Open Source ERP for SMEs

Hyoseob Kim [1], Cornelia Boldyreff [2]
1 Dongbu Information Technology Co., Ltd, 154-17 Samseong1-Dong, Kangnam-Ku, Seoul, 135-879, Korea, demian1987@dongbuit.co.kr
2 Dept. of Computing and Informatics, Faculty of Informatics, Brayford Pool, Lincoln, LN6 7TS, England, UK, cboldyreff@lincoln.ac.uk

ABSTRACT

For the last decade or so, the biggest category of the IT investment has unarguably been Enterprise Resource Planning (ERP). Most of the bigger corporations in the developed countries have implemented ERP systems with an aim to achieving competitive edge in their respective business areas. Now that the top end of the ERP market has been saturated, the main interest has moved to non-commercial sectors such as universities and small and medium-sized enterprises (SMEs). These organisations have not been able to benefit directly from the ERP revolution because an ERP implementation requires huge resources and entails high risks.

Over the same period, the concept of Open Source Software (OSS) has been enthusiastically adopted by the software engineering community. OSS has excelled in many systems software domains, for example, operating systems with Linux and web servers with Apache. Having observed these successes, the software industry has been showing interest in application domains such as enterprise information systems, more specifically ERP systems, as the next OSS candidates. In this paper, we outline the challenges as well as opportunities of OSS ERP development.

1 BACKGROUND ON ERP AND OSS

ERP once implemented is probably one of the largest scale examples of reuse since it represents a large proportion of the business operations at an organisation; and the current ERP implementations use commercial-off-the-shelf (COTS) packages that dominate the global ERP market. These are close-sourced, and used in the black box reuse style although a certain level of customisation or extension is usually carried out.

While there are many ERP vendors, a handful of major companies such as SAP and Oracle dominate the top end of the market. ERP has been eagerly adopted by most of the large

corporations in developed countries for the last decade because if implemented successfully it can serve its owners as a valuable long-lasting business asset, and it is also more maintainable than fragmented legacy systems. By the end of the 1990s, however, the ERP market had already started to saturate at the top end, making major ERP vendors consolidate themselves and, at the same time, seek for new business opportunities. As a timely example, recently Oracle has acquired PeopleSoft, which had in turn acquired J.D. Edwards not so long ago. Customers using these merged companies' products have every reason to worry about the continuous product support. This kind of problem, i.e., vendor lock-in associated with COTS products has been extensively discussed in the literature [1]. As the growth rate of new licenses slows down, major vendors are starting to enter the mid-size ERP market targeted at SMEs with stripped-down versions of their packages. This is because the original ones are heavy-weight and over-complex for SMEs to use effectively and efficiently. Whether these versions suit SMEs' needs well is still a topic requiring a careful attention.

OSS has flourished in many categories of software where well-defined system requirements exist, e.g., operating systems, web servers, database, and Microsoft Office-like applications. Now, Linux, Apache, PostgreSQL, OpenOffice are often preferred solutions to commercial ones not just because they offer a low total cost of ownership (TCO) but also they are considered high quality systems. This has been achieved through the open and collaborative nature of the OSS development. Bugs are fixed more quickly and user enhancements are made in time. The ease of communication amongst distributed developers through the Internet and support environments for web-based projects such as SourceForge have made this collaborative development possible. The ever-falling price of hardware has also contributed to the popularity of OSS. Unlike in the past, nowadays the cost of software applications could easily surpass that of the hardware components in a system. This reversal has forced suppliers and customers to rethink the conventional software licensing models used in business.

Bearing in mind these successes, enterprise information systems (EISs) like ERP and CRM (Customer Relationship Management) that business entities necessarily rely on for their daily operations are still dominated by large commercial vendors. These applications are crucial to the success of an organisation because they handle valuable business processes and logic through which their main revenues are generated. Recently, although it is still in its infancy, the OSS community has started to move into this sector. *Compiere* and *ERP5* are representative OSS ERP projects. They offer great opportunities as well as challenges.

2 OSS ERP: ERP FOR SMES

Because of the high complexity associated with large corporations' business processes, OSS ERP might never be suitable for them. Instead, SMEs are more suitable candidates for it. This observation has been verified by surveying OSS ERP packages available in open source repositories. Almost all of them are geared towards the SMEs market. More detailed results of this survey are given in Section 3. Unlike large companies, SMEs can more easily adapt themselves to ever-changing business environments. This business agility is one of the major strengths that SMEs have. With COTS ERP packages it is difficult to achieve because their evolution is tied to their vendors' support. SMEs need to implement ERP because they also want to use their resources more efficiently, and within supply chains they may have to engage in business transactions with large corporations that have already implemented their own ERP

systems.

