra* note 15, at 2.

[81] The National Alliance for Health Information Technology, Defining Key Health IT Terms (Work in Progress): Interim Draft Report Prepared for Public Comment, at 8-11, http://definitions.nahit.org/doc/InterimDraftPublicComment2.22.pdf (last visited Feb. 25, 2008).

[82] Ashish K. Jha et al., *How Common Are Electronic Health Records in The United States? A Summary of the Evidence*, HEALTH AFFAIRS W503, W504 (web exclusive 2006) (noting that one survey reported that "although 67 percent of the clinics had implemented basic IT systems to support business operations, fewer than 10 percent of clinics surveyed were using electronic systems to support individual patient care").

[83] Jean A. Adams et al., *Workflow Assessment and Redesign*, *in* IMPLEMENTING AN EHR, *supra* note 52, at 36-37.

[84] AMATAYAKUL, *supra* note 15, at 196-198.

[85] Elaine Remmlinger et al., *Grand Challenges of Information Technology in Medicine*, ASPECTS OF EHR, *supra* note 15, at 419-20.

typically a difficult process of change for most health care providers.[86] Many factors determine the degree of difficulty, but among the most important are employee attitudes toward, and proficiency with, computing.[87]

For software vendors, the EMR market is tantalizing because the need is high and the uptake of the technology in healthcare has been low thus far. Estimates put EMR installation in the physician office submarket at less than fifteen percent. Institutional penetration is greater, somewhere in double-digits but below the fifty percent mark. Size matters; the larger hospitals are more likely to have EMR software, as are the larger physician groups.[88]

The common, traditional components one might expect in an EMR software package correspond with what is found in the paper chart: demographic information; common health indicators; physician orders, such as prescriptions; medications and allergies; a record of communications with the patient; laboratory or pathology results; a record of any consults where the patient was directed to other providers; information related to procedures or surgeries; and the provider's evaluation, assessment and treatment plan.[89] In both paper and electronic medical records, some or all of these may be involved for the current episode of care, but patient history is also kept with either paper charts or electronic records. However, an EMR provides opportunities to add some new capabilities using the automation capabilities of computing.

Adoption by the marketplace suggests that the following are some of the areas where computerizing the medical record has had the biggest impact on patient care: decision support systems to assist providers with diagnosis and disease management, including automated access to clinical practice guidelines; standardized medical vocabularies as a step toward harmonizing the description of medical conditions among providers; alerts and reminders based on patient health information; interaction analysis among drugs and among drugs and laboratory tests; enhanced practitioner order entry and management for prescriptions and other directions to implement the plan of treatment, such as lab tests ordered; electronic communication and connectivity to share data with other systems and for multiple points of access to the medical record, perhaps allowing providers from multiple locations to contribute to care; enhanced interface and support for administrative processes such as appointment rescheduling, medical coding and

---

[86] Catherine M. DesRoches et al., *Electronic Health Records in Ambulatory Care – A National Survey of Physicians*, 359 N. ENGL. J. MED. 50, 56-59 (2008).

[87] James M. Walker, *Useability*, *in* IMPLEMENTING AN EHR, *supra* note 52, at 50-53; Wanda L. Krum & Jack D. Latshaw, *Training*, *in* IMPLEMENTING AN EHR, *supra* note 52, at 60; Nancy M. Lorenzi, *Clinical Adoption*, *in* ASPECTS OF EHR, *supra* note 15, at 378-81.

[88] Hoffman & Podgurski, *Finding a Cure*, *supra* note 12, at 2-3 (summarizing research of EMR penetration in various segments); Steve Lohr, *Most Doctors Aren't Using Electronic Health Records*, N.Y. TIMES, June 19, 2008; RWJ, HIT in the U.S., *supra* note 44, at 5:46-5:47.

[89] DAIGREPONT, *supra* note 56, at 25-30; Duke & Bowers, *in* ASPECTS OF EHR, *supra* note 15, at 90, 96.

billing, and charge capture; support for patient-population evaluations such as when there is a drug recall and a provider wants to inform all of its patients who might be using the drug; customizable templates for quickly assembling the physician's note about the visit; and enhanced capacity to handle images of various sorts, such as radiology for x-ray or other images.[90]

The paper medical chart contained the traditional components, while EMR software adds the new capabilities. However, in both technologies, there is tremendous opportunity for variety of implementation. EMR software may range from imaging-based systems whose data addressability is not substantially better than a paper system to packages where all information is addressable and selectable.[91] For full addressability, all of the data needs to be described by its own meta-data. This allows for precise data extractions and for sharing data with other software systems, assuming that both software systems use compatible data interchange mechanisms. Product implementations are more likely to have addressable information in the institutional EMR software market. This means that the hospital EMR systems are less likely to be dependent on image-based EMR approaches which continue the document-level granularity found in the paper chart. These alternative approaches vary in software development complexity, and thus present a range of cost alternatives to buyers.[92]

EMR software, like many enterprise software packages, is often licensed in modules so that customers can control costs by procuring pieces of software only as they become necessary. Some physicians won't need any modules with new capabilities. Some software vendors won't offer them. Some physicians will be happy with an image-based EMR software package that doesn't handle prescription orders (other than as scanned documents) because the pharmacies in her region don't have e-prescribing capability anyway. Faxing or phoning the prescriptions to the pharmacies is all that is available, so there is no need for EMR software to transfer them electronically. Partitioning software into modules priced accordingly allows software vendors a strategy to expand the customer base by differentiating the product offering around a common set of elements, often thought of as the "base software" or "core package." Thus, even within either of the EMR submarkets (physician groups versus institutional settings) the product offerings will exhibit substantial technological variety.[93]

---

[90] Hoffman & Podgurski, *Finding a Cure*, *supra* note 12, at 3-14; AMATAYAKUL, *supra* note 15, at 193-218.

[91] AMATAYAKUL, *supra* note 15, at 147-149.

[92] Signaling that the EMR software niche is a market, information sources have arisen for exchanging information about products in that market. *See* Electronic Health Records News & Views, *ehr*Central @ The Provider's Edge, http://www.providersedge.com/ehr_news_views.htm (last visited Jun. 26. 2008).

[93] RWJ, HIT in the U.S., *supra* note 44, at 2:8.

1. *Institutional Setting*

Many institutions such as hospitals implemented islands of automation that grew into areas of automation, which eventually evolved into fully automated institutions. While the EMR software was often the last piece of the puzzle, institutions have traditionally had several advantages as buyers in the EMR software market as compared to physician offices.

First, many institutions have full-time information technology employees who can manage the EMR software procurement process.[94] Hospitals typically already have computer systems in place for billing, accounting, general office support, and some isolated clinical systems. These systems necessitated in-house technical management personnel at the hospitals, even if the actual technicians and support personnel were contractors. Procuring enterprise software that automates business processes is a non-trivial task.[95] The in-house managers have usually developed their procurement expertise with earlier non-EMR systems. The procurement manager must be skilled in negotiations, vendor evaluation, and internal project promotion. She must understand internal requirements, the offered technology, and the computing platforms on which it will run. There are many pitfalls that can haunt the procurement process for proprietary software, so there is no substitute for judgment informed by experience.

Second, institutions have a greater scale of operations, which allows for more favorable economics in calculating when and how the EMR software justifies its cost in returned value.[96] The final point is related: funding for capital outlays is a more regular occurrence at institutions. Thus, the investment for a computing system and EMR software doesn't seem so much like a once-in-a-lifetime event.

There are perhaps about a dozen EMR software vendors that comprise most of the active installations in the institutional setting.[97] Some EMR software vendors have product offerings for both the institutional setting and the physician office setting. Some vendors have a presence in other clinical systems, such as hospital laboratory automation, but have entered the institutional EMR software market. Overall, the institutional market is fragmented - with no one vendor dominating, but it is likely less fragmented than the market for smaller-scale systems used in doctor offices.

---

[94] RWJ, HIT in the U.S., *supra* note 44, at 2:15-2:17.

[95] Frank Richards, *Managing the Client-Vendor Partnership*, *in* IMPLEMENTING AN EHR, *supra* note 52, at 101-07.

[96] RWJ, HIT in the U.S., *supra* note 44, at 5:42.

[97] The EMR market fragmentation eliminates the plausibility of pegging an exact count; some vendors focus exclusively on the institutional market, while others offer software products for both institutions and physicians' offices. RWJ, HIT in the U.S., *supra* note 44, at 3:26 ("There were very few high quality surveys of inpatient EHR use"). One commentator characterizes the institutional market as "A handful of heavy hitters dominate the acute-care [institutional] market, including Cerner, McKesson, Siemens, Meditech and Eclipsys. Launching systems built by those companies is a major undertaking that can take 18 months to complete at a cost of millions of dollars." Pulley, *supra* note 49.

2. *Physician Office Setting*

Physician groups, particularly those that are small, are unlikely to have full time information technology employees. Instead they often rely on contractors or vendors for technical support of their computer systems. While they likely use some type of practice management software for scheduling and to support billing, they don't necessarily have significant in-house computing expertise.

The EMR software offerings for the physician office setting include products by some of the vendors serving the institutional market, and products from many other vendors that focus on the physician office setting. The product count is perhaps over one hundred, if not more.[98] This greater fragmentation makes sense due to the large number of medical specialties practiced among physicians. Also, many physicians spend more time in their office than in a hospital or other institution. As a result, they are more likely to desire EMR software that specifically suits their needs and tastes, leading to a proliferation of vendor offerings.[99]

Like the institutional setting, physician offices may have varying degrees of automation. Even physicians that use an EMR software package may still have some documents that they either keep in paper form or scans into image storage modules within the EMR software. The imaged information is less accessible because it is not addressable below the document level. Thus, if a physician does not have an EMR that will accept laboratory test results into database fields, the lab results may simply be scanned into the EMR software as an image file. In this case, the EMR can at least store the image and associate it with a patient record, though it likely won't be able to report the results of a cholesterol screening test from an image file alone.[100]

Due to the fragmented nature of the product offerings for the physician office, a wider variety of functionalities is provided, particularly among the new capabilities discussed above. Many of the EMR vendors offer a base system but then have specific modules for the various medical specialties. These products may provide special capabilities that allow for flexibility in the clinical work flow. Some of these products relate to the software's support for mobile computing devices such as handheld

---

[98] One assessment technique for the number of physician-office EMR vendors is the membership of a vendor association, which is at 42 at the time of this Article. HIMSS Electronic Health Record Vendors Association, http://www.himssehrva.org/ASP/members.asp. Another source gives approximately one hundred product names. HER Scope, *supra* note 41, at 25-145 (two products listed per page, not all of which are EMR software packages, but at least a majority likely are, which is well over one hundred). Another source estimates two hundred. Pulley, *supra* note 49 ("Unlike EMR software for acute-care [institutional] facilities, the market for ambulatory systems [physician-offices] is spread among an estimated 200 vendors. And attrition is high. Some 20 percent to 30 percent of such vendors leave the market each year, typically to be replaced by new entrants . . .").

[99] Pulley, *supra* note 49 ("there is a vast difference between what the hospital needs and what ambulatory care needs").

