



Contents lists available at ScienceDirect

# Robotics and Computer-Integrated Manufacturing

journal homepage: [www.elsevier.com/locate/rcim](http://www.elsevier.com/locate/rcim)

## Open source ERP business model framework

David L. Olson <sup>a,\*</sup>, Bjorn Johansson <sup>b</sup>, Rogerio Atem De Carvalho <sup>c</sup><sup>a</sup> University of Nebraska, CBA 209, Lincoln, NE 68588-0491, United States<sup>b</sup> Lund University, Department of Informatics, School of Economics and Management, Sweden<sup>c</sup> Federal Center for Technological Education of Campos, R. Dr. Siqueira 273, CEP 28030-130 Campos/RJ, Brazil

### ARTICLE INFO

#### Article history:

Received 21 December 2012

Received in revised form

24 March 2015

Accepted 7 September 2015

#### Keywords:

Business models

Enterprise resource planning (ERP)

Evaluation

Open source

### ABSTRACT

ERP systems became popular with large organizations in the 1990s. In the 21st Century, these products were expanded by addition of supply chain management (SCM) and customer relationship management (CRM), as well as access through the Web, creating the ERP II concept. Efforts to increase the market led vendors to serve not only large organizations, but also focus more on small-to-medium sized enterprises (SMEs).

Open source software has become a player in the field of enterprise resource planning (ERP) systems. While it is still unclear to what extent it has diffused among organizations, it is clear that opportunities exist. New ways of delivering ERP software, such as software as a service (SaaS) have appeared. Some smaller vendors utilized a free distribution system (Free/Open Source ERP, FOS-ERP) for their source code, relying on various business models for corporate success. There also have been attempts to generate FOS-ERP components found on sites such as SourceForge.com that are not only distributed freely, but also were developed through community participation much as Linux has been developed. Some ERP vendors use community developed components for various purposes to support their proprietary software. Thus one dimension of ERP systems is based upon who directs the development process. Proprietary ERP refers to systems with closely held intellectual property rights, such as the leading market products by SAP and Oracle as well as many smaller proprietary competitors. FOS-ERP can be community based, or sponsored by some organization.

In this paper we present a framework that aims at analyzing FOS-ERP business models. Goals include discussing the differences between FOS-ERP and their proprietary equivalents (P-ERP) in terms of business models, selection, customization, and evolution. We will discuss challenges and opportunities that they offer to adopters and vendors.

© 2015 Elsevier Ltd. All rights reserved.

### 1. Introduction

Open source software (OSS) has existed for several years. However, it can be stated that OSS impact is just beginning in many areas. One such area is enterprise resource planning (ERPs) systems. It can be stated that open source ERP systems have experienced increased interest, but it is still unclear to what extent its use has diffused among organizations [1]. What is clear is that ERP systems became popular with large organizations in the 1990s. In the 21st Century, these products were expanded by addition of supply chain management (SCM) and customer relationship management (CRM), as well as access through the Web, creating the ERP II concept. Efforts to increase the market led vendors to serve not only large organizations, but also focus more

on small-to-medium sized enterprises (SMEs). At the same time, new ways of delivering ERP software, such as software as a service (SaaS) appeared. Some smaller vendors utilized a free distribution system (Free/Open Source ERP, FOS-ERP) for their source code, relying on various business models for corporate success. There also have been attempts to generate FOS-ERP components found on sites such as SourceForge.com that are not only distributed freely, but also were developed through community participation much as Linux has been developed. Some ERP vendors use community developed components for various purposes to support their proprietary software. Thus one dimension of ERP systems is based upon who directs the development process. Proprietary ERP refers to systems with closely held intellectual property rights, such as the leading market products by SAP and Oracle as well as many smaller proprietary competitors. FOS-ERP can be community based, or sponsored by some organization. In this last case, West and O'Mahony [2] used the term Sponsored Open Source.

In this paper our purpose is to present a framework for analyzing FOS-ERP business models. Goals include discussing the

\* Corresponding author.

E-mail addresses: [Dolson3@unl.edu](mailto:Dolson3@unl.edu) (D.L. Olson), [bjorn.johansson@ics.lu.se](mailto:bjorn.johansson@ics.lu.se) (B. Johansson), [ratem@iff.edu.br](mailto:ratem@iff.edu.br) (R.A. De Carvalho).<http://dx.doi.org/10.1016/j.rcim.2015.09.007>

0736-5845/© 2015 Elsevier Ltd. All rights reserved.

## Nomenclature

BI	business intelligence
CRM	customer relationship management software
ERP	enterprise resource planning software
FOS-ERP	free open source ERP software
Metropolis model	crowdsourced software development model
OSS	open source software
OSSg2	second-generation open source software, hybrid between corporate and sponsored development

	approaches
P-ERP	proprietary ERP software
P/F model	model viewing open source development in terms of process and filtering
SaaS	software as a service – software available over the Web
SCM	supply chain management software
SME	small-to-medium sized enterprises
SOA	service oriented architecture, computer architecture capable of accessing software-as-a-service

differences between FOS-ERP and their proprietary equivalents (P-ERP) in terms of business models, selection, customization, and evolution. This can be of use to academics as well as practitioners. We will discuss challenges and opportunities that they offer to adopters and vendors. The focus of the paper is on providing a business model framework for analyzing open source ERP systems.

