The Use of Open Source Software in Enterprise Distributed Computing Environments

A decision-making framework for OSS selection and planning

Jacob Krivoruchko
Nova Southeastern University, Fort Lauderdale, Florida, USA; University of Maryland University College, Adelphi, Maryland, USA; Advocate Health Care, Oak Brook, Illinois, USA. E-mail: krivoruc@nova.edu

Abstract. Firms increasingly rely on open source software for solving business problems and building mission-critical IT solutions. However, there are numerous issues associated with OSS, including its influence on the total cost of ownership (TCO) and supportability and upgradeability risks. While savings from obtaining a free copy of the software can be significant, software accounts for an average of 10% of TCO, while the majority of the costs are associated with project staffing. OSS requires significant investment into staffing because it needs to be carefully selected, customized, and installed. In addition, global communities may gather and dissolve at their will, so guarantees of support, revision, and bug fixes are minimal. Yet companies can gain competitive advantage through an ability to customize software to address specific business issues and exercising control over development, revision schedules, and modifications. OSS is not a panacea from the rising software costs. Instead, it is a serious initiative that has benefits, disadvantages, and risks associated with it.

Key words: Budget, business model, commercial software, distributed systems, enterprise computing, mission-critical project, open source software, reliability, risk, software selection, support, total cost of ownership.

1. Introduction

The open source market has evolved into a powerful force that is increasingly present in many areas of the industry, including web development, e-commerce, infrastructure management, financial applications, ERP, operations management, and more. Open source software (OSS) has established a strong presence among technology solutions that involve building client/server and complex distributed systems. Open source is not just a way to save money on the upfront software

acquisition cost. It enables individuals and companies to tailor the base release to their needs, follow their own upgrade schedules (if needed), and coordinate development activities without any vendor involvement. While many early releases of OSS were rather unstable, the latest releases can effectively compete against the best products sold by giants like Microsoft, Sun, and Oracle, both in terms of functionality and security.

Many organizations become increasingly dependent on software vendors' release schedules, prices, support, and business models. Long project planning, development, and implementation cycles, along with complexity of the systems and dependency of the entire firm's operations, have made switching vendors a costly and undesirable process. Businesses struggle to cope with rising costs and products that do not fully address their needs. Some companies see a real opportunity to obtain a long-wanted freedom from their vendors' plans and ambitions through OSS, while others are cautious and prefer to avoid uncertainty of the unsupported products. Forrester's survey of 120 large North American companies shows impressive statistics: 46% of them already use OSS, while 14% have short-term plans to incorporate it into their existing computing environments. The same research has also revealed that European companies are not far behind: 31% of the surveyed 35 large firms use OSS, while only 17% do not have any plans for utilizing the free software code [3].

Is this a good time for your organization to acquire OSS and enjoy its benefits? Do not take advantages gained through the use of OSS for granted. In reality, OSS is not free. For companies it means dependence on the global community for further upgrades, problem resolution, and support; uncertainty about software stability and reliability; the need to hire and pay additional talent to modify and maintain software, along with other caveats [2]. The dilemma is whether OSS represents a real developer's dream and a panacea against rising software costs or a risky venture for companies mistakenly thinking OSS can help them save IT dollars.

2. Total Cost of Ownership (TCO) and OSS: Benefits vs. Risks

Many IT project managers consider only hardware, software, and infrastructure costs when budgeting for new initiatives. However, there are other critical components involved in the cost consideration. TCO refers to all costs incurred during system acquisition and full-cycle operation until its retirement. Acquisition costs include the processes of product selection, system design, purchasing, installation, deployment, and user training. The cost of system operation includes system management, maintenance, repair, user support, data center environment, and other factors that are highly specific to each individual environment [6].