[100] AMATAYAKUL, *supra* note 15, at 146-47, 196-201.

computers. The screens in these devices are often smaller, sometimes leading to the need for specific software or operating system support.[101]

Due to the increasing availability of broadband Internet connections in the first decade of the twenty-first century, physician offices increasingly obtained a high-speed Internet connection to support general office operations. One result of this was to reawaken the push for network-enabled regional and national health information exchanges, a push which clearly has implications for the EMR systems that would feed data into these exchanges.

3. *Relation to Health Information Networks*

Health information exchanges have been a topic within health care policy since the early 1990s.[102] System-wide, they are envisioned to facilitate more effective and timely data sharing among providers and to support the creation of vast databases of health information to support research.[103] The first goal was originally frustrated by the lack of sufficient bandwidth among health care providers, but the burgeoning high-speed Internet changed that. With a pervasive Internet, providers can share electronic information. This facilitates document-level information exchange with imaging-based EMR software. Moreover, and more effectively, communication channels through the Internet allow addressable data sharing among systems. For example, a physician might send a tissue sample to a laboratory. When the test results are ready, her staff logs into the laboratory's web site, not only to view the results, but also to download them directly into her EMR software as addressable data fields. This electronic data transaction could alternatively be arranged to occur automatically. Extending this example to all information sharing applications in health care, one can begin to see the possibility of greater effectiveness in care as well as significant cost savings.[104]

One common situation where a health information network is employed is for a hospital with attending physicians who practice in the hospital but also have office-based EMR software. Network linkage through the exchange enables the hospital's EMR software to interoperate with the office-based systems. Hospitals may provide health

---

[101] AMATAYAKUL, *supra* note 15, at 196-197.

[102] Don E. Detmer, *Public Policy Issues for Computer-based Patient Records, Electronic Health Record Systems, and the National Health Information Network, in* ASPECTS OF EHR, *supra* note 15, at 144-45.

[103] Helga E. Rippen & William A. Yasnoff, *The Electronic Health Records System in Population Health, in* ASPECTS OF EHR, *supra* note 15, at 65-66; Hoffman & Podgurski, *Finding a Cure*, *supra* note 12, at 6-7. For population-based health research, the health information would be aggregate; information allowing identification of particular individuals would be removed or securely partitioned, leaving what is sometimes called "de-identified" information. Hoffman & Podgurski, *Finding a Cure*, *supra* note 12, at 49-50.

[104] *Transforming Health Care*, *supra* note 62; Hoffman & Podgurski, *Finding a Cure*, *supra* note 12, at 3-4; Don E. Detmer, *Public Policy Issues for Computer-based Patient Records, Electronic Health Record Systems, and the National Health Information Network, in* ASPECTS OF EHR, *supra* note 15, at 147-50.

information inputs such as radiology or other specialized laboratory tests that generate information necessary for the patient care in the physician's office, particularly for office visits after a hospital stay.

Another benefit of health information networks stems from the use of addressable data. The most effective implementation is to associate the data with its meta-data at the time of its creation, that is, when it is initially stored in the EMR software.[105] Health outcome research is more effective if the information in the database is addressable because there is meta-data describing it. If these associations don't happen at the point of care, it will be costly for researchers to review imaged documents for the data. This impedes both the health value of the de-identified research data and limits its efficacy for public health uses. The research activity looks at the health history in the data, and thus is backward looking. The public health uses might be forward looking, such as evaluating whether certain populations are at greater risk from a new infectious disease. An interconnected health information network would benefit both experts and governmental authorities in such a situation.[106] The interconnections are more beneficial if all the information is addressable.

Health information exchanges are an increasingly visible policy issue at the time of this Article, resulting in various suggestions to facilitate their arrival. One suggestion relates to standards for interoperability and data exchange among software that handles health information.[107] The interoperability issue is beyond the scope of this Article, but one experiment by a Federal agency imagines facilitating that interoperability through a FOSS EMR package, with the additional goal of generally promoting EMR software adoption.[108] Increasing adoption of EMR software that is increasingly interoperable establishes a foundation favorable to health information exchanges.

IV. FOSS MARKETS AND MOTIVATIONS

Before reviewing the EMR software that underlies one of the FOSS entrants in the EMR market in Part V below, this Part will briefly describe the landscape of the FOSS movement. While the movement has many strands, to simplify it, this Article divides the movement into two camps. Each camp has distinct motivational preferences for FOSS. This recognition will be helpful later when this Article discusses the motivational mix of FOSS. These preferences also relate to the business models underlying FOSS development and distribution.

---

[105] AMATAYAKUL, *supra* note 15, at 160-163.

[106] Detmer, *supra* note 15, at 141, 144-45.

[107] Transforming Health Care, *supra* note 62; Fischetti et al., Standards *in* ASPECTS OF EHR, *supra* note 15, at 253-60.

[108] Goetz, *supra* note 12.

A. *Free Software*

Arising as a counter-force against proprietary software development and licensing, the free software camp originated the FOSS movement by developing licensing techniques that were novel to the world of software licensing in the late 1980s: require generally available public source code disclosure and prohibit use royalties. Linked to these is the term "copyleft" – a pun of copyright and its institutional values, but also a label for a mechanism of reciprocity or extension of FOSS licensing terms, such as source code availability and the anti-royalty provision, to intermixed or further developed software.[109]

Embodied in a license, these terms are means to implement a philosophy of functional self-determination and freedom with the software on one's computer.[110] The embodying license is version two of the Free Software Foundation's ("FSF") GNU General Public License ("GPLv2"),[111] arriving in 1991.[112] The FSF's progenitor, Richard Stallman, implemented these novel licensing concepts in GPLv2 toward his greater ends of software freedom. GPLv2 became the license for important programs generated by Stallman and others through FSF-affiliated software development projects. By its own language, GPLv2 also suggested itself for use on other software.

A variety of industry developments in the decades following the GPLv2's arrival combined with the license's potent ideological force and clever use of copyright to propel FOSS licensing into a prominent and path-breaking place within information technology world-wide. Its force and presence, and lighting-rod character, has grown over time, with

---

[109] Under one sense, "copyleft" expresses the FOSS goal to protect the general availability of a software work, which is opposite copyright's typical use for software: generally protecting and prohibiting use of the work by others, while perhaps licensing some narrow use for some number of users. Under another sense, copyleft refers to a reciprocity rule given in a FOSS license. *See* ROSEN, *supra* note 2, at 105-06. The FSF, involved in the origination of the label "copyleft," relates it to license term reciprocity with the purpose of software freedom. *See* Free Software Foundation, What is Copyleft? http://www.gnu.org/copyleft (last visited Sept. 19, 2007) ("*Copyleft* is a general method for making a program or other work free, and requiring all modified and extended versions of the program to be free as well." (emphasis in original)). *See also* Greg R. Vetter, *"Infectious" Open Source Software: Spreading Incentives or Promoting Resistance*?, 36 RUTGERS L.J. 53, 129-30 (2005) (discussing GPLv2 copyleft) [hereinafter Vetter, *Infectious OSS*].

[110] Free Software Foundation, The Free Software Definition, http://www.fsf.org/licensing/essays/free-sw.html (last visited Sept. 19, 2007).

[111] Free Software Foundation, GNU General Public License, version 2 http://www.gnu.org/licenses/old-licenses/gpl-2.0.html (last visited Sept. 19, 2007) [hereinafter GPLv2].

[112] *See* GLYNN MOODY, REBEL CODE: THE INSIDE STORY OF LINUX AND THE OPEN SOURCE REVOLUTION 19, 26-29 (2001).

the GPL[113] remaining the dominant license among its many imitations in mind-share, if not code-share.

B. *Open Source Software*

Like many movements, as its success surged, the FOSS movement became increasingly multi-stranded, leading to the open source camp.[114] The free software camp contains the FSF and Richard Stallman. The open source camp contains Linus Torvalds, the leader of the Linux kernel project. The open source camp emphasizes the software development advantages arising from FOSS licensing. The Linux kernel project is the basis for a number of operating system distributions that are popularly called "Linux" but which the FSF argues should be called "GNU/Linux" to emphasize the principles of software freedom associated with the GNU project. A GNU/Linux operating system distribution rests on the Linux kernel, but typically contains critical components from the GNU project.[115] The FSF's vocabulary control argument is but one example of the group's explicitly political orientation, and proclivity to evangelize the merits of free software.

The open source camp is willing to entangle FOSS with commercial interests to a greater degree. FOSS licensing can make strange bedfellows and has gathered corporate advocates as well known as IBM, even though, at first glance, the FOSS premise of open shareable source code is opposite traditional software licensing approaches IBM championed in earlier decades.

Corresponding loosely with the open-source camp, another major license type pre-dated GPLv2: the attribution-only license. Although many important FOSS projects operate under attribution-only licenses, these licenses merely claim copyright and then require that an attribution statement appear with the code. The attribution-only license does not have the features to help ensure that the software remains transparent and shareable, although it often does so under institutional and practical influences. These licenses allow others to do practically anything with the software, including incorporation into proprietary software, as long as there is notice that the software originated from the original project. These licenses do not even require that the source code be available – a

---

[113] Occasionally, there may be a need to refer to the GPL without identifying a specific version. GPLv2 did not explicitly handle granting and terminating permissions to practice software patent rights. This, along with the need for various other changes, resulted in version 3 of the GPL. *See* Free Software Foundation, GNU General Public License, version 3, § 11, June 29, 2007, at http://www.gnu.org/licenses/gpl-3.0.html (last visited Feb. 5, 2006) [hereinafter GPLv3]; GPLv3 First Discussion Draft Rationale, at http://gplv3.fsf.org/gpl-rationale-2006-01-16.ps (last visited Oct. 3, 2007) (discussing the decision to create version 3 of the GPL).

[114] *See* Vetter, *Exit & Voice in FOSS*, supra note 11, at 205 (noting that the line between the two camps is not a bright line).

[115] The GNU/Linux operating system is sometimes referred to as Linux by the popular press, but the free software camp disputes this labeling. *See supra* note 21.

key norm of the FOSS movement. Thus, attribution-only licenses are the least restrictive type of licenses used for FOSS projects.[116]

Under both copyleft licenses such as GPLv2, and non-copyleft attribution-only licenses, the FOSS movement produces software with a decentralized development methodology relying on source code transparency and Internet-coordinated activity. Thus, the group of developers and users for a project may be fluid. Some users are contributing developers in either major or minor ways. A user who discovers a software defect and communicates this to the developers is a contributor to the betterment of the software, even though she isn't programming. The development groups, while typically decentralized, coordinate through a hierarchy of leaders on a project. For a small project there might be just one leader and a few programming contributors. Larger projects may exhibit various organizational forms to coordinate activity. FOSS licenses allow a publicly available distribution, but don't command it. Developer groups, however, often want a user base, which leads to public distribution of the software. Many FOSS licenses trigger the FOSS conditions upon such a public distribution.