## 2. Defining ERPs, proprietary versus open source

ERP systems have provided a great deal of benefit to business operations by integrating legacy systems, providing greater accuracy through combining databases, efficiencies through business process reengineering, and providing platforms to which added functionality can be linked [3]. The goal of ERP is to integrate and consolidate all the systems across an organization into a one system that can meet and serve each department's unique needs and tasks. While we recognize that rarely are all systems integrated, ERP installation moves toward more centralization of organizational computer support. Therefore, every aspect of an organization's business process need to have a unified application interface, which provides high competitiveness in the market – or at least makes the organization compete in equal terms to their competitors that also uses such a system. Enterprises have invested heavily in ERP acquisition while small businesses or entrepreneurs often could not see how they could afford such systems mainly due to its high upfront prices and resources required to deploy and maintain the system. To attack this niche market of ERP in the SME sector, vendors have transformed ERPs by adopting the information technologies that can reduce adoption costs, such as Web-based modules, as well as created simplified versions of their systems. FOS-ERP in general has become a viable alternative [4]. Cereola et al. [5] reported that while large enterprises primarily utilize proprietary ERP systems that are unlikely to migrate to open-source solutions, SMEs are suitable candidates for open-source ERP due to their agility and flexibility. Commercial open source ERP allows modifications to source code enabling firms to exploit their unique business processes and retain competitive advantages. Johansson and Sudzina [1] noted that open-source ERP system interest has exploded, appearing to target SMEs. Poba-Nzaou and Raymond [6] gave two cases of open-source ERP lowering risk for SMEs.

Strategies of ERP development include software as a service (SaaS), open-source software (OSS) and Service Oriented Architectures (SOA), as well as hybrids of these. SaaS offers ERP as a service that clients can access via the Internet. Smaller companies have spared the expenses associated with software installation, maintenance and upgrades [7].

### 2.1. The open source software paradigm

As a reaction on proprietary software development open source

software (OSS) development started as voluntary participation of individuals linked by the Internet to develop code. OSS has become a viable means of software creation. The most commonly understood open source success is the Linux operating system, used by Dell, Compaq, and IBM as well as many other firms. MySQL is an open source database management system. Sun Microsystems have long viewed OSS as a means to develop long-range market strength [8], although this effort was insufficient to avoid short-term failure. Other firms, however, have been able to make OSS work, to include Dell computers [9]. RedHat [10] claims that OSS can save businesses money by:

1. Enabling use of commodity hardware rather than proprietary machines;
2. Avoidance of expensive maintenance contracts;
3. Obtaining greater functionality, reliability, and performance;
4. Increasing productivity through a faster learning curve and availability of support tools;
5. Avoidance of vendor lock-in;
6. Reduction of the need for specialized security consultants and tools.

Real competition between proprietorial vendor products and OSS include Microsoft's IIS software vs. OSS Apache in the Web server software market, MySQL vs. traditional database vendors such as Oracle and Microsoft, and Intuit's Quicken and Microsoft's Money vs. the OSS GNU Cash [11].

Open source software may thus become a viable alternative to proprietary software, with an obvious cost advantage. There are risks, in that one cannot expect the same level of service with OSS as with proprietary alternatives. However, support for many OSS products is available, from such organizations as IBM and Red Hat. Contemporary software selection thus requires considering the tradeoffs between OSS and proprietary software.

Weber [12] reviewed industry surveys seeking to identify why participants gave their time to open source endeavors. The underlying philosophy of OSS is to enhance software reliability and quality through independent peer review and rapid evolution of source code. With OSS, developers and users are free to utilize and modify OSS by accessing open code [11]. OSS projects have become popular since developers and adopters are able to implement OSS easily and give feedback promptly.

### 2.2. Open source software vs. proprietary software

Watson et al. [13] described four models on the spectrum of ways in which software can be developed. In Proprietary and Open Communities, software programmers tended to freely exchange codes. Proprietary firms came to take over, carefully protecting their programs by selling executable versions, having a closed approach to software development. The proprietary model dominated the market, leading to firms hiring their own software

developers with strong protection mechanisms. Examples include Microsoft operating system and Google, developed in-house and operated with closely controlled code. Some proprietary code is sold for fees, but some is distributed free, such as IBM Linux code. The distinguishing characteristic is the limitation of customers to view and modify the code. Conversely, open communities develop and support software through voluntary participation. This can be for a fee, as with MySpace and YouTube, or distributed free, such as the Linux Kernel or Wikipedia.

