



Three Strategies for Open Source Deployment: Substitution, Innovation, and Knowledge Reuse

Jonathan P. Allen

► To cite this version:

Jonathan P. Allen. Three Strategies for Open Source Deployment: Substitution, Innovation, and Knowledge Reuse. 6th International IFIP WG 2.13 Conference on Open Source Systems,(OSS), May 2010, Notre Dame, United States. pp.308-313, 10.1007/978-3-642-13244-5_24 . hal-01056047

HAL Id: hal-01056047

<https://inria.hal.science/hal-01056047>

Submitted on 14 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Three Strategies for Open Source Deployment: Substitution, Innovation, and Knowledge Reuse

Jonathan P. Allen¹

¹ University of San Francisco, 2130 Fulton St., CA 94117, USA,
jpallen@usfca.edu

Abstract. As open source software adoption becomes mainstream, the question shifts from whether organizations should use open source, to how organizations can best deploy and use open source. Based on three distinct types of organizational outcomes for open source use, we propose three different strategies for deploying open source: a substitution strategy, an innovation strategy, and a knowledge reuse strategy. Limiting the deployment of open source to a substitution strategy can lead organizations to underestimate the strategic benefits of open source use.

1 Introduction: Is Using Open Source Different?

As open source software adoption by organizations continues to grow, open source is increasingly perceived as a ‘normal’ option, rather than as a strange new technology requiring special justification and extraordinary precautions. In a way, this is a victory for advocates who have fought to have open source software judged by the same criteria as ‘normal’ proprietary software (e.g., [10]). The very term ‘open source’ itself was invented, in part, to downplay the differences between community-built and traditional software, making open source seem more familiar and acceptable for organizational use than ‘free’ (as in ‘freedom’) software [12].

However, in the rush to make open source software appear ‘normal’, there is a risk that the potentially unique benefits of open source might not be fully considered. Forcing a disruptive innovation to compete using existing performance criteria, rather than along new dimensions where it excels, often puts the disruptive innovation at a disadvantage [3]. There continue to be cases where using open source as a direct replacement for proprietary software is easily justified (e.g., [6]). However, it can be difficult to make the case for ‘ripping out’ established proprietary software that ‘already works’ and replacing it with an open source equivalent that the organization has no experience with. In these situations, it would be helpful to have a clear argument for other performance dimensions along which open source use might be superior. It might also be helpful to think of these new performance dimensions not only in terms of justification, but also in terms of strategic use. How would open source allow us to do things differently from proprietary software use? What difference does using open source make?

The case for open source in software development has been made elsewhere (e.g., [7]), along with choices for open source development strategies. Here, we focus on the choices for user organizations that are not primarily in the business of software development. Based on three different types of organizational outcomes, we offer three strategies for open source deployment in user organizations.

2 Open Source Deployment: Three Types of Outcomes

Open source has been seen as a revolutionary, disruptive force for software development (e.g., [2]), but debate continues as to whether the organizational outcomes of open source use are similar to those of traditional proprietary software (e.g., [8]). We find it helpful to distinguish between the typical business benefits that come from using open source as a substitute for proprietary software, and the innovation and knowledge sharing benefits that are unique to open source software.

2.1 Substitution

Substitution takes place when open source software is used to replace the equivalent proprietary software. Studies of open source adoption suggest that organizations are motivated by the desire to replace costly proprietary software with open source equivalents, providing similar functionality and performance (e.g., [4]). Some go further and argue that organizations ignore the ‘ideological’ dimensions of open source—such as having the freedom access to source code—and only focus on practical benefits such as functionality and cost (e.g., [13]).

Table 1. Open source deployment: Three types of organizational outcomes

Type	Activity	Example	Outcomes
Substitution	Open source used to replace equivalent proprietary software.	Microsoft Office is replaced by OpenOffice.	Direct business benefits from software use.
Innovation	Open source used as a platform for creating new applications.	A new product promotion website uses WordPress.	Increased rate of innovation within organizations.
Knowledge reuse	Open source used as a platform for sharing new applications.	A new distribution of Drupal or Joomla is shared among not-for-profits.	Increased rate of innovation sharing between organizations.

We expect that the main outcomes of open source use for substitution will be cost reduction and increased functionality. A typical example of substitution can be seen in the case of an Irish hospital, searching for software that was “zero cost or as cheap as possible.” ([6], p. 54) The main outcomes reported in this case were a 6.5 million

Euro initial purchase savings, and 12 million Euros in total savings over a 5 year period. In other examples of substitution, pure cost savings are not as important as selecting the ‘best technology’ for the job, usually in terms of functionality, reliability, or security (e.g., [16]).

