

Open Source Software and the New Zealand Education System: A Response to Roy

RICHARD WYLES

OPEN POLYTECHNIC OF NEW ZEALAND
LOWER HUTT, NEW ZEALAND

INTRODUCTION As the project leader of the New Zealand Open Source Virtual Learning Environment (NZOSVLE) project, it may appear difficult for me to be dispassionate about the strengths or possible pitfalls and weaknesses of the open source software development paradigm. However, open source is a generic term and does not automatically equate to good software regardless of the philosophical persuasions of proponents. If your purpose is to deploy mission critical application software then there are a number of considerations before embarking on the open source direction.

Why is the consideration of open source important? Software is ubiquitous in twenty-first-century developed societies. Software now permeates all aspects of our lives, from farm production and the telephones we use to the education we receive or deliver. Equitable access to educational technology is a key issue and open source technology is part of the answer. In 2003 much of the tertiary education sector was without access to a robust, fully featured Learning Management System (LMS). Now, every institute of technology and polytechnic in New Zealand has a production LMS. Open standards also underpin equity since no citizen should be forced to use the software of any particular vendor.

FREE VS OPEN Richard Stallman started the Free Software Foundation (FSF) in the mid-1980s with the GNU project to create an operating system. Although the most widely used license in open source software projects is the GNU General Public License, there are some important nuances between Stallman's original conception of "free software" and the common notions of what open source stands for. It is reasonable to sum up the split by describing Stallman's position as holding a strong moralistic and political stance relating to the freedom to use, modify, and redistribute software code.

The term "open source" began in the late 1990s and was viewed as a more corporate-friendly term to describe a software development methodology. Part of the rationale for new terminology is the understandable confusion over the word "free." "Free software" is a matter of liberty, not price. To understand the concept, you should think of "free" as in "free speech," not as in "free beer" (Free Software Foundation, n.d.).

The freedom that Stallman extols is consistent with the liberty to teach, pursue, and share knowledge common in academia.

The advances in all of the arts and sciences, indeed, the sum total of human knowledge is the result of the open sharing of ideas, theories, studies and research. Yet throughout many school systems, the software in use on computers is closed and locked, making educators partners in the censorship of the foundational information of this new age. (Vessels, 2001)

The emergence of open source is certainly intertwined with higher education, and it is the tenets of academic freedom that often support the application of, and experimentation with, open source technologies within educational institutions.

QUALITY Open source now plays a major part in mainstream information technology economic activity and has started to dominate some market areas. Anyone who surfs the Web will visit sites using open source software since the Apache Web server dominates the server market. Google, Amazon, and Ebay use open source technologies and global technology corporates such as IBM, Sun Microsystems, and Novell are building successful commercial models around the open source paradigm. Much of the success is also obscured by the reality that it is small reusable scripts, compilers, debugging tools, and building blocks that are re-used for bespoke or in-house solutions. In this sense, open source underpins an enormous amount of technology-based economic activity.

However, for application software the success of open source still appears confined to relatively few projects. "People think just because it is open-source, the result is going to be automatically better. Not true. You have

to lead it in the right directions to succeed. Open source is not the answer to world hunger" (Torvalds, n.d., cited in Bezroukov, 1999). There may be more than 130,000 open source projects registered on SourceForge but very few of them are active or will flourish into successful projects (Open, but Not as Usual, 2006).

In this harsh Darwinian context where only a few survive, what characterises the successful open source projects? This is a difficult question as a successful open source community is as complex as all the human interactions and community dynamics of which it is comprised. It is easy to postulate that no two open source projects are the same.

In general, as the research from Stürmer (2005) describes, positive preconditions include great initial source code; meeting an identified need, level of innovation, or novelty factor; the programming language used; and visibility and presence on a collaboration platform such as EduForge or SourceForge.

The human element cannot be underestimated. Leadership skills and behaviours are paramount. The leader needs to not only be a highly skilled developer with a clear vision to attract other skilled contributors, but also must exhibit the personality traits necessary for leading or being the figurehead for an essentially voluntary community. It is small wonder that many open source initiatives fail to reach a critical mass of participation.