With fluid developer and user groups, over time an actively developed FOSS program becomes a composite of code from a number of software developers. Typically, the FOSS program is most useful in whole. Thus, users who download and run the software are beholden to a group of copyright authors, or to a trusted central organization to whom they may have assigned their copyright. While sometimes one wants only a component of the project, often the entire program is desired. From both a copyright and a patent perspective, this suggests the need to "clear rights" in the program's instructional composite.[117] Thus, the program as a whole (all of its source code, object code, and related files and instructions) benefits if intellectual property rights arising from copyright and patent law are "cleared" by upstream contributors having granted permissions through the web of FOSS licensing.

Finally, it is important to understand that FOSS-licensed software is not public domain software. The conditions, particularly those of copyleft licenses, seek to ensure that the code remains in a FOSS mode of development. Either FOSS or proprietary software can benefit from incorporating public domain software into their code base. The

---

[116] Given that attribution-only licenses do not require that the software be free of royalties, or that source code be available, there is some question as to whether attribution-only licenses are properly called FOSS. They are often categorized this way, however, because the programmers manage these projects using freely available source code and internet-based collaborative development.

[117] The instructional composite is the lynchpin of computing. It defines what the computer will do. It is a necessary, but not sufficient, predicate to a successful computing result. It is what many people are referring to, in part, when they use the term "source code." The instructional composite, however, takes different forms at different stages in the software development process. These variations in form produce the crux of one problem at which FOSS is aimed: a nonhuman readable form of the instructional composite, often called the "object code," is the only instructional composite available with most traditional software.

primary example of a FOSS-based EMR discussed in the next Part begins in just that way.

V. THE VISTA EMR SOFTWARE AND ITS FOSS INCARNATION(S)

Unlike the server operating system software market, where GNU/Linux is a viable FOSS competitor with significant market share, the EMR market has negligible FOSS penetration.[118] Although there are a number of FOSS EMR products,[119] this part will focus on products derived from a large, government-developed enterprise software system, called VistA,[120] used in Veterans Affairs hospitals. The VistA system offers a unique opportunity for FOSS-based market penetration at the institutional level in the EMR software market – in part because the FOSS offerings based on VistA don't have to start from scratch.

A. *The Veterans Affairs VistA Software*

The U.S. Department of Veterans Affairs (VA) operates hospitals. Beginning in the late 1970s, a splinter group of geographically decentralized technologists at the VA began programming software that would eventually evolve to automate most aspects of the medical record in the VA's institutional setting.[121] The software is a collection of modules under the label VistA. The software was developed with a high degree of physician user input by decentralizing development. This development approach was possible because the late 1970s and early 1980s ushered in the affordable minicomputer. Each institution had its own minicomputers and VistA programmers who worked closely with clinical staff to conceive and program the functionality.[122] This allowed for greater responsiveness to user needs while promoting a wide variety of functionality.[123] Even with the decentralization, because the VA is a single organization, the software

---

[118] Sujansky & Associates, Appendix E at 2, *supra* note 14.

[119] Sujansky & Associates, Executive Summary & Appendix C-D, *supra* note 14. The Sujansky & Associates report assesses the functionality and business model for several FOSS EMR software products, and briefly reviews other noteworthy FOSS EMR products. The products reviewed in detail are aimed primarily at the physician-office segment of the EMR market. *Id.* at Executive Summary.

[120] United States Department of Veterans Affairs, VistA Monograph Home, http://www.va.gov/VISTA_MONOGRAPH/ (describing the current system, the Veterans Health Information Systems and Technology Architecture (VistA), and its predecessor system, the Decentralized Hospital Computer Program (DHCP), noting that VistA is a "rich, automated environment that supports day-to-day operations at local (VA) health care facilities").

[121] George Timson, The History of the Hardhats, http://www.hardhats.org\history\hardhats.html (last visited June 23, 2008) [hereinafter VistA-Hardhats]; Joseph Conn, VistA: A look back and a look forward, Modern Healthcare Online, http://www.modernhealthcare.com/apps/pbcs.dll/article?AID=200770118002 (last visited June 23, 2008).

[122] *See* VistA-Hardhats, *supra* note 121.

[123] Fred Trotter, Why is VistA good? the VistA open source development model, http://www.fredtrotter.com/2007/11/10/why-is-vista-good-the-vista-open-source-development-model.

functionality could be incorporated into sharing repositories in a self-reinforcing cycle after the splinter groups' efforts were recognized as legitimate and approved advances.

As the VistA EMR software evolved, it became known inside and outside the Federal government. A group of programmers involved with VistA used the Freedom of Information Act (FOIA) to prompt disclosure of the source code to the public.[124] It developed a reputation as a quality software system, leading to some acclaim.[125] This established an ongoing FOIA feed of the source code as the VA created new versions because the "vast majority" of the source code was, and is, releasable without redaction.[126] The VistA system remains at the time of this Article the primary system for virtually all automation of clinical information at the VA, although some proprietary software has been applied in certain instances.[127] Thus, VistA is under active development even though its technological roots are close to three decades old, and it has a place in the information technology planning for the VA's future needs.

While VistA has impressive functionality for the EMR, it does not include complete medical billing capabilities because the VA's needs in this area are minimal.[128] Thus, while providing a FOSS possibility for the EMR, VistA needed additional capability or interfaces to practice management or medical billing software with those additional capabilities. Similarly, the FOIA-released VistA source code does not have easy-to-commercially-reuse data exchange interfaces to third-party laboratory companies,[129] a common need for both institutional and office-based EMR software.

B. *FOSS Offshoots of VistA*

The disclosed VistA code provided an opportunity for a new FOSS presence in the EMR software market. Several companies involved themselves with the VistA

---

[124] DVA's Vista Software available through FOIA, http://www.hardhats.org/foia.html; WorldVistA, VistA History, http://worldvista.sourceforge.net/vista/history/index.html.

[125] United States Department of Veterans Affairs, VA Receives 2006 Innovations in Government Award, http://www1.va.gov/opa/pressrel/pressrelease.cfm?id=1152 (Jul. 10, 2006). *See also* Nancy Tomich, interview of Tom Munnecke, *available at* http://blip.tv/file/405389 (discussing the "history of the Veteran's Administration's Decentralized Hospital Computer Program (DHCP), now called VistA.")

[126] DVA's Vista Software available through FOIA, http://www.hardhats.org/foia.html..

[127] Peter Buxbaum, *VA's health IT gamble: Can the Veterans Affairs Department tighten security without stifling a culture of innovation that has fielded some of the best health IT in the world?*, GOVERNMENT HEALTH IT, (Feb. 4, 2008) (discussing a contract to install a proprietary-software lab system from Cerner).

[128] Trotter email, *supra* note 14 (explaining that the VA has the need to bill secondary medical insurers). Trotter explains:

> VistA only recently added billing functionality and for the most part, it has proved to be worthless for commercial installations. It lacks an advanced billing system and most successful commercial installations of VistA move billing information into a proprietary billing system.

Id.

[129] Trotter email, *supra* note 14.

software, and a non-profit foundation was established to "extend and collaboratively improve the VistA electronic health record and health information system for use outside of its original setting."[130]

The FOIA VistA source code provides a unique experiment for FOSS because it can go forward under open source approaches such as attribution–only licensing, or under free software approaches such as the GPL. The VA-supplied source code is effectively in the public domain. Viewing that source code as a resource to be harvested, very few physician offices or even hospitals have technical personnel who can directly implement it. There are other factors counseling against such a move, including risk aversion by institutional managers responsible for information technology. Having a vendor to blame when things don't go well is better than a decision to implement internally if the project runs afoul. Regardless of the FOSS mode of deployment, the FOIA code needs supplementation for use outside the VA.[131] This provides the opportunity to entangle some copyright protected code with the original VA code. Such entanglement is often the basis for wrapping a license[132] around the entire supplemented package, regardless whether that license is a proprietary one or a FOSS license. In other words, in addition to multiple FOSS modes of deployment, the FOIA VistA code could also be incorporated into a proprietary software product.

1. *The WorldVistA Community*

A non-profit foundation, named WorldVistA, has a broad mission to make "medical information technology better and universally affordable."[133] Its efforts focus on leveraging the FOIA-disclosed VistA source code and promoting a community of technologists to collaborate to improve the software. WorldVistA's other efforts include marshalling a tailored version of VistA through a certification process to allow it to claim an interoperability baseline.[134] The tailored version carries the label "VistA EHR VOE," where "VOE" stands for Vista Office EMR. Thus, this version is for physician offices.

---

[130] WorldVistA, Welcome to the WorldVistA homepage, at http://worldvista.org (last visited Mar. 1, 2008). At the request of the Federal agency supporting Medicare, a third-party analyst reviewed the WorldVistA version after a period of trial use at several clinics. Sujansky & Associates, *An Evaluation of VistA-Office EHR in the Small Practice Setting: Functional Performance, Economic Costs, and Implementation/Support Processes Final Report Submitted to The Iowa Foundation for Medical Care and The Centers for Medicare and Medicaid Services* (Nov. 30, 2006), http://www.sujansky.com/docs/VistaOfficeEHR_EvaluationReport_2006-11-30.pdf [hereinafter Sujansky, WorldVistA Evaluation].

[131] Gina Shaw, Vista EHR: right product, right price?, American College of Physicians Observer (Sept. 2005), http://www.acponline.org/clinical_information/journals_publications/acp_internist/sep05/vista_ehr.htm (noting that "The CMS' offer sounds exciting—but important service questions remain unresolved").

[132] On shrink-wrap licenses generally, see Michael J. Madison, *Legal-Ware: Contract and Copyright in the Digital Age*, 67 FORDHAM L. REV. 1025, 1054-70 (1998).

[133] WorldVistA, About WorldVistA, at http://worldvista.org/WorldVistA (last visited Mar. 1, 2008).

[134] The WorldVistA announcement includes:

WorldVistA's mission includes developing and supporting a list of service vendors who are available to help health care providers install and implement VistA EHR VOE. Because WorldVistA's versions of the VA VistA code have been supplemented and revised in certain ways for non-governmental use, part of WorldVistA's task is to evaluate the many dozens of source code changes issued by the VA each month.[135] Those that are applicable are incorporated into the WorldVistA version. In other words, there is technical effort required to keep the WorldVistA software current with the VA's system. This also allows direct subsidization: improvements funded by the VA are made available to a theoretically much larger user base. The ultimate goal for WorldVistA is to create a viable and growing community of technologists that collaboratively invest in the WorldVistA version over time. The model posits that the service vendors would be integral to the community as an investment in the ecology of a FOSS implementation of VistA. The approach assumes that employees of the ultimate end-users, the health care providers that work in a physician office, will typically not be involved in the technologist community. This would only occur in the infrequent case of a physician or other health care worker who has a strong information technology background or self-trained aptitude.[136] To promote the community, and in effect to lock the community into a free software model, WorldVistA applies the GPL to the code posted on the internet.[137]

At the time of this Article, the WorldVistA project is still in the early stages, which cautions drawing conclusions from its activity thus far. The VA's VistA software, even though created for a hospital environment, is applicable to many physician practices. Some physician specialists, however, may conclude that the WorldVistA version is inapplicable for their needs. In addition, the limited availability of interfaces to

---

> January 31, 2008 - WorldVistA announces the release and availability of WorldVistA EHR VOE/ 1.0, the only open source EHR that meets Certification Commission for Healthcare Information Technology (CCHIT℠) ambulatory electronic health record (EHR) criteria for 2006. WorldVistA EHR VOE/ 1.0 is based on and compatible with the U.S. Department of Veterans Affairs (VA) world renowned EHR, VistA®

WorldVistA, Welcome to the WorldVistA homepage, at http://worldvista.org (last visited Mar. 1, 2008).