A second variant is Corporate Distribution, where many open source products have been developed, to include Linux. Most business users would have difficulty interacting directly with open source developers. Thus an industry has developed including firms such as Red Hat that create value and collect fees by providing support to those firms that would like to adopt open source software products. These firms identify best-of-breed open source products, improve distribution, and offer complementary services such as maintenance support.

Yet a third variant is where firms sponsor open source development. Under Sponsored Open Source, Apache Software Foundation encourages development of the Apache server and other open source projects. IBM contributes many developers to the Apache Web server. IBM released their formerly proprietary Eclipse integrated software development environment as open source, while encouraging their employees to continue development work on it. Sun (prior to being acquired by Oracle) was a heavy sponsor of open source products.

A more recent variant, Second-Generation Open Source (OSSg2) is a hybrid between corporate distribution and sponsored open source. Companies generate revenue by providing complementary services, while providing most of the development. Unlike the corporate distribution model, OSSg2 firms usually do not sell licenses, and unlike the sponsored open source model, they typically retain control over the software code. Examples include JBoss and MySQL.

The open source software industry has undergone a transformation. Raymond [14] contrasted the traditional method of development with open source approaches. The traditional approach was referred to by the term cathedral, as cathedrals required strict and closely controlled organization with many rules and requirements. The open source approach was referred to by the term bazaar, implying a free market of ideas without central planning. More and more, as the Internet enables widespread participation of the masses, people have been cooperating in developing new products, especially in the software arena. The open source movement has made significant inroads into the field. Benkler [15] cited open source software as superior in attaining high quality and dependability in any economy, and providing additional benefits to developing economies in the form of lower cost, higher transparency, release from vendor lock-in, and improved development of local programming capabilities. Bonaccorsi et al. [16] surveyed 146 Italian software firms concerning their business models with respect to adopting open source development. There is a trade-off in that open source software is free, with a proven record of reliability and security, and the support of a large community of talented participants. Negative factors include a relative lack of compliance with industrial standards, documentation, maintenance, product updating, and liability responsibility. Adopting an open source approach reduces the ratio of fixed cost to total cost, which lowers the breakeven point and risk. However, potential customer perceptions of reliability need to be overcome.

SourceForge.net is the largest OSS on-line OSS community and it is the global technology community's hub for information exchange about OSS distribution and services, listing approximately 348,000 open-source projects including enterprise software, such as ERP, business intelligence (BI), CRM, and so on (as of February

2009). There were over 1000 on-going ERP projects on SourceForge.net as of May 7, 2009, overlapping with CRM and SCM projects. In 2015, there were 56 ERP products listed, with download links. Most have free download options with stronger variants available for a fee. Usually service, training, and maintenance packages are also available to provide OSS vendors with revenue streams. The zero-cost licensing structure of most open source projects has opened up the acceptance of these products into a number of previously untapped markets. Among the projects in SourceForge.net, OpenBravo is an example of successful enterprise OSS, which has been developed by 82 participants over three years with more than 970,000 downloads recorded.

However, research on FOS-ERP software is limited, and there are many aspects differentiating FOS-ERP from proprietary ERP that are not well understood. Smets-Solanes and de Carvalho [17] published an early paper on FOS-ERP, describing development of ERP5. De Carvalho [18] demonstrated that evaluation of FOS-ERP involves high levels of uncertainty relative to evaluation of proprietary software. OSS has been cited as a source of innovation [12]. Dreiling et al. [4] cited proprietary motivation to use software architecture as a means to direct the market to their products, actually stifling innovation through vendor lock-in.

Another issue is strategic differentiation. Davenport [19] argued that ERP might not provide strategic benefit to clients if all clients adopted the same system. De Carvalho [18] pointed out that proprietary software restricts real differentiation and raises customization costs more than FOS-ERP.

We see a spectrum of ERP products, with high-end proprietary products like SAP and Oracle dominating the large organization market, and completely internally developed systems at the other extreme. There are many proprietary competitors to SAP and Oracle, with Microsoft notably seeking a niche with medium-sized organizations through pricing. There are a variety of forms of free open source ERP products around too, as indicated by Johansson and Sudzina [20]. Dreiling et al. [4] differentiated products in this realm on the dimensions of open or closed (proprietary) development along with variants of what "free" implies in distribution of the product, suggesting three variants:

1. Proprietary development, free distribution (with various business models for more functionality);
2. Open development, commercial distribution (possibly OpenMFG);
3. Open development, open distribution (possibly GNU ERP).

### 3. Open source ERP

Recently, ERP vendors have realized that open source systems (OSS) have capabilities, both as a source of content for vendors as well as a threat to the proprietary enterprise system market share from competitors based on OSS development or delivery [21]. Open source ERP systems are used by firms such as Home Depot, Toyota, and Fidelity [12]. There is a market of open source ERP products, to include Compiere, and OpenMFG. Recently, as open-source projects became more popular, there have been some studies on OSS ERPs. As ERPs are commonly implemented by organizations, it is hard to attain competitive advantage through implementation of ERP [22]. Jaisingh et al. [11] analytically compared software developed through open source and closed source. They cited open source software as emerging as a significant alternative to closed source software in the server market, in office productivity tools, accounting software, and database systems. They argued that the primary difference between these two sources of software was that in OSS, source code is accessible to all, and that OSS can benefit from modifications and improvements

from all participating programmers. They concluded that OSS was eliminating motivation for innovation by proprietary firms in markets where there was only one closed source competitor (such as Web servers), but not when there were multiple closed source competitors (as in database systems, where proprietary products attract organizations with large scale database uses, and OSS dominates for small and large firms with simpler database applications).