There is another reason why OSS ERP development is currently viable. The maturity of OSS in general has made it possible that many components, i.e., building blocks, of an ERP system can be open-sourced, e.g., DBMS, OS, and web servers. Obviously, a company using some of the alternative COTS packages/components could find its ERP system's maintenance process seriously complicated because it will have to rely on the vendors of those COTS products for its future evolution. Equally companies relying on OSS component based developments will face a similar situation, but arguably there is a better basis for this co-dependent development amongst projects in the OSS community.

3 MARKET SURVEY OF OSS ERP PACKAGES

OSS projects offer a wealth of data to empirical software engineers. Unlike corporate data, this data is freely available for empirical studies. In order to determine the current status of the open source ERP market, we surveyed SourceForge.net, which is the largest open source repository. There are only a very small number of OSS ERP projects that are being maintained outside this repository. We have identified a total of 129 OSS ERP packages in this repository. Below we describe some of the main findings of this survey.

The ideal situation would be a small number of projects where a reasonable number of developers are taking part in, or at least the frequency curve showing a normal distribution. Instead the number of developers engaged in projects is highly positively skewed. This is due to the fact that most projects are one-person projects although one project surveyed has 52 developers. 75 projects out of 129 projects (58%), i.e., more than half are one-manned projects, and 115 projects (89%), i.e., almost 90% having less than or equal to five developers. According to this statistical data, it is clear that developers tend to prefer starting their own projects rather than joining existing ones.

By looking at the lifespan of each project, we have observed that the interest in this category of software has been accelerating for the past couple of years because of the popularity of OSS in general and the enormous economic benefits in particular.

However, analysis showed a very weak correlation between the number of developers and the lifespan of each project, indicating that developers will not join simply as time passes by. They have to be convinced of the usefulness of the software each project is developing. There must be other factors that cause some projects to have more developers joining up than the rest. In a similar phenomenon, there is also no strong correlation between the lifespan and the volume of CVS (Concurrent Versions System) activities of each project. In other words, software age does not correlate to its maturity in open source development. An open source software project can go years without producing a single line of code! The Pearson product-moment correlation coefficients (r_{xy}) of these two cases were 0.20 and 0.26, respectively.

The volume of CVS activities and the number of downloads are highly correlated with the number of developers. Their correlation coefficients are 0.77 and 0.76, respectively. Having many co-developers is essential for an open source project as it can lead to higher quality of software produced following the maxim known as dubbed Linus's Law by Eric Raymond: "Given enough eyeballs, all bugs are shallow."[2].

Although there is not a strong relationship between the number of developers and page views (r_{xy} = 0.38), there is a strong relationship between the number of developers and downloads (r_{xy} = 0.76). We can hypothesise that this is due to the fact that the more developers an open source project has the more likely it will release useful pieces of software that potential users want to exploit for their own uses.

Next we classified all the projects into two separate groups. The first group of projects have corporate sponsors whereas the second are independent from any commercial ties. We had initially anticipated that corporate-sponsored ones would have performed much better than the other group because they have greater means than the independent ones. However, after conducting t-tests, it turned that there are not significant statistical differences between the means of these two groups, i.e., in terms of number of developers, lifespan, CVS activities, and downloads. Since the variances of the corporate-sponsored projects are also much greater than those of the independent ones, these two cancel each other, resulting in no statistically significant differences in their means. A possible explanation on these seemingly strange statistical data would be that being fairly small in size the companies behind the corporate-sponsored projects have not been able to exert significant influence on them. Table 1, 2, 3, and 4 show the results obtained after conducting these t-tests.

We have also checked whether each project has released files to share with other developers. Because this is qualitative data, i.e., answerable in either *Yes* or *No*, we conducted a chi-square test instead of a t-test. The degree of freedom, *df* was 1, and the chi-square χ^2, 18.46. Because the critical value of chi-square, with one degree of freedom and alpha of 0.05, is 3.84, the test result is statistically significant. That is to say, corporate-sponsored projects are more likely to release files than the other group.