[135] *See* VistA Notification, http://www.mcenter.com:8080/vns/signin.jsp (showing logon screen to access the a third-party software patch notification system that distributes the VA's VistA software patches).

[136] Conn, *supra* note 121 (noting that during VistA developers included "more than a few geek docs who combine their clinical knowledge with programming expertise.").

[137] Sourceforge.net, WorldVistA, http://sourceforge.net/projects/worldvista (product page indicating use of GPL license). To meet certain certification requirements, however, a services agreement needs to be associated with the software license. *See* WorldVistA EHR, http://worldvista.org/World_VistA_EHR; WorldVistA, License and Readme, http://worldvista.org/World_VistA_EHR/license-and-readme. The services agreement acknowledges that some of the WorldVistA version is public domain, but claims copyright and asserts GPL licensing in other parts of the code that are particularly tied to the certification requirement; it further prohibits claiming certification if the WorldVistA supplied code is changed. *See* WorldVistA, Master Services Agreement, http://worldvista.org/World_VistA_EHR/license-and-readme/WorldVistA%20EHR%20GPL%20License.txt.

practice management software and other systems impedes deployment. Nonetheless, the WorldVistA version is a notable experiment in government support of FOSS and a rather direct example of technology transfer from the government to the private sector. It also stands in contrast to other visible VistA FOSS activity, such as that of Medsphere and its version of VistA.

2. *Medsphere's OpenVista Product*

While several companies and institutions around the world have taken the VistA FOIA code as a starting point to adopt the system,[138] a recent entrant, Medsphere, has self-proclaimed its open source approach to VistA deployment. Medsphere is a venture-backed company specifically formed to leverage the VistA software. It markets its software under the brand OpenVista. Its competitive advantage as compared to the proprietary software suppliers is the cost subsidization inherent with starting from FOIA VistA.[139]

Like many suppliers of enterprise software, Medsphere's business model is to sell the software system. It might negotiate a price to deliver the software and install it, but its pricing model also includes service subscription payments over time. In Medsphere's approach, the allusion to the open source approach within FOSS is descriptive of only some of the code it supplies. Some modules or components are derived from FOIA VistA and Medsphere deploys them under the GPL. Other components are proprietary. Under this approach, Medsphere customers have a more open code base to diminish vendor lock-in to a significant degree.[140] However, they do not fully benefit from the anti-lock-in effect of free software.[141] The opportunity for this bundling arises from the need to supplement the VA's VistA code for commercial health care providers and from its own efforts to modernize the user interface in some areas of

---

[138] VistA Adopters Worldwide, http://www.hardhats.org/adopters/vista_adopters.html.

[139] Medsphere, White Paper: VistA-Office EHR: Diffusing Healthcare IT to the Ambulatory Market 4-5 (on file with author) ("[Medsphere's] OpenVista contains the same features, functionality, scalability, and reliability of core VistA but with the necessary modifications for the private sector. Additionally, Medsphere has aggressively pursued advancing the technology at each layer of the stack and has added value by providing greater choice of technology components."); Heather Havenstein, *Medical software from feds could benefit big health care: Low-cost app for small practices could aid efforts to computerize records*, Computerworld, http://www.computerworld.com/industrytopics/healthcare/story/0,10801,103738,00.html (Aug. 8, 2005) (discussing Medsphere's involvement in the market).

[140] The label "vendor lock-in" describes the disincentives a company has to switch to an alternative technology, which include switching costs and network effects of the installed technology. *See e.g.* Charles Ferguson, *How Linux Could Overthrow Microsoft: The Open-Source Movement Is the Largest Threat the Software Giant Has Ever Faced. Does Bill Gates Have a Plan*?, TECH. REV., June 2005, at 69, *available at* 2005 WLNR 8789992.

[141] On the anti-vendor-lock-in benefits of FOSS, see Vetter, *Exit & Voice in FOSS*, supra note 11. *See also* Ferguson, *supra* note 140 (positing that open source "severely limits the possibility of propriety 'lock-in'—where users become hostage to the software vendors whose products they buy—and therefore eliminate incentives for vendors to employ the many tricks they traditionally use on each other and on their customers")

the software. Medsphere offers its OpenVista software product for both physician clinics and hospitals, but most of its reported activity has been for hospital installations.

Like WorldVistA, Medsphere is harvesting, supplementing, and deploying the FOIA VistA source code. Medsphere relies on itself primarily, although it also promotes its desire to foster community development around the software it places under the GPL. WorldVistA reverses the roles, which is understandable since it is a non-profit entity. It needs vendors and community members to be involved to a greater degree than needed by Medsphere. Medsphere can fund programmers through the revenues it achieves by price undercutting the proprietary software vendors. WorldVistA has to generate funding as a non-profit to support the facilitative activities it seeks to implement.

These two approaches bring perspectives from the free software and open source software camp, respectively. The origination of the software from some non-FOSS source has occurred before, such as when Netscape converted its browser to FOSS to generate Mozilla, or when IBM did the same for its Eclipse software.[142] There are, of course, numerous influences on both WorldVistA and Medsphere's approach. These influences arise from the nature of the EMR software market and the difficulties inherent in automating business processes that involve clinical information.

VI. CHARACTERISTICS OF FOSS-DISFAVORING SOFTWARE MARKETS

The FOSS incarnations of VistA raise a question within the greater inquiry to describe the characteristics of software markets that might disfavor FOSS: would there be any significant FOSS development in the EMR market without the donation of the baseline system from the VA?[143] This Part will examine each suggested FOSS-disfavoring characteristics in light of both sides of that question. On one hand, the question is whether FOSS systems would have developed to the same degree without such a large subsidized input. On the other is whether the VistA FOSS approaches, or FOSS EMRs generally, can overcome any impedance arising from such characteristics. This second question leads to Part VII's discussion of facilitators for FOSS-disfavoring

---

[142] *See* Jim Hamerly & Tom Paquin, *Freeing the Source: The Story of Mozilla*, *in* OPEN SOURCES, *supra* note 2, at 197, 203-06 (describing the events leading up to Netscape's decision to release the source code for its web browser, Mozilla); Eclipse, About the Eclipse Foundation, http://www.eclipse.org/org (last visited June 10, 2008) ("Eclipse is an open source community, whose projects are focused on building an open development platform . . . The Eclipse Project was originally created by IBM in November 2001 . . . . The Eclipse Foundation was created in January 2004 as an independent not-for-profit corporation to act as the steward of the Eclipse community.").

[143] The answer to whether there has been significant FOSS penetration in the EMR market depends on what counts as significant, but one assessment counts it as minimal for the approximately dozen products functionally assessed. *See* Sujansky & Associates, Appendix E, *supra* note 14, at 2 ("many medical practices have now availed themselves of [the FOSS option, but] these practices remain very much in the minority among health care organizations that have adopted clinical information systems").

markets. Finally, this Part will generalize each characteristic outside of the specific context of the EMR software market.

There is a market characteristic that some might expect to appear on the list as FOSS-disfavoring that I do not include: software markets where privacy and data protection are important.[144] These are clearly valid issues in healthcare. The omission is because the expectation of inclusion rests on a faulty premise – that FOSS has some inherent disadvantage in this area.[145] Effective privacy and data protection rest on information technology practices and procedures that are applicable to both FOSS and proprietary software. Just because a company uses FOSS software to automate some part of the enterprise doesn't mean that users, or even most of the technologists, have access to the source code, and therefore have some theoretically greater capability to extract data from the software. If access to the source code is removed, the technological disposition is the same as compared to proprietary software. A few technologists having access to source code that they would not see under proprietary software is not a significant difference warranting inclusion.[146] The misconception that privacy and data protection issues are substantially more troubling with FOSS is perhaps a barrier to FOSS in health care, but if it is a barrier, it is a misinformation barrier, not a structural characteristic.

The characteristics discussed below draw from the description in Section II.A above of the technological features that help to define a software market. The discussion will also go beyond this technical focus as becomes necessary. The perspective of the discussion is the impedance arising from the characteristics as they aggregate. That is, how do they diminish the likelihood of FOSS adoption in a software market dominated by proprietary vendors? This type of analysis naturally leads to an assessment of what changes may need to take place in the supply-side dynamics of FOSS to overcome the impedance.[147]

---

[144] Health care is clearly a market with privacy and data security concerns. *See supra* text accompanying notes 58-61.

[145] Peter P. Swire, *A Theory of Disclosure for Security and Competitive Reasons: Open Source, Proprietary Software, and Government Systems,* 42 HOUS. L. REV. 1333, 1335-36 (2006).

[146] That small group of technologists does not raise the risk profile for a data disclosure when any regular user of the software is able to access and disclose the software's data in the normal mode of use. Moreover, the FOSS enterprise scenario is no different from software developed in-house on this issue.

[147] One analyst renders a specific list of factors limiting FOSS in the EMR market from the perspective of the existing vendors and FOSS products, and prescribes actions that might help the FOSS effort to grow generally. Sujansky & Associates, Appendix E at 2, *supra* note 14 (limiting factors include provider acceptance, scarce vendor support, duplication of effort among vendors, and lack of access to critical proprietary resources such as medical code and terminology databases; recommendations include public development of alternatives to those proprietary resources, and greater vendor collaboration to reduce duplication of effort and disproportionate costs bearing on early FOSS adopters).

A. *Low Technical Aptitude*

When software users have low interest or aptitude in programming, configuration, integration and installation, this signals a potentially FOSS-disfavoring market. While there are notable exceptions, such as the Firefox browser, most FOSS is by technological users for technological users. The examples illustrating this point are too numerous to discuss, given that one popular Internet repository of FOSS has over 180,000 registered projects.[148] The inclination of FOSS to evolve in ways amenable to technologists is a point sometimes cited to explain why the GNU/Linux operating system has not significantly penetrated the desktop computing market. One common explanation for FOSS development is that a programmer wants to "scratch an itch." The software developed to solve whatever problem represents the "itch" need only be operable by the programmer (or any peers with whom she wants to share it), thus reducing incentives to spend extra programming time to create a user interface amenable to the novice.

Many FOSS projects offer software components, not complete products, requiring technical skill for their use. This is particularly true for small, hobbyist projects. Some of these projects offer valuable functionality, but for an organization to deploy the software, the organization must have its own software integration capabilities, or be able to confidently contract with third-parties for such services. Neither avenue is simple, and there are life-cycle technology management implications that may not be apparent on the front end. While hospitals sometimes have one or both capabilities, most physician groups are ill-equipped to do either.[149]

Thus, low technical interest or aptitude can translate into feeble technological procurement skills for the organization, and translate into diminished opportunities for stealth FOSS installations that help build a FOSS user base. Many organizations discovered that they were running FOSS without information technology management knowing about it. This occurs because the engineers and programmers can easily find it on the Internet, and easily take it to solve problems as they program internal systems, or, worse, program software products for resale. These stealth installations assisted the growth of FOSS, even though they were unsavory from the corporate perspective. In addition, low technical interest or aptitude predisposes an organization towards acceptance of non-computing substitutes, such as paper-based business processes.