### 3.1. OSS ERP products

Dörner et al. [23] cite open source projects such as Adampiere ERP Business Suite to have yielded high quality results, due to the ability of all participants to see the code, to reuse code, and to branch to new projects through forking. Even major ERP vendors such as SAP and Oracle have switched to service-oriented architectures, making it easier to program add-ons. Johansson and Sudzina [6] classified open-source software into community (developed by the open-source community and so-licensed) and commercial (software owned by a for-profit organization but with source code available to user, with for-profit organization retaining copyright) categories.

There are open source developed and freely distributed ERP products like GNU Enterprise. There are many more company sponsored FOS-ERP to include Compiere, ERP5, OpenMFG, and SQL Ledger, that have achieved some success, owned and developed by a for-profit entity. Some of these company sponsored products were at least partially developed through open-source participation (community), but others were developed internally. Thus the spectrum of products is complicated a bit. SAP was originally developed internally, and distributed with closed source code (an example of completely proprietary). There are many products that were developed internally but are distributed for free (Compiere is an example). Other products may involve utilization of open development for improvements (like IBM, SAP encourages employees to participate in open projects with the intent of developing improved modifications of portions of their systems). At the other extreme, GNU Enterprise is openly developed under a license that is open. However, this extreme seems dubious as a viable business model.

To demonstrate the relative success of free ERP products distributed by commercial firms, we note relative downloads from SourceForge.com. Johansson and Sudzina [1] presented data on downloads of six open source ERP products. However, it is important to note that, given that downloads are for free, the number of downloads gives an indication of success, but, it is very hard to check how many of these downloads became deployments. In fact, for P-ERP, to check the number of users is a question of counting the number of contracts in the partner network. For FOS-ERP, the network has formal (partners of the developer company, in the case of Sponsored FOS-ERP) and informal partners, as well as, companies that download and use the system by themselves.

Other OSS ERPs listed by Stoilov and Stoilova [24] include CK-ERP, ERP5, Fisterra, Hipergate, iPyME, Jerpa, OfBiz, and SugarCRM. These authors compared the functionality of Compiere, TinyERP (now part of ODOO), and OfBiz, highlighting the variety of functionality provided by various systems. Leina [25] reported a case of Compiere customization by a SME in China, finding that OSS makes IT affordable for Chinese SMEs by reducing cost across the board, as well as providing transparency, avoiding vendor lock-in, and providing flexibility in options provided. Baharum et al. [26] interviewed two Malaysian SMEs about their experience with OSS ERP. The business driver as would be expected was financial benefit, but the ability to customize was also present. Barriers to overcome included database migration, synchronization of software to company workflow, developing user interfaces and user

support, and integrating third-party software. Leina et al. also conducted a survey of other Chinese SMEs to identify barriers to OSS ERP implementations.

OSS ERPs allow SMEs' access to ERP. The benefits of applying OSS are as follows [27]: (1) Increased adaptability: Since ERP is not plug and play, implementation processes are necessary to match the company's business processes and local regulations. Having full access to the ERP source code is beneficial. (2) Decreased reliance on a single supplier: Proprietary ERPs impose heavy reliance on maintenance from vendors and distributors. (3) Reduced costs: Proprietary ERP licenses are expensive. OSS ERPs' average implementation costs are at between one-sixth and one-third of the costs for typical proprietary ERPs. However, total cost of ownership is difficult to estimate. There are costs in both proprietary and open ERPs for internal IT staff, for training users and staff, and impact on organizational operations through reengineering business processes. Almost every article published recognizes the need for consideration of total cost of ownership, but there is no convincing empirical evidence of relative advantage. Each organization would face their own context, and if they do actually measure total cost of ownership, they are unlikely to be motivated to share accurate numbers.

