

Chapter XXXVI

Open Source Technology and Ideology in the Nonprofit Context

Jonathan Peizer

Internaut Consulting, USA

ABSTRACT

This chapter contextualizes open source development and deployment in the nonprofit sector and discusses issues of ideology that often accompany it. The chapter separates and defines the ideologies of application development, selection and use, describing the different issues and impacts each creates in the nonprofit context. The purpose of the article is to clearly articulate the unique dynamics of application development and deployment in the nonprofit or social value context and where to apply ideological considerations for best effect.

INTRODUCTION

This chapter contextualizes open source development and deployment in the nonprofit sector and discusses issues of ideology that often accompany it. Open source has intensified the ideological debate over what technology to deploy in a given circumstance. The nonprofit sector, always price sensitive to any technology solution, has embraced the idea of open source as a cheaper alternative to commercial applications. Open source is also viewed by some as embodying the humanistic and cooperative (vs. competitive) philosophy that defines the best practices of the sector.

Open source refers to a program in which the source code is available to the general public for use and/or modification from its original design free of charge. It is typically created as a collaborative effort in which programmers improve upon the code and share the changes within the community (Sacchi, 2002). Open source has come to mean different things to different constituencies. To software programmers it reflects a particular development methodology and philosophy. The more legally minded see it as licensing ideology that more easily allows sharing intellectual property. To users, especially nonprofit institutions with typically limited resources, it means

free software and freedom from dependence on proprietary technology and related service models. In addition to these definitions, there is also a strong ideological lobby that sees open source as the alternative to commercial dominance by any one player in the software industry and as an equalizer with the potential of wresting control away from U.S. predominance in the software industry (Stewart & Gosain, 2001).

Because open source methodology and ideology have become so intertwined it is appropriate to ask if the right debate is taking place around it, particularly in the context of nonprofit implementations of this technology. Ideology and technology cohabit the same plane of existence on three distinct levels:

- **Development ideology:** How is the technology developed?
- **Selection ideology:** Why is the technology chosen?
- **Ideology of use:** What is the technology ultimately used for?

The most important and thorniest ideological consideration is the ideology of use. Unfortunately, far too much time is spent obsessing about the ideology of software selection to meet a particular need and far too little time considering the effects of its application. How software is deployed, particularly in a world that is hypersensitive to global security concerns, has much farther reaching implications and consequences than the ideologies used to create and select it (Kling, 1983).

BACKGROUND

Development Ideology: How is the Technology Developed?

Ideological considerations occur early in the development process. Is software developed for free, on a commercial basis or as a hybrid of the

two (Lerner & Tirole, 2002)? Is an application designed to meet a social mission, a personal interest or a business requirement? On the legal front should applications be fully available to the public for the purposes of modification, or hidden behind proprietary legal constructs? From a standards point of view are considerations purely technical or are the needs of the disabled and disadvantaged taken into account when designing new technology specifications?

Developers ultimately decide why they build applications. They decide if they wish to generate profit, simply sustain ongoing development and maintenance costs or if contributing a piece of code to the world is payment enough for their efforts. In the current reality, lower price points, mass distribution networks, and a proliferation of useful toolsets have allowed software developers a far more significant range of ideological decisions to make when they create software. They have a plethora of commercial and open source languages, tools, operating systems, and even legal frameworks to choose from in order to develop and distribute their creations.

In this new environment it is also far easier to develop tools for the social sector than it ever has been. The advent of the PC in the 1980s made technology affordable for the first time to many nonprofits. The PC created a market for the social sector that in large part did not exist in the costlier mainframe context. In the '90s, the Internet once again lowered the barriers by providing a technology that allowed nonprofits to reach out and extend their constituencies at a far lower cost (Lee, 1997). Open source tools have unlocked even more development opportunities for this market. They have spurred commercial software developers to rethink their price structures in order not to lose this relatively new market consisting of literally millions of social purpose nonprofits, educational institutions and health facilities globally.

Developers of commercial software maintain a straightforward profit-based ideology for any market they sell to. However, that does not preclude

10 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/open-source-technology-ideology-nonprofi/21209

Related Content

Should Open-Source Technology be used in Design Education?

Tom Page (2015). *International Journal of Open Source Software and Processes* (pp. 17-30).

www.irma-international.org/article/should-open-source-technology-be-used-in-design-education/170474

Weaving a Semantic Web Across OSS Repositories: Unleashing a New Potential for Academia and Practice

Olivier Berger, Valentin Vlasceanu, Christian Bac, Quang Vu Dangand Stéphane Lauriere (2010).

International Journal of Open Source Software and Processes (pp. 29-40).

www.irma-international.org/article/weaving-semantic-web-across-oss/44970

Open Growth: The Impact of Open Source Software on Employment in the USA

Roya Ghafeleand Benjamin Gibert (2014). *International Journal of Open Source Software and Processes* (pp. 16-49).

www.irma-international.org/article/open-growth/104678

Replacing Proprietary Software with Open Source Software: Implications

Albert Akyeampong (2015). *Open Source Technology: Concepts, Methodologies, Tools, and Applications* (pp. 1356-1368).

www.irma-international.org/chapter/replacing-proprietary-software-with-open-source-software/120973

Translating Traditional Writing Process Tools to Digital Ones: Integrating Digital Writing in K-12 Classrooms

Keri R. Franklinand Kathy Gibson (2015). *Open Source Technology: Concepts, Methodologies, Tools, and Applications* (pp. 1406-1420).

www.irma-international.org/chapter/translating-traditional-writing-process-tools-to-digital-ones/120977