In the case of the EMR software market, the technical inaptitude characteristic is a factor for both physician offices and hospitals, although hospitals are more likely to have computing aptitude. While the VistA FOSS incarnations offer beneficial functionality, they carry the challenge of a user interface and internal structure that is

---

[148] The popular Sourceforge FOSS repository in the Summer of 2008 listed over one hundred and eighty thousand projects. Sourceforge.net, http://sourceforge.net/index.php, (last visited Jun. 16, 2008) ("Registered Projects: 180,405 Registered Users: 1,875,494").

[149] Rosoff, *supra* note 17, at 143-44.

based on older technologies.[150] These cut against adoption by technically inapt users. Moreover, this diminishes FOSS interest among programmers helping providers because often the desire to work on FOSS is due to the opportunity to work with the newest software technology.[151]

B. *High Work Flow Differentiation*

If software is going to be applicable to many user organizations, then it needs to be configurable to varying workflows, and buyers often desire that the configuration be achievable by a non-technologist user.[152] Otherwise, each organization must reprogram the software (if it can) to fit its workflow. Workflow requirements can change almost anything in the user interface of a business automation software package, including: the order in which fields must appear on screens; the sequence of successive screens, dialog boxes, or other user interface prompts; what users are allowed to do at various times or steps in a sequence; or how data manipulations sequence across a transaction. Many proprietary enterprise software vendors design user-administrable workflow reconfiguration capabilities into the software, and the relative strength of these are an advantage for some vendors.[153] Sometimes workflow issues relate to the computing devices intended for use, especially in cases where mobile computing is part of the enterprise software system.

If workflow re-configurability is necessary for success in the EMR software market, this disfavors FOSS because this capability requires substantial additional investment in the software.[154] As mentioned under the technical inaptitude characteristic, channeling energy to the user interface of FOSS projects tends to push against the typical inertia of FOSS development. Of course, if the proprietary software competitors don't provide re-configurability, FOSS might have an advantage because it can at least be reprogrammed by the health care provider to adopt it to her workflow, provided she can find a contractor who can do so, and provided that the contractor's design does not cut off the opportunity to take future updates for the FOSS product, which she may want. In other words, "one of" customized versions of software products have numerous life cycle feasibility issues, whether FOSS or proprietary. When a proprietary software vendor offers workflow re-configurability as a standard product feature, the typical expectation is that the users will be able to upgrade to later-provided versions.

On the one hand, FOSS might seem to have the advantage for workflow re-configurability because each user could reprogram the software for perfect

---

150 Conn, *supra* note 121; Sujansky, WorldVistA Evaluation, *supra* note 130, at 3-4, 16.

151 *See* Vetter, *Exit & Voice in FOSS*, supra note 11, at 234-35.

152 Jean A. Adams et al., *Workflow Assessment and Redesign*, *in* IMPLEMENTING AN EHR, *supra* note 52, at 36-38.

153 Ed Scannell, *Tivoli Automates IT Processes*, InfoWorld, Vol. 27, Issue 21, 23, May 23, 2005.

154 Sujansky & Associates, Appendix E at 2, *supra* note 14 (discussing the disincentive on FOSS development teams to program capabilities for handling data in particular ways).

customization. On the other hand, this is a disadvantage because it diminishes the possibilities and incentives to channel the customizations back to the FOSS project. Unless there is a framework of design present in the software originally to allow for beneficial reapplication of workflow configurations, it is unlikely to happen. Such a framework is an extensive software engineering endeavor. If it is not designed into the structure of the software from the beginning, it is often difficult to achieve later. This also cuts against its presence in FOSS, due to FOSS's evolutionary, accretive (and often under-funded, volunteer-supported) development style.

C. *Minimal Complementary Effects*

The tendency for FOSS to originate from technologists for technological problems has enabled some of the most successful FOSS projects to succeed in their markets as platform technology supported by complementary effects. Software is layered technology in the first place, so in contrast to EMR software that exists primarily at the user interface level, software such as the Linux operating system kernel is at the core level. The success of the GNU/Linux operating system is at least in part due to the many hardware, FOSS, and proprietary software complementary technologies it engenders. In addition, the ecology that has developed around GNU/Linux provides vast opportunities for complementary services from the largest companies in computing, such as IBM, to sole proprietorships deploying or servicing systems based on GNU/Linux. These observations apply to many other platform software technologies, many of which have enabled much of the Internet's infrastructure.[155]

With minimal complements for EMR software, other than services associated with a software installation and perhaps hardware sold to run the software, there is little impetus for other technology companies to support or initiate a FOSS EMR software package.[156] The better game, from at least the short-term perspective of profit-oriented entities, is the proprietary software model, where development costs can be spread over a paying user base over time. This is why the VistA FOIA software is important: as a large government-supplied input, it makes services, support, hardware, and installation complements viable. Thus, companies such as Medsphere can compete without the upfront investment necessary to program all of the EMR software from scratch.

---

[155] The platform effect of FOSS spilled-over into the cell phone market. First, Google announced a Linux-kernel based mobile phone operating system. Google, Android - An Open Handset Alliance Project: What is Android?, Jun. 10, 2008, http://code.google.com/android/what-is-android.html. Then, Nokia, in a competitive response, open-sourced the "world's foremost smartphone platform" – the Symbian operating system. Eric Zeman, *Nokia, Others Deal Major Blow to Android*, Information Week Blog, Jun. 24, 2008, http://www.informationweek.com/blog/main/archives/2008/06/nokia_others_de.html.

[156] Some FOSS EMR software vendors, however, have found success with a services and support complements business model. *See* Sujansky & Associates, Appendix C at 5, *supra* note 14 ("The main source of revenue for ClearHealth comes from the support services that it provides to its commercial customers. Clearhealth offers a full set of such services on a contracted basis, including installation, configuration, customization, maintenance, and support.").

Other types of complements are technologically plausible but practically impossible in some markets. For a workable example, consider Google's core business. Google uses a significant cost subsidy from FOSS, a no-cost operating system kernel (Linux) implemented across many thousands of computers, to provide search services funded by the complementary advertising revenue. Consider, however, Google's planned foray into a retail electronic health record.[157] The market quickly gave numerous signals that it would frown upon advertising associated with the system.[158]

The EMR market has little opportunity for software complements "above" the EMR software, but it is significant that the support "below" the software could be FOSS. In other words, a FOSS EMR software package is complementary to the Linux kernel if running on the GNU/Linux operating system. It becomes yet another instance that might trigger a quantum of affiliation back to the Linux kernel.[159] Such affiliations can have impact in the aggregate, especially in the general information technology industry where platforms compete.

D. *Dispassionate Computing Agendas*

The free software strand of the FOSS movement originated from passionate views about a person's right to functional freedom with her computer. The political message behind this view buoys the FOSS movement to this day. The message amplifies when directed toward large proprietary software providers, most notably Microsoft. Thus, anti-Microsoft passion is sometimes a manifestation of anti-proprietary-software principles. These principles motivate many individual FOSS programmers who contribute to projects. The open source camp feels these views less strongly, but fundamentally prefers source code transparency and thus remains motivated to work against the pure proprietary software model. These energetically felt views seem to center within the information technology industries, although they can be found anywhere.[160]

Health care, on the other hand, from the perspective of computing, is one of many industries with a pragmatic view of information technology. Providers view the technology as a necessary business asset, and consider proprietary software to be an

---

[157] Google Health, www.google.com/health. The Google offering is best classified as a Personal Health Record (PHR) or Continuity of Care Record rather than as a part of an EMR system. *See* About Google Health, https://www.google.com/health/html/about.

[158] *See* Travis Reed, *Google Tries to Calm Fears Over Privacy of Health Service: The New Project Will Be Free of Ads, No Data Shared Without Prior Consent*, San Jose Mercury News, Feb. 28, 2008; *but see* U.S. Patent Application No. 20070282632, Method and apparatus for serving advertisements in an electronic medical record system (filed Dec. 6, 2007).

[159] For a health care provider, the affiliation won't be a programming suggestion for the Linux kernel, or even a reported bug, but might be as simple as increasing the quantity of computers Dell, for example, sold that month with GNU/Linux preinstalled. This quantity, in the aggregate, might increase Dell's attention to GNU/Linux.

[160] *See* Vetter, *Exit & Voice in FOSS*, supra note 11.

available means to various ends. It is a tool to help provide health care services, and be paid for those services.[161] FOSS alternatives, when they are available, present user organizations with a different set of pros and cons for adoption and life cycle ownership.[162] A passionate perspective would look beyond the current pros and cons to FOSS principles suggesting that users might help themselves by adopting and contributing to FOSS, trading short-term challenges for long-term advantages of software supported by a viable community.[163] Some users in some industries take the active perspective, but the message encounters resistance in organizations that shy from risk in operational matters unless the FOSS at issue is extremely well-proven.

It is significant to note that in the EMR software market, cost reduction is a primary influence behind the FOSS incarnations of VistA, an already-proven software system.[164] This shows the pragmatic, cost-weary perspective of an industry that is generally disinclined to aggressively invest in information technology change for various structural reasons. Moreover, when investments occur, FOSS is often viewed as risky in the EMR market compared to the numerous mature proprietary software products. Organizational procurement officers tend to be risk adverse by default, so the novel FOSS value proposition often dims in comparison to the puffery, promise, and performance of established software suppliers.

E. *Entrenched Proprietary Competitors*

The makeup of the proprietary competition might influence whether users take a dispassionate approach and will determine the degree of difficulty for users to switch to FOSS alternatives in a software market. In the fragmented EMR market, the search and evaluation process to select software can be very time consuming, in part because there are so many products to choose from.[165] However, even though the cost of switching between proprietary vendors can be high, users may feel some comfort in the fact that numerous alternative products exist if their relationship with their existing EMR software vender sours. Contrast this with a software market with a single, dominant monopolist provider, such as Microsoft for general purpose operating systems. Within information technology, this stirs passions toward anti-Microsoft action, which benefits FOSS and leads to FOSS adoptions and contributed effort.[166]

---

[161] Bernstein, *supra* note 26, at 18.

[162] *See* Vetter, *Exit & Voice in FOSS*, supra note 11, at 226-33; Scott Wilson, et al., *Open Source Salvation or Suicide?*, Harvard Business Review, Vol. 86 Issue 4, 40-44 (Apr. 2008).

[163] Moglen, *supra* note 18 (analogizing free software to a commons, and remarking that "If you've become dependent on a commons for whatever role in your business, then what you need is commons management.").

[164] Sujansky, WorldVistA Evaluation, *supra* note 130, at 23.

[165] Frank Richards, *Vendor Selection and Contract Negotiation*, *in* IMPLEMENTING AN EHR, *supra* note 52, at 15-18; AMATAYAKUL, *supra* note 15, at 253-54.