### 3.2. Open source business models

Raymond [14] gave nine basic open source business models for software development. All methods involve development cost reduction through use of open source communities, and generation of revenues through indirect sales:

1. Cost sharing: This can be done by the company joining or establishing an open source community at the sacrifice of sales. A variant is a community source model where end user organizations jointly develop applications intended for use by all participants.
2. Risk spreading: Risk spreading refers to offering software as open source, enabling the developing organization to share maintenance work with the community of participating software engineers. Sales are again sacrificed, but this form is especially attractive for software to be used internally.
3. Loss leader (or market positioned): Developing companies can offer free software as a loss leader, with the aim of inducing sales for related products (software or service).
4. Widget frosting: This approach targets hardware manufacturers, seeking to have them add your software to their product.
5. Give away the recipe, open a restaurant: Distribute the software without charge, offering associated services for a fee. Examples include JBoss's distribution of their open source products without charge, relying upon support contracts for revenue, and the subscription model used by RedHat and Linux who offer annual service agreements bundled with open source software, customer support, or software updates.
6. Accessorize: Distribute the software without charge, offering associated products for a fee. An example is SugarCRM, which distributes basic open source products without charge, but offers proprietary versions or extensions for a fee. Another form of hybridization is exhibited by MySQL, offering dual licenses.
7. Free the software, sell the present: Sell the software with a date of expiration, after which the software will become open source. This offers proprietary software, giving users the ability to tailor it to their needs once the expiration date ensues.
8. Free the software, sell the brand: Distribute the software without charge, but with associated compatibility standards. Revenues are generated through compatibility validation tests and statements.

9. Free the software, sell the content: Distribute the software without charge, but with fees for content. This can include vendors selling and supporting hardware devices or appliances including open source software as complementary products. Mazu Networks offer a network security product under this model. The open source software component is a complementary product.

Kazman and Chen [28] proposed a Metropolis Model for crowdsourced systems, relating to both software development and to general business. They stated the world was also contended to be moving to service orientation as opposed to traditional product orientation. This is an undeniable trend, where tangible products need to consider service aspects in order to make any sales. Kazman and Chen [28] see the implications of this new approach to creating products emphasizing crowd management, and separation of kernel content and peripheral components. The requirements process needs to be much less formal, as it evolves rather than following a carefully laid out plan. There is a need to focus on architecture to provide a structure under which commons-based activity can fruitfully evolve. Testing is much better, given the many participants that check on content.

Software development firms have the opportunity to utilize open source bazaars. This is attractive because of higher reliability and quality, as well as cost advantages. How this develops into a sustainable profit-making operation for firms remains to be seen. Furthermore, many large firms such as Sun and IBM have encouraged their employees to participate, creating a new motivation for participation. This muddles the differentiation between community and commercial development somewhat.

Open source technology makes possible cooperative development of information technology tools, making it possible to use small bits of functionality developed by others and tested by the market rather than having to develop everything yourself. Open source products have been developed for financial applications [29], marketing applications [30], and many other fields.

Economides and Katsamakos [31] found that the OSS industry is more lucrative than the proprietary software industry when adopters have weak needs for platform while they want strong applications or customization, reinforcing the tendency to grow the service side of software. In this regard, OSS and proprietary software complete in many areas of the software market [11]. According to Lee and Mendelson [32], the optimal strategies of these two industries depend on both product timing and compatibility. Network effects (the ability to link work to other organizations, as in supply chains) make compatibility a key competitive factor. Network effects also create an intense competitive environment, driving profits down if the technologies are incompatible. If the OSS can establish itself first, the commercial firm is always better off mitigating the network effects by following the OSS lead and designing a compatible product. Sen [33] pointed out that most OSS are late entrants to the market, already consisting of proprietary and its complementary/supplementary products. In order to attract users in this market, the OSS will have to be compatible with these existing complementary/supplementary products. Even in a situation where all consumers eventually adopt the proprietary software, the presence of OSS as a credible threat drives down prices, thereby increasing consumer surplus. Open source developers derive not only personal satisfaction from developing software but they also seek to maximize a measure of consumer surplus and therefore suffer if incompatibility splits the market.

Implementation of proprietary systems such as SAP and Oracle require large amounts of investment and its inherent risk often makes firms hesitant to invest large amounts in such systems, especially in the case of SMEs. Paulson et al. [34] studied tradeoffs

between OSS and proprietary software products. The study rejected hypothesized OSS advantages in speed of system development, and expected OSS advantages by their being simpler and more modular. OSS and proprietary projects studied were found to have similar development times. Proprietary projects were generally found to be simpler in terms of complexity. And OSS projects had a significant correlation between growing rate and changing rate, while closed proprietary projects did not. That study did find support for two hypotheses. OSS projects led to more creativity, as measured by functions added over time. And OSS projects were found to have fewer defects, expected because of OSS ability to tap a large number of reviewers.

#### 4. The evaluation framework

We build our business model hierarchy shown in Table 1 following Osterwalder and Pigneur [35]. There are two extremes for software development – in-house development and open source. There are also two extremes for distribution – copyrighted/patented fees, and free distribution. In-house development and proprietary distribution is the common industry approach in developed economies, exemplified by Microsoft and Google. In-house development combined with free distribution is practiced by IBM with respect to Linux code. Open source development is used by many, to include companies that use open source code with proprietary distribution, such as MySpace and YouTube, as well as with free distribution, as has been done with Linux operating system and Wikipedia. In-house development with proprietary distribution has led to enormous wealth generation in the cases of Microsoft, Google, SAP, Oracle, and many other software firms. But open source development is gaining reputation as better code with respect to quality and reliability. It also is highly attractive in developing economies, especially since the options of distribution are evolving.