Table 1. t-test for the number of developers

Statistics	Corporate-sponsored	Independent
Mean	10.13	2.21
Variance	204.41	8.43
Observations	15.00	114.00
Hypothesised Mean Difference		0.00
Df		14.00
T		2.14
P(T<=t) Two-tail		0.05
t Critical Two-tail		2.14

Table 2. t-test for the lifespan

Statistics	Corporate-sponsored	Independent
Mean	639.20	612.66
Variance	242315.03	187638.28
Observations	15.00	114.00
Hypothesised Mean Difference		0.00
Df		17.00
T		0.20
P(T<=t) Two-tail		0.84
t Critical Two-tail		2.11

Table 3. t-test for the CVS activities

Statistics	Corporate-sponsored	Independent
Mean	2156.20	141.02
Variance	52202076.17	1162491.95
Observations	15.00	114.00
Hypothesised Mean Difference		0.00
Df		14.00
T		1.08
P(T<=t) Two-tail		0.30
t Critical Two-tail		2.14

Table 4. t-test for the number of downloads

Statistics	Corporate-sponsored	Independent
Mean	63865.80	3443.45
Variance	42391197192.17	768179425.81
Observations	15.00	114.00
Hypothesised Mean Difference		0.00
Df		14.00
T		1.14
P(T<=t) Two-tail		0.28
t Critical Two-tail		2.14

4 RELATED WORK

Because of the huge interest in ERP, papers reporting the research results on it have been published extensively; however, there has been very little reported research specifically on open source ERP. Only one paper has been published in the whole of ACM and IEEE Computer Society journals and proceedings until now [3] whereas more numerous articles have been published in non-academic industrial trade magazines. This implies that currently there is a gap between the market demands and the research carried out by academics. ERP crosses over two different disciplines: software engineering and business studies. In other words, dealing with EISs like ERP requires business expertise as well as technical expertise. This may have hindered the progress of the research in this area.

5 CONCLUSIONS AND FURTHER WORK

In this paper we have explained why it is an opportune time to consider OSS ERP packages as a good alternative to commercial ERP solutions. We have investigated the current status of OSS ERP packages by surveying ERP projects on SourceForge.net, a large open source software repository. We have observed both unexpected as well expected findings. Certainly, many developers and users are interested in this category of software for various reasons, e.g., potential big dotcom opportunities, improving their programming skills, exploitation of the projects at their corporate sites. Except for a very small number of projects, these projects have failed to produce quality software in which business users can trust based on the low volume of software produced and its low take-up determined by downloads. Learning lessons

from this survey's results and comparing ERP projects with more successful OSS projects, we offer the following observations.

First of all, it cannot be recommended to originate an OSS ERP project in the bazaar style, i.e., coding from ground up. This claim had been already made by Raymond [2]. An ERP system is a complex piece of software; therefore joining a promising and stable project is better than starting a new project from scratch in the bazaar style. This advice is aligned with the traditional wisdom of reuse as well. It can be predicted that some vendors will make their ERP packages open source because of the pressures from the major global vendors entering the SMEs market. Reflecting the history of open source development, many successful open source projects had started in this way.

Second, OSS ERP suppliers and developers need to consolidate their efforts towards producing good quality software that can compete with COTS based ERP products. An industry-wide consortium similar to OMG (Object Management Group) needs to be established to facilitate these efforts. Since the license fees of OSS ERP packages are free or at most very low, instead consortium members would have to generate their income by providing various services to their customers rather than selling software itself. In fact, this is not a great departure from the way the major vendor-dominated ERP market is run currently. Customers pay much more for services such as implementation and consulting than purely for license fees.

To extend the research described in this paper, we intend to investigate why the developers participating in OSS ERP projects behave in the ways identified in our survey. This may give clues to understanding why many OSS ERP projects have not been successful. In addition, we will use a questionnaire to determine more fully the needs of business users regarding the use of OSS ERP packages in their organisations.

To conclude, OSS ERP has not revolutionised the business environments of SMEs yet. However, it will be an interesting time ahead in this area because of the enormous potential economical benefits of OSS ERP and the fierce competition between commercial vendors and OSS suppliers that is likely to arise.

REFERENCES

1 **Carney, D.** (1997) Assembling large systems from COTS components: Opportunities, cautions, and complexities. Technical report, SEI Monographs on the Use of Commercial Software in Government Systems.

2 **Raymond, E. S**. (2001) The Cathedral & The Bazaar, pages 19–63. O'Reilly, revised edition.

3 **Smets-Solanes, J.** and **Carvalho, R. A.** (2003) ERP5: A next-generation, open-source ERP architecture. IEEE IT Pro, 5(4):38–44.