[166] *See* Vetter, *Exit & Voice in FOSS*, supra note 11, at 258-62.

The fragmented EMR software market signals that product tailoring has some users satisfied with their software, and also makes it more difficult for a FOSS alternative to facilitate the switch. The data conversion process would need to support numerous starting points from the various proprietary vendors in the EMR market. Consider a counter-example from the word processing software market. The FOSS alternatives in that space need only provide conversion utilities or interoperability with Microsoft's Word product in order to target most of the market. The word processing software market would require completely automatic conversion. Most users wouldn't switch to FOSS alternatives unless their existing documents could be readily converted without issue. The EMR software market, like most enterprise software applications, will require a technologist to extract, convert, and import the data into the FOSS EMR software system. While such data repurposing projects are common in enterprise information technology, the degree of risk and difficulty will depend on technological details about the source, the proprietary EMR product, and the destination—the FOSS EMR software.

The other issue related to the entrenched competition characteristic is whether the proprietary software vendors are organized to oppose FOSS. If the market is dominated by a small number of suppliers, it will be easier for them to overcome collective action problems and strategically maneuver within the limits of antitrust law. Thus, Microsoft, which has no such collective action issues, receives much attention for its strategic maneuverings with respect to FOSS. Even fragmented markets, however, need only a few vendors willing to exert pressure at various points, such as notable trade associations[167] or organizations that set standards.[168] The standard-setting pressure is a factor in the ongoing certification of the VistA FOSS implementations for physician

---

[167] For the software industry generally, the Business Software Alliance has advocated the merits of proprietary software in relation to FOSS. Business Software Alliance, letter to United Nations Development Programme: Asia-Pacific Development Information Programme, concerning International Open Source Network E-Primers, http://www.iosn.net/publications/foss-primers/bsa-response/Letter_to_IOSN__final_-_with_letterhead_and_attachments_-_reduce_.pdf. Within the EMR market, the American Academy of Family Physicians decided to repurpose an abandoned commercial EMR software product as FOSS, but this effort later dissolved into an approved short list of proprietary EMR vendors with AAFP secured price discounts. *See* Daniel L. Johnson, AAFP EHR project summary, openhealth-list, Apr. 1, 2003, http://www.mail-archive.com/openhealth-list@minoru-development.com/msg08214.html (describing open source EMR project by the AAFP); Eric G. Brown, Forrester Research, An Open Source EMR For Real, March 28, 2003, http://www.forrester.com/ER/Research/Brief/Excerpt/0,1317,16535,00.html ("The American Academy of Family Physicians (AAFP) is spearheading an open source electronic medical record"); The Health Care Blog, Open Source EMRs, the AAFP, and CMS grants gone awry? (with apologies to George Lucas), Jun. 15, 2004, http://www.thehealthcareblog.com/the_health_care_blog/2004/06/technology_open.html (describing AAFP's transition from promoting a specific open source EMR product to promoting EMR software products, mostly proprietary, that support open standards).

[168] Within the EMR market there is an organization certifying software products for interopreabiltiy. CCHIT, Certification Commission for Healthcare Information Technology (CCHIT), at http://www.cchit.org/about/overview.htm; Hoffman & Podgurski, *Finding a Cure*, *supra* note 12, at 24-25.

office setting.[169] The certification requirement is one example of the seemingly inexhaustible regulatory provisions bearing on health care.

F. *Regulatory and Bureaucratic Pressures*

External forces arising from regulatory and bureaucratic sources might create direct and indirect inertia against FOSS.[170] In the EMR software market, and in health care generally, positive law may chill provider collaboration on FOSS software because giving something of value to another provider is often a regulated action. This regulatory regime is carried out by the federal fraud and abuse laws for health care providers. The applicable details will be discussed in Section VII.B below, which will also discuss the related possibility that safety-regulating law might chill collaborative development or tinkering with EMR or health care software.

Operating within a regulated industry, health care providers encounter numerous public and private regimes that impose internal operating costs. These bureaucratic forces include private payers such as private insurance companies and public payers such as Medicare. State insurance regulatory agencies are also involved with the private payers, while federal agencies are involved with Medicare. The requirements for receiving payment increasingly require health care providers to meet technical and administrative specifications - not only in the medical billing software but also in the EMR software that supports the billing system.[171] This sometimes makes the provider dependent on software updates that implement new regulatory requirements in the EMR or billing system.

All these influences add complexity to the provider's operations and her EMR software, either raising software production costs, or raising other operational costs and thus starving capital investment in EMR software.[172] Since much FOSS is developed through volunteer or contributed effort, a regime that raises the implementation effort diminishes FOSS viability.[173] This is particularly true because most providers would prefer to install an EMR system just once, meaning that they might not prefer the evolutionary style of FOSS development. Proprietary software vendors typically offer

---

[169] FOSS EMR software products must pay attention to additional contractual and technical issues for certification due to the open nature of the technology. *See supra* note 137.

[170] Dana Blankenhorn, *What is stalling open source in healthcare?*, Open Source, ZDNet.com, Aug. 2nd, 2007, http://blogs.zdnet.com/open-source/?p=1272&tag=btxcsim (noting that proprietary advantage and bureaucracy were stalling FOSS adoption in health care).

[171] *See e.g.* Anne Zieger, CMS now says NPI must match IRS data, Jun. 18, 2008, http://www.fiercehealthfinance.com/story/cms-now-says-npi-must-match-irs-data/2008-06-18 (last visited Jun. 25, 2008).

[172] Hoffman & Podgurski, *Finding a Cure*, *supra* note 12, at 20-22.

[173] Sujansky & Associates, Appendix E at 2, *supra* note 14 ("[O]n the whole collecting clinical data in a coded form amenable to analysis and decision support is not among the development priorities of the FOSS EHR projects. This is likely the case because support for coded data entry can add significant complexity to an EHR application (relative to free-text entry) and can slow the clinician workflow.").

software modules to meet all of the provider's needs. Even with a large baseline of functionality, the VistA FOSS implementations - for either the institutional or the physician office setting - needed additional development to be viable.

In light of this tentative list of FOSS disfavoring characteristics, the next Part discusses potential changes for the supply-side dynamics of FOSS to overcome the impedance that might arise from these characteristics.

VII. FACILITATORS FOR FOSS-DISFAVORING SOFTWARE MARKETS

The discussion in this Part evaluates a number of approaches to overcome the impedance from FOSS-disfavoring characteristics. Section VII.A will deal specifically with FOSS facilitators in the EMR software market, while Sections VII.B-D will deal with FOSS facilitators in software markets generally as well.

A. *Prospects for FOSS in the Growing EMR Software Market*

Given the characteristics discussed in VI above, the EMR market would seem to be one in which a community-grown FOSS would be unlikely to develop from scratch. With a subsidy such as the VistA source code base, however, even though it is composed of older technologies, the FOSS dynamics change. At least in the institutional setting, one company, Medsphere, has achieved some success with an implementation model using some open source components.[174]

A counterweight to the pessimism, however, might be the passionate political message of free software.[175] This point acknowledges the possibility that the passage of time with greater promotion of free software principles might convince a critical mass of health care providers to direct their contractors to contribute to FOSS or encourage their own technologists to do so.

Between the free software and open source software camps, free software principles might be more likely to take hold in health care because the technology development emphasis of the open source camp does not resonate as well. Health care providers are not technology developers. They are technology users. As users, the free software principles expressing organizational self-determination for computing might be attractive. Moreover, health care has always had volunteer elements within its delivery system. The volunteer and non-profit heritage of health care, particularly at the institutional level, resonates with the volunteerism that underlies many FOSS projects.

If more evangelism of the free software message would bring the health care information technology decision makers to an eventual embrace of FOSS, this

---

[174] Another prominent vendor for VistA-based implementations in the institutional setting is DSS. *See* DSS Inc., VistA experts, http://www.DSSinc.com.

[175] Fred Trotter, GPL Medicine - The morality of licensing in medicine, http://www.gplmedicine.org/index.php?module=htmlpages&func=display&pid=3.

opportunity may be hastened away by the policy emphasis within the U.S. government to accelerate EMR software adoption, particularly in the physician office setting. Enacted measures include allowing hospitals to subsidize the EMR software costs of its attending physicians.[176] Proposals include mandating use, subsidizing use, and adjusting Medicare payments for physicians who adopt interoperable EMR software packages. With a minimal number of viable FOSS EMR offerings, federal pressure to accelerate EMR adoption in the physician office setting is most likely to drive more providers to the well-established proprietary software vendors.

The EMR software market is growing.[177] The technologies provided by the proprietary venders are improving and computing hardware costs continue to decrease, creating greater affordability. The federal pressure to adopt EMRs might further accelerate the growth. In addition, as new physicians join the ranks, they are more likely to have confidence in computers and software, helping adoptions in the physician office setting. Many new physicians likely used the VistA EMR software in training rotations through a VA hospital or have used another EMR in medical school. The question is whether this anticipated growth will include any significant increase in FOSS EMR installations as a percentage of the market.

Free software evangelism might help the VistA FOSS offerings gain a toehold, but, even with this reinforcing effect, an artificial growth pressure seems foreboding for FOSS in the EMR software market. If the market saturates over the next decade with minimal FOSS penetration, dislodging the proprietary software model will be even more difficult. Business process automation software, such as the EMR, becomes the electronic nervous system of an organization, rendering a swap to a different technology provider a serious - and often cost-prohibitive - matter.

B. *Safe Harbors for Anti-Collaboration and Anti-Tinkering Law*

Regulation can influence how participants in a market collaborate or whether approved or accredited technology can be reprogrammed in the field. In health care the former manifests itself in the fraud and abuse laws. The latter is best exemplified by FDA approval of medical devices.

The FDA device approval regime does not presently extend to EMR software.[178] This is in part because traditionally EMR software does not take action; it merely

---

[176] *See* United States Department of Health & Human Services, New Regulations to Facilitate Adoption of Health Information Technology (Aug. 1, 2006) available at http://www.hhs.gov/news/press/2006pres/20060801.html; Hoffman & Podgurski, *Finding a Cure*, *supra* note 12, at 21.

[177] Tyler Chin, *Small practices fuel sales of EMR systems*, amednews.com, Feb. 9, 2004, http://www.ama-assn.org/amednews/2004/02/09/bil20209.htm (noting that "Pressure from payers and a growing interest by physicians have analysts expecting large growth in electronic medical record sales to small groups.").

[178] Hoffman & Podgurski, *Finding a Cure*, *supra* note 12, at 26-28.

provides data that the health care providers use to provide care. If the FDA or some similar safety-focused regulatory approach covered EMR software, this anti-tinkering influence would need a safe harbor to accommodate FOSS development.[179]

The federal fraud and abuse laws, particularly the anti-kickback prohibitions and the Stark anti-referral provisions for designated health services, may already provide a present chill for some collaborative FOSS development. Both anti-kickback and Stark share an operative principle: referrals from one provider to another should be uninfluenced by financial entanglements.[180] The two regimes have numerous differences; notably, each takes a different approach toward regulating invalid referrals. Anti-kickback disallows referrals in exchange for something of value.[181] Stark prohibits referrals for defined activity, called "designated health services," when there is a financial entanglement.[182] The effect of both is that providers are (hopefully) very careful about their transactional and structural relations with other providers.