In the Web 2.0 environment, information consumers are becoming information providers. Pro-sumers (producer and consumer or professional and consumer), highly skilled hobbyists before Web 2.0, are being replaced by Pro-Ams (those who are somewhere between professional and amateur) who are creating contents of value that contribute to society and the economy. The human brain has limited capacity to process the abundance of information being generated. The massive amount of information available from Web 2.0 diverts attention. Attention itself is the most important value in the Web 2.0 world. How to create new business models based on this new value is indeed a critical question.

Value creation includes two processes: production (P process) and filtering (F process). The majority of Web 2.0 business models can be explained in P/F model [36]. The P process produces value and the F process filters by editors selecting and retaining specific

**Table 1**  
Clusters of the P-F Model in OSS ERP.

	Process	Filtering	Implication	OSS ERP products
Model 1	Limited	Limited	Free software, no user participation	Compiere Nexedi Openbravo Fisterra others
Model 2	Open	Limited	Hybrid collaboration	OpenMFG
Model 3	Limited	Open	Competition	Internal knowledge management
Model 4	Open	Open	User participation in production, filtering	External knowledge management

software features. There were only a limited number of people who, called professionals, could participate in P or F steps in traditional industrial economic systems. P was limited to reporters in the press or developers of software firms and F was limited to editorial writers, managers and marketers. However, in the current Web 2.0 era, production can be done by the crowd. Scarcity of information is achieved from filtering or screening by professionals. Professionals can make the information retained more valuable by filtering and screening. Therefore, most of the current Web 2.0 businesses belong to one of the following four models. When extending this model to open-source ERPs, we limit the meaning of “open-” only to “free-of-charge” [37].

**Model 1:** Limited production process, limited filtering process (free software, no participation by users): All activities are done by the members/employees. This guarantees high quality of final products and services after the F process.

**Model 2:** Open production process, limited filtering process (hybrid collaboration model such as Linux): P is done by users and F is done by a small number of professionals. High quality products and services are guaranteed.

**Model 3:** Limited production process, open filtering process (competition model): P is done by a small number of professionals or Pro-Ams and F is done by public users. F is pre-screened by professionals then open to public for voting, inducing public participation. Public opinions are emphasized. While no existing ERP systems appear to exist in this model, it can represent internal knowledge management activities.

**Model 4:** Open production process, open filtering process: Everyone can be a producer. F can also be open to everyone. F process is more important because of the scarcity of attention. Quality of final products or services is not guaranteed. While no existing ERP systems appear to exist in this model, it can represent external knowledge management activities, drawing upon Web 2.0 links for gaining additional understanding of organizational problems. Table 1 gives examples of OSS ERP software for each of these models.

Of these models, only Model 1 has proven to be commercially viable. The most common business model of OSS is based on a simple idea – free for use, modification, resale and fee for services including implementation. Table 1 shows that most ERP-related open-source software uses the Web for delivery of free software (Model 1). There is at least one product (OpenMFG) allowing users to participate in software development, but with software vendor filtering (Model 2).

While OSS ERP components are widely reported on SourceForge.com, rarely if ever are viable commercial products evolving in a form developed by the community model. Rather, viable products, distributed as FOS-ERP or not, have some company distributing them [18]. Major differences between FOS-ERP and proprietary products are the exposure of source code, meaning that the client is able to customize. Vendors can also participate in customization, under various licensing arrangements to impose control.

In the requirements and preliminary design phases, there is no clear difference between FOS-ERP and proprietary development for a specific client, other than that proprietary products offer what they view as best practices for generic clients. When design gets to the stage of business process reengineering, greater differences may arise. Evaluating proprietary ERP products involves comparing alternatives in terms of functionality, total cost of ownership, and technological criteria. For FOS-ERP, the functionality is likely to be reduced (as a price for free software), total cost of ownership is clouded by uncertainty with respect to implementation, training, and future benefits, and technological criteria involve uncertainty due to lack of experience. FOS-ERP offer clear cost savings in terms of fees paid to proprietary

organizations and ensuing maintenance expenses due to vendor lock-in. But there is value that proprietary organizations provide for those fees.

Among the open-source ERP vendors given in Model 1 in Table 1, Compiere has most often been appeared in research articles and business reports. Compiere recorded more than 1.2 million downloads of its software and has more than 100 partners in 25 countries. They do not sell software but sell services – security and support. They do not allow just anyone to contribute code – the majority of code contributors are trained partners who understand company's business model. Therefore this belongs to limited P/limited F model in the P-F model.

The ERP software OpenMFG allows community members including customers and partners to get the source code and extend and enhance it. The company utilizes the enhancements in their product. This could be an example of typical open P/limited F model.

So far we have not seen products fitting Models 3 and 4, but Web 2.0 development can lead to systems of value fitting these models. Both are open, offering a collaborative environment inductive to innovation. The difference between Models 3 and 4 is the degree of participation. Internal focus groups working on improved business process development would fit Model 3, while the inclusion of external ideas would fit Model 4.