MYTHS AND REALITIES The suggestion that the open source movement is analogous to a utopian society is as erroneous as the rose-tinted notion that most open source software

is the output of a group of altruistic hobbyist programmers undertaking some niche activity. The motivations of individuals and the structures within open source communities tend to reflect the complexities of any small society or grouping. While the underlying ethos advanced is one of collaboration, the reality in many successful open source projects can be described as a meritocracy centred by a strong-minded but charismatic leader. "Open source may sound democratic, but it isn't. At the LinuxWorld Expo on Wednesday, leaders of some of the best-known open source development efforts said *they function as dictators*" (Torvalds, n.d., cited in Bezroukov, 1999).

The, presumably benevolent, dictatorship that Linus Torvalds expresses is tempered by the underlying freedom for anyone involved in an open source project to fork the code and start a new community. A project fork is when the community splits and one group takes a copy of the code and develops it in a new direction. Therefore, successful open source projects are often characterised by a type of servant-leadership. While assertive, leaders of open source projects necessarily have a high profile within the community and rely on persuasion and their own coding prowess to convince others to follow a particular development roadmap.

The community consists of a meritocracy which is fluid in its make-up as interested parties come and go from the project. However, at the heart of seemingly open communities is a small close-knit group of core developers who are trusted with access to the version control system of the codebase. "Codebase" is a term that describes the collection of source code used to build a particular software

application. Weber (2004) likens open source developers to artisans, in contrast to the Fordist development methodology typical in commercial software projects.

Beyond this core, there is a wider community of users which may discover bugs, contribute ideas, swap experiences, ask for help, and of course download software and its updates. A vibrant community of users requires neither funding nor encouragement in their use and support of a project. It is self-interest that drives participation in open source communities.

Moodle has a large and diverse user community with more than 100,000 registered users on Moodle.org, although it is fair to point out that many would have low levels of participation. In larger projects such as Moodle, it is atypical for much of the code to be developed by volunteers, rather than employees of companies or educational institutions that see benefits accruing from the project (Open but Not as Usual, 2006).

The reality that open source projects may require high-level expertise is a moot point. Software development at a higher level, when open or closed, requires expertise and this is often paid for. However, a key benefit of accessible software is that those interested in learning software development can hone their skills. Surely, a great way to learn how to write good computer programs is to study what has been written by others, and then perhaps even tinker with improvements even if that means learning through mistakes.

SELECTING AN OPEN SOURCE LMS

In hindsight, the selection of Moodle by the NZ Open Source VLE project now looks relatively simple. As of March 2006,

Moodle has more than 10,000 registered installations worldwide. However, in early 2004 the landscape was different. The initial proposal for funding used Moodle as an example of an open source LMS and cited its then installation base of 350.

The selection process was multi-staged, took several months, and needed to be impartial and robust. The first stage was an initial evaluation using a 22-point Likert scale checklist approach under the broad categories of “fitness for purpose,” architecture, usability, standards compliance, cost of ownership, and strength of community. Whether or not a project had existing or ongoing funding was not part of the selection process, as this factor on its own has little relationship to quality in the open source development model.

The second phase focused on considerations for online pedagogy, particularly on system flexibility and communication tools. This phase leveraged the timely work of Britain and Liber (2004). At this point we were able to shortlist three LMSs and undertake an in-depth technical evaluation. The inherent transparency of open source is of great benefit, as the internal code structures can be closely examined with due focus on security, scalability, and extensibility.

The stated goals of the NZ Open Source VLE project were to establish an open systems and open standards framework, deliver flexibility that accommodates different pedagogical approaches, reduce barriers to entry and barriers to collaboration, provide a catalyst for innovation, and reduce the total cost of ownership.

The NZ Open Source VLE project has certainly never implied that the cost of ownership of an open source LMS could ever be free. However, we continue to stand by the claim of being able to “dramatically reduce the total cost of e-learning infrastructure at a system-wide level, by reducing the wasteful duplication of infrastructure, licensing fees, and upgrade costs” that was stated in the New Zealand Open Source VLE application for funding to the eLearning Collaborative Development Fund in 2003. Empirical evidence strongly supports the claim. The promise for an economically sustainable ICT investment pathway for New Zealand education using open source technologies is evident throughout the sector, particularly the institutes of technology and polytechnics where Moodle has become the most widely deployed LMS during an eighteen-month period.