These evolutionary (rather than revolutionary) outcomes from open source software use would be similar to those expected from the use of proprietary software. An open source software package might provide the same or better organizational benefits—return on investment, functionality, security, or standards compliance—as a corresponding proprietary package. For example, a Windows server could be replaced with a Linux server because it provides better total cost of ownership, or Firefox could be chosen as a browser over Internet Explorer for improved security.

2.2 Innovation

The second outcome in our framework, innovation, takes place when open source software is used as a platform or foundation for creating new applications within an organization. Open source software is a ‘generative system’ [25] that allows organizations to create new applications by building on the freely available work of the community. The use of open source leads to increased innovation because of the leverage it provides, its accessibility for experimentation, and its adaptability due to source code access and modular design.

Open source can increase the rate of innovation by providing frameworks and libraries for programmers, such as when Django or Rails are used to develop new web applications quickly. Open source repositories can be used to share software across projects within an organization (e.g., [11]). Open source applications such as Drupal, WordPress, Joomla, or SugarCRM have modular architectures that facilitate innovative new applications with little or no custom programming. One example is a new website built by the City of San Francisco in a few weeks using the WordPress platform, instead of through the usual lengthy development process [1].

The unique aspects of open source licensing allow successful experiments to quickly spread throughout the organization, without having to be constrained by strict licensing terms and their associated costs. Organizations can commit serious resources only after an innovation has proven itself.

2.3 Knowledge Reuse

Knowledge reuse is the “sharing of best practices or helping others to solve common technical problems” ([9], p. 59). As software, open source facilitates knowledge reuse not only through shared repositories of knowledge about facts, but also by sharing procedural knowledge—code that runs business processes.

Knowledge reuse comes from the sharing of organizational expertise through open source software. One type of knowledge reuse comes from creating explicit partnerships or alliances to jointly develop open source business applications, such

as the substance abuse treatment system developed in Maryland and Texas and now adopted by other states [15]. A different type of knowledge reuse comes from the creation of distributions, or versions of open source software that are pre-configured for specific business applications. For example, the CiviCRM project configures open source content management systems for the specific needs of not-for-profit organizations. Other open source projects allow users to easily create and share add-ons for specific business applications, such as plug-ins for WordPress sites.

Knowledge reuse can be seen as the most revolutionary, or disruptive, type of outcome from open source use in organizations. It might seem difficult to imagine that organizations would freely reveal their novel business applications to others. And yet, this is what the research on ‘user-centric’ or ‘democratized’ innovation implies will happen (e.g., [14]), if open source business software becomes widespread. The open innovation literature suggests that user organizations, not enterprise software vendors, might someday provide the majority of innovations, share them freely, and pool their work with other user organizations, as they do in surprisingly many other industries. This could lead to the free sharing of organizational innovations and best practices, through the use of open source software as platforms. Open source may be much more than low-cost software. It could be a mechanism for sharing and reusing organizational knowledge.

3 Three Strategies for Open Source Deployment

Because the types of outcomes for open source use are fundamentally different, we expect that achieving different outcomes will require different strategies.

The substitution strategy is probably the most commonly used today. The substitution strategy is to evaluate and adopt open source software in exactly the same way as proprietary software. The advantage of this approach is that it fits the way organizations already make decisions. The disadvantage is that evaluation and use might not take advantage of the unique strengths of open source software. The substitution strategy might force organizations to ‘rip out’ proprietary software that ‘already works’ for an unproven open source equivalent with roughly the same features. The open source package could have initial cost advantages, but the substitution strategy burdens the open source case with the switching costs.

Table 2. Strategies for open source deployment and use in organizations.

Strategy	Action Plan
Substitution strategy	Replace current software with cheaper and/or better open source equivalents.
Innovation strategy	Focus on new applications or needs that are not being addressed by proprietary software.
Knowledge reuse strategy	Collaborate with projects, or partners, who are already innovating with open source.

A different strategy is to focus on business needs that are not currently being addressed by proprietary software. Open source software can be deployed and used without many of the usual cost and license considerations that limit proprietary software use. If there are many business processes that could be improved by using new applications, but are individually too small to justify a full-scale proprietary software acquisition project, then an innovation strategy might be effective. Once open source has been brought in for ‘experimental’ or ‘prototype’ projects, growing experience and expertise with open source could lead to wider deployment. In the banking industry, Linux servers at first were not sold as a direct substitute. But as the years went by, and Linux servers ‘just worked’, it was easier to make the case for using Linux more widely. Open source applications for business can follow a similar path of guerrilla first, mainstream afterwards.