TCO may also include other unexpected components, such as cost of poor performance, unexpected capacity considerations, satisfactory functionality, system availability, ease of user interface, and security. When these critical components fail

or do not adhere to the end-user requirements, the cost of the project increases, thus having a direct influence on the TCO formula. It is not nearly enough to project all the costs associated with system acquisition and operation in order to accurately predict the total cost. The low price of an acquired system does not necessarily lead to cost savings, unless the system performs as expected, satisfies the users, and is available in accordance with service level agreements.

The price of software itself is low relative to TCO. It may represent about 10% of the TCO, with staffing costs adding a huge 50% to 70% portion [8]. This means that despite a common myth that software significantly reduces the overall available IT budget, its price has relatively low importance when compared to tasks following acquisition. This is where OSS may not stack up well against proprietary software. The myth of huge savings, therefore, loses its importance relative to the full-cycle development process. Moreover, TCO is difficult to calculate in the situation of uncertainty associated with lack of support and bug fix guarantee.

It is helpful to know what motivates companies to look for alternative software and doing away with traditional IT purchasing habits. The Emerald Hill Group - the company that manages pubs in Singapore - was able to customize its open source software, tailor it to its needs, and run on a reliable Linux distribution [10]. Gartner's research concludes that Windows XP costs on average 15% to 20% less to own and operate compared to Linux. But despite higher operational cost, the firm derived benefits from customization and running its own software under the company's control and on the firm's own revision schedule.

The Beaumont Health Care system in Ireland is a practical proof of how serious companies can be when it comes to OSS. The organization plans to save over 30 million euros over the period of five years by utilizing only OSS for desktop and front-office applications in addition to traditional infrastructure tools like Linux and Apache [5]. Considering the criticality of system availability in a clinical environment, this step signifies tribute to how far OSS has gone since its inception in terms of reliability and functionality. Beaumont and Emerald Hill Group also represent a new trend in OSS utilization - moving away from using separate software tools and modules and toward complete OSS environments that include more than just basic operating systems and web server applications. In other words, OSS makes its way into enterprise environments.

Many firms find it easier to build what they want out of OSS instead of living with unsatisfactory set of features provided by commercial suppliers. For example, highly individual knowledge management (KM) applications that depend on the users' needs are good candidates for OSS projects. As KM applications change and evolve together with an organization, frequent modifications are required in order to maintain an application up-to-date [4]. Such services may either be unavailable from a vendor or carry prohibitive costs. Education is another area where OSS is a highly desirable option. Low upfront acquisition cost, ability to avoid forced upgrades, flexible platform, and a chance to give students real tools to experiment with make OSS attractive to school district administrators [1]. However, given schools' usually

tight IT operating budgets, the long-term costs might be too high, so it is important to weigh the benefits against ability to maintain the minimum level of staffing needed to keep the systems up and running.

There is a danger that firms and especially non-profit organizations may get too excited about low cost opportunities provided by OSS and forget about TCO considerations. This puts many organizations at risk of successfully implementing projects they cannot support. Firms considering switching to OSS environments from more traditional supported systems should also consider the risk of losing key personnel during critical project phases, expenses associated with migration, and retraining users and support people on the Linux or similar system, along with downtime required to accomplish the goals [7]. Smaller businesses lack both project management expertise needed to accurately calculate TCO and the budget needed to support their desired systems [9].

3. OSS Selection and Decision-Making Framework

The discussion about benefits and disadvantages of OSS can be summarized in two categories: TCO and freedom of choice for product development and maintenance. Some organizations implementing OSS will save money, others will be able to implement the exact kind of system they need to go about their business, and the rest of the firms will likely lose money or run into technical supportability issues [2]. IT managers must understand the savings model when it comes to budgeting technology projects.

While saving 10% to 15% of TCO on software acquisition is a significant achievement for any project manager, it is important to consider potentially higher costs of software selection, development, and maintenance. These costs are inevitable when it comes to OSS that must be modified, customized, compiled, and kept up-to-date. Those firms that employ a large number of IT staff and are in need of heavy customization for their business environments will likely reap the rewards of OSS, as they can save money on software licensing while going through customization efforts no matter what kind of software they own.