Cogent evidence is available by exploring individual cases. In 2003 Nelson Marlborough Institute of Technology established a project to select and implement a commercial LMS.

However, up-front costs of hardware, licence and local technical support proved too great a barrier in difficult financial times and we lacked experience or confidence in utilising Open Source systems. The advent of the NZOSVLE project has changed all this. Moodle is a highly functional, stable and relatively intuitive LMS compared to many of the commercial products. An external service provider now hosts our installation of Moodle and the quality of the support available via the NZOSVLE project and the wider Moodle user community has

been outstanding. Rather than pay for expensive hardware and license fees, a greater percentage of available funds have been able to be used to establish an internal support team. (Sturrock, 2006)

Time, effort, and funds will continue to play a major role in the success in developing and using open source technologies, but associated economies from shared infrastructure, services, and zero licensing will become more apparent. Rather than the costs being hidden, they become more transparent by being directly related to the services received.

RISK AND REWARD Open source communities do not equate to ideal societies. Human foibles are as much on show in open source communities as anywhere else. As such, open source projects do not necessarily mean quality outcomes. Often quality correlates with the vision, leadership, and competencies of the project's figurehead, such as Martin Dougiamas (Moodle) or Linus Torvalds (Linux).

There are now frameworks to measure quality in open source projects. The Business Readiness Rating initiative provides an unbiased source for determining whether specific open source software is mature enough to adopt (see <http://www.openbr.org>).

However, while due care is important, and selection of open source solutions encompasses risk, it is risk worth embracing. Risks and benefits go hand in hand, particularly with software. The benefits of getting open source right are enormous, especially for learners of the future.

Open source helps equitable access to technology. First, it can be contextualised for local conditions thereby enhancing access. Secondly, it can be modified to address particular niche needs unserved by commercial operations. Thirdly it can dramatically reduce the total cost of ownership. Open source is also driving the tenets of interoperability and thereby delivering more choice and, with that, more equitable access. There is an inherent tension for proprietary vendors to move towards open standards of their own volition. Finally, open source as a production methodology enables stakeholders to be far more involved and to collaborate, innovate, and contribute to the outputs rather than being passive end-users. The level of participation is a choice each stakeholder makes.

REFERENCES

- Bezroukov, N. (1999). A second look at the cathedral and the bazaar. *First Monday*, 4(12). Retrieved April 14, 2006, from http://www.firstmonday.org/issues/issue4_12/bezroukov/
- Britain, S., & Liber, O. (2004). A framework for the pedagogical evaluation of virtual learning environments. Retrieved March 23, 2004, from http://www.jisc.ac.uk/uploaded_documents/VLE%20Full%20Report%2006.doc
- Free Software Foundation (n.d.) *The free software definition*. Retrieved April 15, 2006, from <http://www.fsf.org/licensing/essays/free-sw.html>
- Open, but not as usual. (2006, March 16). *Economist*. Retrieved April 12, 2006, from http://www.economist.com/business/displaystory.cfm?story_id=5624944
- Stürmer, M. (2005). *Open source community building*. Unpublished master's thesis. Retrieved April 18, 2006, from <http://opensource.mit.edu/papers/sturmer.pdf>
- Sturrock, D. (2006). *Case study: Nelson Marlborough Institute of Technology*. Retrieved April 16, 2006, from <http://>

moodle moot.org.nz/moodle/mod/
resource/view.php?id=7

Vessels, T. (2001). *Why should open source software be used in schools?* Retrieved April 13, 2006, from <http://edge-op.org/grouch/schools.html>

Weber, S. (2004). *The success of open source*. Cambridge, Massachusetts: Harvard University Press.

*From early 2004, **Richard Wyles** has led the New Zealand Open Source Virtual Learning Environment Project, involving a consortium of 20 universities, institutes of technology, and polytechnics. His work with the Moodle Learning Management System, Eduforge, EPrints, and Elgg has gained him international recognition in e-learning technology circles. His current focus is on developing a national e-learning network in New Zealand. Wyles is also a director and co-founder of Flexible Learning Network Ltd.*