The knowledge reuse strategy uses open source to find bodies of valuable knowledge (software, and people) that have already been created, and join that community in order to facilitate your organization’s ability to reuse and refine that shared knowledge. In contrast with the innovation strategy, which involves deploying open source software that already exists, the knowledge reuse strategy is an attempt to improve a software project’s usefulness to a user organization through contributions and community interactions. This strategy opens the possibility of obtaining the full benefits of open innovation. And it addresses the risk of not having enough influence on an essential software platform’s future direction, if an organization does not contribute to its ongoing evolution (e.g., [5]).

6 Conclusion: The Promise of Open Source

We expect that, like many new technologies, open source is mostly understood and used in the same ways as the technology that came before it. Open source use that substitutes for proprietary software can have a significant impact by changing cost structures, or by preventing any one competitor from controlling a technology standard. But open source use for innovation can make a dramatic difference as well. Within organizations, it gives IT departments the ability to create new business applications that would never be practical otherwise, possibly dramatically improving the performance of business tasks. Open source use for innovation also allows organizations to launch new products or services that would not have been possible with the license restrictions of proprietary software.

However, the most revolutionary potential for open source use is when organizations decide to jointly develop and deploy open platforms. The extension of democratized innovation [14], generative systems [17], and peer-production [2] to enterprise applications could result in an explosion of knowledge sharing and reuse around basic business processes. When sharing organizational knowledge through software becomes not just ‘a nice thing to do’, but actually the more efficient and effective way to operate, we will have reached an important cross-over point where freely-revealed software becomes the norm, rather than the exception; where the

majority of business software innovations come from the business that use it, rather than from proprietary enterprise software vendors.

7 References

- [1] Allen, J. P. (2010). Open source deployment at the city and county of San Francisco: From cost reduction to rapid innovation. Proceedings of the 43rd HICSS Conference, Kauai, USA.
- [2] Benkler, Y. (2006). The Wealth of Networks: How Social Production Transforms Markets and Freedom. Yale University Press, New Haven.
- [3] Christensen, C. M., and Overdorf, M. (2000). Meeting the challenge of disruptive change. Harvard Business Review, 78 (2), 66-77.
- [4] Dedrick, J., and West, J. (2004). An exploratory study into open source platform adoption. In Proceedings of the 37th Hawaii International Conference on Systems Sciences, IEEE.
- [5] Enkel, E., Gassmann, O., and Chesbrough, H. (2009). Open R&D and open innovation: Exploring the phenomenon. R&D Management, 39 (4), 311-316.
- [6] Fitzgerald, B., and Kenny, T. (2004). Developing an information systems infrastructure with open source software. IEEE Software, 21 (1), 50-55.
- [7] Grand, S., von Krogh, G., Leonard, D., and Swap, W. (2004). Resource allocation beyond firm boundaries: A multi-level model for open source innovation. Long Range Planning, 37, 591-610.
- [8] Kessler, S., and Alpar, P. (2009). Customization of open source software in companies. In Proceedings of the 5th IFIP WG 2.13 International Conference on Open Source Systems, Springer.
- [9] Markus, M. L. (2001). Toward a theory of knowledge reuse: Types of knowledge reuse situations and factors in reuse success. Journal of Management Information Systems, 18 (1), 57-93.
- [10] Open Source for America (2009). Charter for Open Source for America. <http://opensourceforamerica.org/charter> (accessed August 23, 2009).
- [11] Riehle, D., Ellenberger, J., Menahem, T., Mikhailovski, B., Natchetoi, Y., Naveh, B., and Odenwald, T. (2009). Open collaboration within corporations using software forges. IEEE Software, 26 (2), 52-58.
- [12] Stallman, R. (2009). Why “open source” misses the point of free software. Communications of the ACM, 52 (6), 31-33.
- [13] Ven, J., and Verelst, J. (2009). The importance of external support in the adoption of open source server software. In Proceedings of the 5th IFIP WG 2.13 International Conference on Open Source Systems, Springer.
- [14] von Hippel, E. (2005). Democratizing Innovation. MIT Press, Cambridge, MA.
- [15] Wanser, D. (2008). Crossing state lines to build better software. Behavioral Healthcare, 28 (7), 19-23.
- [16] Wheatley, M. (2004). The myths of open source. CIO Magazine, March 1.
- [17] Zittrain, J. (2008) The Future of the Internet—And How to Stop It. Yale University Press, New Haven.