Many large firms do not want to accept responsibility for unexpected problems and prefer contracted software to the tools and applications coming from the community. Certainty may be worth the additional expenses in critical corporate business environments. This is just one of the reasons why commercial software's value will not diminish in the near future. It will still mark a sense of relative reliability, accountability, and certainty, assuming that the software vendor is financially stable to remain in business. However, we may also observe a rapid growth of OSS consumption. Some of it is associated with firms' desire to control their IT destiny - one of the best and better justifiable reasons to use OSS. Other reasons include savings, and many firms are on their way to disappointment in this case.

Based on this summary, we may derive the following general decision-making framework for software selection:

1. Clearly state and understand the project goals.
2. Discuss and fine-tune these goals with your end-users; finalize system requirements and ensure commitment to the project, including executive sponsorship.
3. Determine the criticality of the project, system availability and reliability requirements, and dependency of business operations on this particular software application.
4. Investigate whether there are any commercial software products offered on the market that will address the needs of the project. Solicit requests for proposal (RFP) and determine the initial costs.
5. If your project is long-term, critical, requires high system availability and reliable support, and its needs can be satisfied by commercial software, you may think about purchasing the product. OSS should still be considered if commercial product's cost is prohibitive, you expect future justified customization needs, or its features do not address the core project goals. In other words, ensure that all or any of the common factors pointing to possible OSS application are present.
6. If additional exploration is decided on, team up with senior architects and/or developers to run a small pilot project of OSS selection.
7. Determine if the selected software addresses the needs of the project, can rival or exceed commercial software's functionality and security, and represents initial financial savings.
8. Determine what tasks will be involved in the preparation of the selected OSS for final rollout and come up with a cost schedule for these tasks.
9. Compare the costs of development, customization, and installation between the OSS and commercial methods. Plug figures into the overall TCO formula, assuming you have also collected information about other components of the system. If these are not available, perform an analysis and make sure all other major components are present in the TCO calculation.
10. Determine the benefits of ownership and control over OSS relative to dependency on the vendor. The results will represent the intangible OSS project benefits.
11. Weigh the cost savings (if any) against the risks and determine the importance of intangible benefits discovered in the previous step.
12. Consider technical, supportability, operational, and financial risks associated with your software alternatives.
13. Make an educated final decision regarding software selection and architecture.

References

1. Alfonsi, B. (2005, June). Open source in the classroom. *IEEE Distributed Systems Online,* 3.
2. Berger, M. (2002, August 12). Eating the free lunch. *InfoWorld*, 24(32), 1.
3. Bruce, G.L., Wittgreffe, J.P., Potter, J.M.M., & Robson, P. (2005, July). The potential for open source software in telecommunications operational support systems. *BT Technology Journal*, 23(3), 79.
4. Donnelan. B., Fitzgerald, B., Lake, B., & Sturdy, J. (2005, November). Implementing an open source knowledge base. *IEEE Software*, 22(6), 92-95.
5. Fitzgerald, B. & Kenny, T. (2004, January/February). Developing an information systems infrastructure with open source software. *IEEE Software*, 21(1), 50-55.
6. Hawkins, M.W. (2001, June 22). Total cost of ownership: The driver for IT infrastructure management. *Prentice Hall PTR*; http://www.phptr.com/articles
7. Hutlestad, L. (2004, May 31). Linux TCO: Fact or fiction? *VAR Business*, 2012, 54.
8. MacCormack, A. (2003, August 18). The true costs of software. *Computerworld*, 37(33), 44.
9. Prichard, S. (2006, October 4). Why open source stays out of reach small and medium-sized enterprises. *Financial Times*, 6.
10. Vallejo-Yeo, G. (2005, May 14). Linux can save money. *Asia Computer Weekly*, 1.