## 5. Conclusion

The research suggests that FOS-ERP offer many opportunities for their clients. For smaller adopters, there are lower fees paid to proprietary firms. This enables ERP to be a viable option for SMEs, who simply do not have the budget for most proprietary products. There is a need to differentiate between small and medium enterprises. Medium-sized organizations can afford smaller proprietary ERP products, such as those produced by Great Plains (now Microsoft Dynamics GP), Lawson (now part of INFOR), Sage, and other vendors. Small enterprises may not be able to afford this level of expenditure. FOS-ERP offers a viable alternative for these small organizations. This is also true for local governments and for firms operating in developing countries. On the downside, lower costs imply lower functionality, placing added burden on IT staffs of FOS-ERP adopting clients. For FOS-ERP vendors and consultants, there are opportunities to gain experience in this emerging market, enabling them to provide value to many clients as they gain experience. There are also opportunities for individuals working in IT. Experience in implementing FOS-ERP can be a valuable skill with many, many potential SMEs, governments, and non-profit organizations as potential employers.

There are massive differences in markets making each of these variants attractive in some settings. Proprietary mainstays such as SAP and Oracle target large organizations, and provide top-of-the-line functionality. Microsoft and other proprietary vendors sell their systems for a magnitude less, with less functionality. Vendors such as SAP utilize OSS as a source of specific functionalities added to their products. Other vendors, such as Compiere and Nuxeo, develop their systems internally, but distribute these systems using alternative business models (such as providing the software free, with billing for services such as installation or maintenance). This enables coverage not only for medium sized businesses, but potentially for small enterprises, governmental agencies, or non-governmental organizations (NGOs). In general, the more functionality or support required, the greater the fees. Thus we think that there is a great opportunity for small organizations to obtain enterprise computing at affordable prices as long as they have technically versed employees able to install and maintain such systems.

The primary finding from the evaluation of FOS-ERP business models is that OSS projects offer opportunities to selectively enhance enterprise computing with potential cost savings. FOS-ERP can be especially rich in enhancing support activities for firms utilizing ERP, CRM, and SCM. The tradeoff is that implementing OSS products into existing systems (or constructing systems with OSS components) is not without cost. There is a need to integrate the products with services. While the software cost is minimal if anything, the total cost of ownership can be high. For some products, such as Compiere, OSS ERP products have industry firms available to serve as consultants, much as Red Hat does with Linux. While there is cost for such support, it is still unclear what the total project cost would be comparing with mainline ERP systems such as SAP or Oracle.

Limitations of this study are that while it reviews open-source ERP products, there is much that we do not know. Interesting questions remain about relative usage, as well as trends. We feel we have detected a viable market for open-source ERP, but do not have solid evidence of actual use, nor is such knowledge available at this time.