These regimes could become an issue in the FOSS context for small projects where value transfer could easily be traced from a first provider to a second, that is, where the second provider refers patients to the first provider. Whether this actually fits into either regime is not to be fully proven in this Article. The point is that the regulations present the possibility of chilling FOSS development.[183] For example, suppose Doctor Orange is an orthopedist who writes software as a hobby and contributes to a FOSS medical imaging software project. The software augments medical images with embedded links that allow the user to click to the patient's demographics. Assume that state law requires medical images transferred among providers to have this capability. Five other physicians in the same state use the software, one of which is Doctor Frank, a family practice physician who regularly refers difficult orthopedic cases to Dr. Orange. Because the project has a small number of users, there is an argument that Dr. Orange tailored the valuable code for Dr. Frank's uses. Clearly, the programming and updating of the software is something of value that has been transferred from Dr. Orange to Dr. Frank. If this example is close to an anti-kickback or Stark issue,

---

[179] A safe harbor for tinkering with approved software might need a process of regulatory approval for the changes, perhaps similar to the role Hoffman and Podgurski propose for EHR System Oversight Committees (SOCs), for EHR system regulation. Hoffman & Podgurski, *Finding a Cure*, *supra* note 12, at 34-38.

[180] FUNDAMENTALS, *supra* note 74, at 117-119, 138-40.

[181] FUNDAMENTALS, *supra* note 74, at 117-119.

[182] FUNDAMENTALS, *supra* note 74, at 138-40.

[183] Chilling from the potential value of code contributed to a project would parallel other issues that have arisen over time with technology donations from one provider to another. Paul F. Danello, Preparing for Interoperability: EHRs and the Law, Health Management Technology, (Sept. 2006) http://archive.healthmgttech.com/archives/0906/0906preparing_interoperability.htm. For a discussion of another aspect of the fraud and abuse regime bearing on health care providers, beyond Stark and anti-kickback, see Richard S. Saver, *Squandering the Gain: Gainsharing and the Continuing Dilemma of Physician Financial Incentives*, 98 NORTHWESTERN UNIV. L. REV. 145, 154-66, 171-72 (2003).

presumably a much larger user base dilutes the issue. In other words, if hundreds of doctors use the software, the targeting is diluted.

Fashioning a Stark/anti-kickback safe harbor for FOSS development would not be difficult in theory, and seems justified by the policy promise of FOSS. The collaborative development approach in FOSS would have providers, their personnel, or contractors under their direction, contribute to FOSS projects. If any of these potential contributors feel chilled by these two regimes, a safe harbor approach should be implemented. Other anti-collaboration laws have similar mechanisms. For example, in antitrust law, standard setting organizations that use consensus processes can register for a remedy-reducing shield.[184] Scholars have likewise proposed that antitrust regulators should deemphasize enforcement when collaboration has the purpose of clearing intellectual property rights, such as patents, in a standard.[185]

While safe harbors might sometimes be necessary facilitators to change the motivational mix for FOSS in markets that disfavor it, the licensing models deployed in such markets will also be important to give FOSS a chance.

C. *Licensing*

New licensing schemes such as FOSS are often developed to revise and compete with existing licensing practices, seeking to impart a beneficial impact through their new terms. FOSS's copyleft licensing structure offered novel terms with the promise of greater social benefit than proprietary software. In the following three subsections, this Article will examine different means through which this promise can be brought to fruition in FOSS-disfavoring markets. First, because the FOSS movement developed based on copyleft and attribution-only licensing, many new licenses appeared that were essentially refinements of these approaches. This has led to a proliferation of licenses, some of which are more amenable than others to the close intermixing of FOSS and proprietary software. Second, a new macro-refinement of FOSS licensing known as the dual license was developed. Finally, an underappreciated approach is the opportunity for licensing schemes to channel contract programmers towards contributing their code to FOSS projects.

Each of these three subsections will focus on licensing practices that can be implemented within the framework of existing laws. While there are examples of

---

[184] See Vetter, *Scattering Opportunism*, *supra* note 24, at 120-30.

[185] Mark A. Lemley, *Intellectual Property Rights and Standard-Setting Organizations*, 90 CALIF. L. REV. 1889, 1937 (2002) (concluding that "that antitrust law should show great deference to legitimate efforts to set collective rules for dealing with IP, even if those rules require competitors to discuss both the technical merits of their products and the price of an IP license").

positive law revisions in the United States to accommodate FOSS,[186] the movement's progress has come primarily from putting the ideas into practice through the use of licenses. Additionally, these facilitators may have differing impacts for users adopting technology for the first time versus users switching from proprietary software to FOSS. Both possibilities are present in the EMR market, whose particulars will provide context to the discussion.

1. *Proprietary / FOSS Layering*

Software works in layers, via interconnected components in a hierarchy. Different FOSS licenses have differing degrees of acceptability for far or near coupling and intermixing of proprietary software with FOSS.[187] Many attribution-only licenses are highly permissive in this regard because their conditions allow most uses (even in proprietary software) as long as there is attribution. The original copyleft license, version two of the GPL, tends to repel close intermixing of GPL-licensed code because the other code, if arguably a derivative work of the GPL-licensed code, is at risk of a claim that it should be distributed under the GPL.[188] If the other code in question is proprietary software, its owners typically won't want to distribute it under the GPL.

To the extent that a FOSS license repels close intermixing with proprietary software, relaxing enforcement of this provision may allow a software supplier to bundle FOSS components with proprietary software and enhance its competitiveness.[189] The FOSS components could serve as a subsidized input for the supplier's total offering.[190] The proprietary components would allow the software supplier some control over the customer, but not with the full degree of lock-in classically leveraged in the proprietary model. If FOSS components are utilized in the software's underlying architecture, there is perhaps a greater chance of FOSS practices taking hold among other software suppliers in the market, general-purpose information technology contractors, or even among the

---

[186] As an example, there are revisions to state law as a result of the National Conference of Commissioners on Uniform State Laws ("NCCUSL") project which has generated the Uniform Computer Information Transactions Act ("UCITA"). *See* NCCUSL Web, *at* http://www.nccusl.org/nccusl/ (last visited July 6, 2008) (describing NCCUSL and providing links to UCITA project). Maryland's adoption of UCITA was with revisions to account for open-source software: "[n]o implied warranty of merchantability is given where a product is distributed for free unless the product is distributed in conjunction with some other sale or lease." Charles Shafer, *Scope of UCITA: Who and What are Affected?*, UNIFORM COMPUTER INFORMATION TRANSACTIONS ACT: A BROAD PERSPECTIVE 325, 248 (Practicing Law Institute Intellectual Property Course Handbook Series, vol. 672, 2001). Later, the NCCUSL UCITA committee recommended "a new section that exempts from implied warranty rules the transfer of a computer program where no contract fee is charged for the right to use, copy, modify or distribute the program." Report of UCITA Standby Committee, § 3(F) (2001) (Recommendation 10), *available at* http://www.nccusl.org/nccusl/UCITA-2001-comm-fin.htm (last visited Feb. 4, 2008).

[187] Vetter, *Infectious OSS*, *supra* note 109, at 88-94, 110-113.

[188] Vetter, *Infectious OSS*, *supra* note 109, at 88-94.

[189] Vetter, *Infectious OSS*, *supra* note 109, at 114-15.

[190] Sujansky & Associates, Appendix E at 1, 6, *supra* note 14.

users themselves.[191] One might argue that this would have the negative effect of confusing FOSS with proprietary software because the software supplier would still be charging for the proprietary software components, and it might not be apparent to the user what parts are FOSS and what parts are not. However, this problem already exits to a substantial degree with many for-profit software companies fully committed to FOSS because they charge service subscription fees that allow them to internally cross-subsidize their further software development. The difference is that the developed software is released under a FOSS license.

Bundling proprietary and FOSS layers into the same software system has the risk of diluting the FOSS message and diluting a market's commitment to FOSS - but this approach might be a second best alternative for FOSS-disfavoring markets. The composition of the bundle will matter in judging its efficacy. If the FOSS layers are so thin or trivial that the FOSS message is a shill, then the approach is not efficacious.[192] This might offend the sensibilities of rights-holders acquiescing to use of any strong copyleft-licensed FOSS in the bundle. To develop marketplace confidence, companies using the bundled layer approach should ensure and publicly state that its proprietary components are truly new investment, and are not merely harvested from the other type of FOSS, i.e., attribution-only licensed software.

In the EMR market, the bundling strategy is more transparent for institutional users who will likely have greater aptitude to comprehend which components are FOSS and which are not and the advantages attendant with each. The institutional users would also likely have a better sense of the cost of competing fully-proprietary alternatives, and would thus be better able to appreciate the value of the FOSS/proprietary bundle.

The bundled layer approach - ornamented with a large, but sensible, dose of marketing puffery about its embrace of open source software - is the Medsphere strategy.[193] It has developed and provides some components to the EMR FOSS community as GPL-licensed software. The core of its system is the VistA FOIA software, which, due to the regular revisions issuing from the VA, requires some investment to update for any particular customer or FOSS repository. At least in the EMR software market, riding atop the subsidy represented by the VistA software, bundling proprietary and FOSS software seems a potential facilitator.

---

[191] One commentator questioned one vendor's commitment to open source software approaches, describing that the vendor's actions did not, at least at one point in time, match its rhetoric. I. Valdes, *Is Medsphere an Open Source Company or Not?*, LinuxMedNews, Oct. 12, 2006, http://linuxmednews.com/1160704658/index_html.

[192] *See e.g.* Sujansky & Associates, Appendix D at 4, *supra* note 14 (noting that "Medsphere is in the process of working out its approach to the open source model").

[193] Sujansky & Associates, Appendix D at 4, *supra* note 14.

2. *Dual Licensing*

In contrast to bundling proprietary and FOSS layers or components, dual licensing offers a different approach to serving a multitude of interests among software suppliers, users, and any contractors or distributors that operate in between.

A typical approach to dual licensing operates as follows: if a distributor uses a FOSS license with her users, then the originating dual licensor provides the software under a FOSS license. On the other hand, if the distributor takes a non-FOSS approach, licensing only object code and charging royalties, the dual licensor applies traditional, royalty-bearing, proprietary software licensing terms. In essence, the dual licensor offers bifurcated terms, and the distributor-licensee chooses to operate on one side of the bifurcation or the other. The originating dual licensor, however, often provides for itself the ability to incorporate software revisions it finds on the open source side into the proprietary side.[194]

From the perspective of the dual licensor, a benefit is that it can in effect "harvest" code from the open source community and include the harvested code in the original software project for future licensing under either a FOSS model or propriety terms. The originator's permission to do this is in the original dual license. Under this structure, as soon as a FOSS licensee of the dual-licensed software distributes the code, the FOSS side of the dual license requires source code availability, and the dual license also allows the originator to incorporate the code into the master software project. The structural benefit of the dual license is that a partial commons created by a FOSS license is available to the originator for relicensing under commercial terms on the other side of the dual license - so long as the originator also makes the code available under the FOSS license. Under this approach, commercial use equates to proprietary software licensing, and the FOSS license applies for non-commercial use.[195] It is possible, however, that other concepts of "commercial use" could also act as the fulcrum of the dual license.

In the EMR market, as a subset of the health care market, much would depend on what constituted "commercial use." If the non-profit entities in the health care delivery system were deemed to be noncommercial users for purposes of dual licensing, it would allow FOSS community development for those installations. To the extent those software changes made their way back to for-profit providers, one benefit is greater

---

[194] *See generally* Heather Meeker, *Db4objects and the Dual Licensing Model*, 12 No. 3 CYBERSPACE LAW. 9, ___, (2007) (noting that there are "several flavors of dual licensing models" and that "[d]ual licensing is a business model where the licensor offers software under both commercial licensing terms (sometimes called 'proprietary' terms) and open source licensing terms."). Partitioning the market is sometimes the goal:

> [S]ometimes the product in the two different channels is different (with the commercial channel including extra features) and sometimes the two channels offer identical products - sometimes referred to as a "pure" dual licensing model.

Id.

[195] *See* Vetter, *Exit & Voice in FOSS*, supra note 11, 224-27.

standardization among that software vendor's users. On the other hand, the dual licensing approach typically doesn't allow for cross-pollinating code from one vendor to another via their FOSS-side installations. The nonprofit users may be the least likely to have technical personnel to revise the code, but might benefit the most from the contributions of other FOSS licensees.

It seems unlikely that a proprietary software vendor in the EMR market would suddenly make its entire product suite available under a dual license offering nonprofit providers FOSS use, but equally startling events have occurred in the history of FOSS. A proprietary vendor's reason to shift to dual licensing might include: increasing its user base, some of which might purchase services; promoting contributions by third-party programmers; and promoting general good will about its presence in the marketplace. Thus, dual licensing, like bundling, can support a marketing campaign where a proprietary vendor can appear progressive, even if internally reluctant. Both these licensing facilitators will have a greater impact if the contractors in the marketplace are also involved in the FOSS experience.

3. *Contractor Channeling*

Users in some software markets, such as the EMR software market, rely on third-party contractors for general support, technical acumen, software development when necessary, and software integration, configuration or customization. Sometimes the contractors are from a software supplier, but often they are not. The desire for local support, and support of varying expertise - often means that a user has multiple contractors.

If users could promote the fact that their software-writing contractors contribute any developed code back to FOSS projects, it might start a FOSS-supporting pattern in a market that lacks one. One way to facilitate this phenomenon would be the development of model contract clauses that users could insert into their agreements when engaging contractors. The clauses would be authored and structured so that they would allow the contractor to meet her obligations for the project at hand, but arrange an additional set of obligations to: (i) promote use of FOSS for the project if feasible; (ii) require contribution of authored software to an appropriate FOSS project (with assignments of copyright when necessary); and (iii) promote the fact that, so long as the contractor works for the user, she will involve herself in the FOSS project for the benefit of the user. The model clauses would also have to allow for different modes of FOSS licensing, such as attribution-only licenses or copyleft licenses like the GPL. The design principle for the clauses is to author them so that they will be capable of interoperating with most or all of the FOSS licenses used in the market.

Channeling contractors to support FOSS is an initial coercive step where users leverage their position over their contractors with the hope of initiating a habit of FOSS use and development in the software market. If model contract clauses make this easier for users to implement, and if their contractor costs do not increase too much (costs might in fact go down if FOSS inputs are discovered to reduce the programming time), this

approach could be beneficial. Its range of effect, however, depends on the degree to which end users commission the development of custom programming or integration in that market (which depends in part on the availability of standard software products). In the EMR software market, custom code development is likely infrequent given the number of software products available, particularly at the physician office level. As a result, other facilitators should also be considered.

D. *Other Facilitators*

In addition to removing barriers from anti-collaboration law and emphasizing certain licensing approaches, there may be several other policy approaches to facilitate FOSS for markets that disfavor it. Technologists who serve a market, such as contractors writing code for users, may want to hold themselves out as FOSS experts. Two of the approaches below relate to that need. The third, subsidies, has been exhibited through the government's support of FOSS activity in the EMR market.

1. *Service Markets*

If user-mandated channeling initiates a group of contractors to the experience of FOSS development, those contractors, as well as FOSS programmers generally, might want to market themselves with this differentiating expertise. In the proprietary software area, various certification programs by large vendors perform a similar role.[196] Thus, a registry of service firms, or some other mechanism to enable users to find FOSS-trained technologists in a market, might facilitate use of FOSS.

This has already occurred, to some extent, on one popular online FOSS repository. In early 2008, the Sourceforge web site, which houses over 170,000 FOSS projects, established the "SourceForge.net Marketplace," touting it as "the best place to buy support for your open source software."[197] Within the EMR software market there are two related comparable mechanisms. These are not outright service marketplaces, but are groups associated with the externalizing of the VistA FOIA code. One of these is the WorldVistA community, and the other is a group that has a long history with the VistA software, known as the Hardhats.[198] Neither, however, are market makers designed to match service buyers with service providers. The success or failure of the SourceForge.net Marketplace will give some indication whether it - or a similar

---

[196] *See* Microsoft Certifications Overview, http://www.microsoft.com/learning/mcp/default.mspx; CISCO, Cisco Career Certifications: IT Certification and Career Paths, http://www.cisco.com/web/learning/le3/learning_career_certifications_and_learning_paths_home.html (discussing three levels of CISCO certifications).

[197] SourceForge.net, Marketplace, at http://sourceforge.net/services/buy/index.php (last visited Mar. 5, 2008).

[198] WorldVistA, About WorldVistA, http://worldvista.org/WorldVistA; Hardhats.org, http://www.hardhats.org ("Welcome to a web site dedicated to fostering a virtual community for the worldwide users of the VISTA software!").

marketplace specially designed for a FOSS-disfavoring market - will have a facilitating impact.

2. *Active Attributions*

Another potential FOSS facilitator, which this Article will term "active attributions," presents a more glamorous - albeit far-flung - means for providing attribution in FOSS development projects. Many FOSS programmers are motivated by the attribution they receive for their work. This recognition comes from their peer technologists, in part because the attributions are typically recorded as comments in source code or as postings in a source code control system.[199] Thus, they are usually buried in the code where a regular user would never find them. Sometimes, programmers' names are also listed in the help files or show up on a "splash screen" that appears momentarily when the software starts. The term "active attributions" is a proposal for a methodology to show the programmer to the world, when they want to be seen. FOSS programmers would be able to opt into the system. It would allow regular users to learn about the humans who developed the FOSS code that the user is currently running.

Consider this hypothetical implementation from the EMR software market. A physician whose office uses a VistA FOSS implementation has just directed her contractor to install a health screening module the contractor discovered on SourceForge.net, an online repository of FOSS projects. This module operates in conjunction with VistA and pre-checks a patient's record before a visit so that it can suggest preventive care measures indicated by the data in the EMR. The active attributions approach would have a selection on the Help menu or a button somewhere on the screen that would open a web browser or similar interface and take the user to a listing for the programmers who created the screening module. For each programmer, the listing could include as much information as the programmer would be willing to provide and give further links to any social networking pages, resumes, or other professional and appropriate information the FOSS programmer desired to post. The active attributions method would operate through a centralized clearinghouse where FOSS programmers could register and provide secondary links. As a user, if you wanted to, you might be able to quickly go to a picture and description of the FOSS programmer whose code you were using.

If active attributions were indicated by a button on the screen, it could always be present and change from programmer name to programmer name as the user moved through the software. If multiple names are associated with a particular area, the methodology could pick one at random. This approach would need to be standardized, and should probably require the use of real human names (or perhaps at least first names) as opposed to online aliases. The point of this subsection is not to enumerate every detail of active attributions, but rather to point out that, first, the software technology to

---

[199] Vetter, *Collaborative Integrity*, *supra* note 2, at 583-86.

implement it is readily available. Second, this Article argues that, by raising the intangible value of attribution, active attributions may increase developers' motivation to contribute to FOSS.[200] Finally, this approach leverages the trend toward social networking and, specifically, open source customizations of social networking sites.

3. *Subsidies*

The final facilitator is the one that may give FOSS the greatest chance to succeed in the EMR software market: government subsidies. Of course, subsidies could also come from companies. Although some FOSS contributions from companies might *seem* to be purely a donation of source code, most are likely to be strategic business maneuvers with some rationale of complementary benefit. Thus, in mentioning this possibility, the focus is government subsidies.

Government support for FOSS - including the pros and cons of subsidies - as a general topic is beyond the scope of this Article. The aim of this Article is simply to note subsidies among the possible facilitators of FOSS development developed in this Part. Government action for FOSS ranges from high levels of support in some jurisdictions to the "level playing field" approach prevalent in the United States.[201] Subsidies to fund development of new FOSS programs raise a variety of issues. Subsidies where already-developed technology is made available are a different affair.

VistA, the government-subsidized program that facilitates the FOSS toehold in the EMR software market, was both a fortuitous and rare occurrence. The early history of VistA development within the VA established a practice of source code availability. Much of the VistA code base is directly applicable to the commercial sector, although it does not provide every component needed for a fully functioning system. In contrast, much of the software developed by government, particularly in the realm of national defense, is not directly applicable in other contexts.

Whether or not the policy balance justifies a governmental subsidy to create new FOSS in a particular market, it seems clear that the balance alters when the government software is already complete and in use. Thus, the fortuitous subsidization that fostered FOSS in the EMR software market seems to suggest that the most prudent approach to the subsidy issue is to consider what components can be harvested from the work that the government has already done, rather than considering what components the government could build from scratch. This translates into a policy preference for making government source code available, a recommendation that stands on independent grounds if the government itself is trying to become a FOSS user and participate in community development.

---

[200] For an active attributions implementation to be successful, Karim Lakhani points out that programmers would have to feel confident that the publicity wouldn't turn negative in the sense of users over-helping themselves in communicating with the programmer (assuming she provided contact information) for such things as support requests, or to complain if disgruntled with the software.

[201] Lee, *supra* note 13, at *55-*68.

VIII. CONCLUSION

If FOSS-enabling facilitators for software markets that disfavor FOSS are to have effect, they must work against the market's structural characteristics contributing to that inclination. Licensing approaches may be the most important facilitators. But non-licensing facilitators may be necessary in some markets, particularly those such as the electronic medical record (EMR) market in health care information technology, where other positive law or other factors might chill use of FOSS. Those factors include some of the characteristics observable in business process automation in health care, such as low end-user technical aptitude, differentiated workflow modeling needs, and environments where, for software vendors, there are primarily only non-platform complementary goods and services. This Article develops those three factors, along with three others: dispassionate computing agendas, extensive preexisting proprietary competition, and cost-accreting regulatory pressures. While perhaps not an exclusive list, the thesis that these factors signal a FOSS-disfavoring market arises from a detailed examination of the information technology needs for the EMR software market. If that case study generalizes to other enterprise software, or to other broader software markets, then facilitating efforts should try to counter the factors' influences while allowing them to mark a domain in which to operate.