## References

- [1] B. Johansson, F. Sudzina, ERP systems and open source: an initial review and some implications for SMEs, *J. Enterp. Inf. Manag.* 21 (6) (2008) 649–658.
- [2] J. West, S. O'Mahony, Contrasting community building in sponsored and community founded open source projects, in: Proceedings of the 38th Annual Hawaii International Conference on System Sciences, Hawaii, 2005.
- [3] C.A. Watts, V.A. Mabert, N. Hartman, Supply chain bolt-ons: investment and usage by manufacturers, *Int. J. Oper. Prod. Manag.* 28 (12) (2008) 1219–1243.
- [4] A. Dreiling, H. Klaus, M. Rosemann, B. Wyssusek, Open source enterprise systems: towards a viable alternative, in: Proceedings of the 38th Annual Hawaii International Conference on System Sciences, Hawaii, 2005.
- [5] S.J. Cereola, B. Wier, C. Strand Norman, Impact of top management team on firm performance in small and medium-sized enterprises adopting commercial open-source enterprise resource planning, *Behav. Inf. Technol.* 31 (9) (2012) 889–907.
- [6] P. Poba-Nzaou, L. Raymond, Managing ERP system risk in SMEs: a multiple case study, *J. Inf. Technol.* 26 (3) (2011) 170–192.
- [7] M. Santosus, Impex Embraces Open-Source ERP, *eWeek.com*, 2006. (<http://www.eweek.com/c/a/Enterprise-Applications/Impex-Embraces-OpenSource-ERP/1/>).
- [8] C. Babcock, Open Source Will Pay Dividends in the Cloud, *Informationweek.com*, 18 (30), 30 March 2009.
- [9] A. Conry-Murray, Can Enterprise Social Networking Pay Off? *Informationweek.com*, 18 (29), 23 March 2009, pp. 23–29.
- [10] RedHat, The perfect storm, 2009. ([www.redhat.com](http://www.redhat.com)).
- [11] J. Jaisingh, E.W.K. See-To, K.Y. Tam, The impact of open source software on the strategic choices of firms developing proprietary software, *J. Manag. Inf. Syst.* 25 (3) (2008/9) 241–275.
- [12] S. Weber, *The Success of Open Source*, Harvard University Press, Cambridge, MA, 2005.
- [13] R.T. Watson, M.-C. Boudreau, P.T. York, M.E. Greiner, D. Wynn Jr, The business of open source: Tracking the changing competitive conditions of the software industry, *Commun. ACM* 51 (4) (2008) 41–46.
- [14] E. Raymond, *The Cathedral and the Bazaar: Musings on Linux and Open Source by Accidental Revolutionary*, O'Reilly, Sebastopol CA, 1999.
- [15] Y. Benkler, *The Wealth of Networks: How Social Production Transforms Markets and Freedom*, Yale University Press, New Haven, CN, 2006.
- [16] A. Bonaccorsi, S. Giannangeli, C. Rossi, Entry Strategies Under Competing Standards: Hybrid Business Models in the Open Source Software Industry, *Manag. Sci.* 52 (7) (2006) 1085–1098.
- [17] J.-P. Smets-Solanes, R.A. de Carvalho, ERP5: A Next-Generation, Open-Source ERP Architecture, *IT Prof.* 5 (4) (2003) 38–44.
- [18] R.A. De Carvalho, Issues on evaluating free/open source ERP systems, in: A. M. Tjoa, L. Xu, S. Chaudhry (Eds.), *Research and Practical Issues of Enterprise Information Systems*, Springer-Verlag, Boston, 2006, pp. 667–676.
- [19] T. Davenport, Putting the enterprise into the enterprise system, *Harv. Bus. Rev.* 76 (4) (1998) 121–131.
- [20] B. Johansson, F. Sudzina, "Choosing Open Source ERPs: What are the Reasons," in: CeaE. Boldyreff (Ed.), *Open Source Ecosystems: Diverse Communities Interacting*, Springer Verlag, 2009, pp. 143–155.
- [21] R. Grewal, G.L. Lilien, G. Mallapragada, Location, location, location: How network embeddedness affects project success in open source systems, *Manag. Sci.*, 52, (2006) 1043–1056.
- [22] J. Karimi, T.M. Somers, A. Bhattacharjee, The impact of ERP implementation on business process outcomes: a factor-based study, *J. Manag. Inf. Syst.* 24 (1) (2007) 101–134.
- [23] C. Dörner, S. Draxler, V. Pipek, V. Wulf, End users at the bazaar: Designing next-generation enterprise-resource-planning systems, *IEEE Softw.* 26 (5) (2009) 45–51.
- [24] T. Stoilov, K. Stoilova, Functional analysis of enterprise resource planning systems, in: Proceedings of International Conference on Computer Systems and Technologies 2008, 2008.
- [25] Z. Leina, Development and Implementation of ERP/CRM System Based on Open Source Software to Small and Medium-Sized Enterprise in China, 2008.
- [26] Z. Baharum, M.S. Ngadiman, H. Haron, Critical factors to ensure the successful of OS-ERP implementation based on technical requirement point of view, *International Conference on Modelling & Simulation*, IEEE Computer Society (2009), p. 419–424.
- [27] N.S. Serrano, J.M. Sarriegi, Open source software erps: a new alternative for an old need, *IEEE Softw.* 23 (3) (2006) 94–97.
- [28] R. Kazman, H.-M. Chen, The metropolis model: a new logic for development of crowdsourced systems, *Commun. ACM* 52 (7) (2009) 76–84.
- [29] D. Kane, J.D. Masters, Open source finance, *J. Invest.* 18 (1) (2009) 92–96.
- [30] C.S. Fleisher, Using open source data in developing competitive and marketing intelligence, *Eur. J. Mark.* 42 (7/8) (2008) 852–866.
- [31] N. Economides, E. Katsamakos, Two-Sided Competition of Proprietary vs. Open Source Technology Platforms and the Implications for the Software Industry, *Manag. Sci.* 52 (7) (2006) 1057–1071.
- [32] D. Lee, D., H. Mendelson, Divide and conquer: Competing with free technology under network effects, *Prod. Oper. Manag.* 17 (1) (2008) 12–28.
- [33] R. Sen, A strategic analysis of competition between open source and proprietary software, *J. Manag. Inf. Syst.* 24 (1) (2007) 233–257.
- [34] J.W. Paulson, G. Succi, A. Eberlein, An empirical study of open-source and closed-source software products, *IEEE Trans. Softw. Eng.* 30 (4) (2004) 246–256.
- [35] A. Osterwalder, Y. Pigneur, C.L. Tucci, Clarifying business models: Origins, present, and future of the concept, *Commun. Assoc. Inf. Syst.* 16 (1) (2005) 1–25.
- [36] H.M. Kim, K. Lyons, M.A. Cunningham, Towards a Framework for Evaluating Immersive Business Models: Evaluating Service Innovations in Second Life, 2008.
- [37] S.M. Lee, S.M., D.L. Olson, S.-H. Lee, Open process and open-source enterprise systems, *Enterp. Inf. Syst.* 3 (2) (2009) 201–209.