Open Governance in Higher Education

Extending the Past to the Future

By Patrick Masson

Open educational resources, open content, open access, open research, open courseware—all of these open initiatives share, and benefit from, a vision of access and a collaborative framework that often result in improved outcomes. Many of these open initiatives have gained adoption within higher education and are now serving in mission-critical roles throughout colleges and universities, with institutions recognizing reduced costs and/or increased value related to access and quality. If such a social organization of cooperation and production (i.e., openness) does indeed enhance the creation, delivery, and management of the critical products and services required by an institution of higher education to fulfill its mission, the next logical question is whether open development and governance can have a broader applicability—beyond software, resources, courses, learning objects, and content. Can understanding the principles and practices that govern open-source initiatives, and the communities of practice that
manage them, provide a potential reference model for institutions of higher education? Can colleges and universities improve administrative and academic planning and decision-making processes within institutional governance through these open principles and practices?

Our Introduction to Openness

According to the Open Source Initiative (OSI): “The ‘open source’ label was invented at a strategy session held on February 3rd, 1998 in Palo Alto, California.” The idea was inspired by the seminal work “The Cathedral and the Bazaar,” first presented in 1997 by Eric Raymond, whose analysis, “centered on the idea of distributed peer review, had an immediate and strong appeal both within and (rather unexpectedly) outside the hacker culture.” Originally, Raymond believed there was “a certain critical complexity above which a more centralized, a priori approach was required” and that the most important software “needed to be built like cathedrals, carefully crafted by individual wizards or small bands of mages working in splendid isolation.”

One of the earliest open-source technologies to enter the campus portfolio was Linux, in the 1990s. As Raymond noted: “Linus Torvalds’s style of development—release early and often, delegate everything you can, be open to the point of promiscuity—came as a surprise. No quiet, reverent cathedral-building here—rather, the Linux community seemed to resemble a great babbling bazaar of differing agendas and approaches (aptly symbolized by the Linux archive sites, who’d take submissions from anyone) out of which a coherent and stable system could seemingly emerge only by a succession of miracles.” Raymond, who had been an active developer of open-source tools for Unix, added: “I thought I knew how it was done.” However, “the fact that this bazaar style seemed to work, and work well, came as a distinct shock.”

Like many others, Raymond understood the value—the feasibility—of open source yet did not at first see the viability in open, community-driven governance and decision making.

In 2003 Gartner, offering guidance to CIOs assessing the appropriateness of Linux, pointed to “concerns regarding mission-critical readiness” yet predicted: “Linux will achieve mainstream adoption by 2006.” Technologies like Linux and Apache, which lived primarily in the data center, were part of the first wave of open source to penetrate the campus due to their relatively small level of exposure: not many faculty, students, or administrators cared too much about what operating system the ERP systems or websites ran on—just that they ran. This attitude was expressed in the 2004 Campus Computing Survey: “The data suggest an ‘affirmative ambivalence’ about Open Source administrative applications among campus technology officials, says Green [Kenneth C. Green, founding director of The Campus Computing Project]. He notes that Open Source applications are already a common part of the ‘back room’ IT operations on campuses and in corporations. The long-term questions about Open Source, says Green, involve the costs and benefits of developing and supporting complex administrative applications intended for the desktops and laptops of non-technical users.”

Technologies like Linux and Apache, which lived primarily in the data center, were part of the first wave of open source. Of promiscuity—came as a surprise. No quiet, reverent cathedral-building here—rather, the Linux community seemed to resemble a great babbling bazaar of differing agendas and approaches (aptly symbolized by the Linux archive sites, who’d take submissions from anyone) out of which a coherent and stable system could seemingly emerge only by a succession of miracles.” Raymond, who had been an active developer of open-source tools for Unix, added: “I thought I knew how it was done.” However, “the fact that this bazaar style seemed to work, and work well, came as a distinct shock.”

Like many others, Raymond understood the value—the feasibility—of open source yet did not at first see the viability in open, community-driven governance and decision making.

In 2003 Gartner, offering guidance to CIOs assessing the appropriateness of Linux, pointed to “concerns regarding mission-critical readiness” yet predicted: “Linux will achieve mainstream adoption by 2006.” Technologies like Linux and Apache, which lived primarily in the data center, were part of the first wave of open source to penetrate the campus due to their relatively small level of exposure: not many faculty, students, or administrators cared too much about what operating system the ERP systems or websites ran on—just that they ran. This attitude was expressed in the 2004 Campus Computing Survey: “The data suggest an ‘affirmative ambivalence’ about Open Source administrative applications among campus technology officials, says Green [Kenneth C. Green, founding director of The Campus Computing Project]. He notes that Open Source applications are already a common part of the ‘back room’ IT operations on campuses and in corporations. The long-term questions about Open Source, says Green, involve the costs and benefits of developing and supporting complex administrative applications intended for the desktops and laptops of non-technical users.”

The debate over open source as a reliable development methodology that could deliver quality products expanded with the growth of the World Wide Web as a delivery platform and with the introduction of Learning Management Systems (LMSs), which took advantage of these Internet-based technologies. According to a summary of a 2005 Alliance for Higher Education Competitiveness (A-HEC) survey of primarily higher education IT, instructional support, administrative, and executive staff: “Vulnerability of a category [to competition from open source] seemed to track inversely to how well penetrated or established a category is with existing commercial products. The only major inconsistency with this was the strength of the [LMS] initiatives, Sakai and Moodle, in a category that according to previously published studies is highly penetrated by existing products. This indicates a potentially interesting competitive dynamic developing between the incumbents and the open source initiatives.” In addition, the analysis revealed an emerging interest in portals (e.g., uPortal), e-portfolios (e.g., OSPI), content management (e.g., LionShare), and assessment tools (even though there was no open-source initiative known in the last category at the time of the survey).

With applications like the LMS and others now touching the desktop, rather than being hidden in the data center, discussions were no longer limited to IT professionals; faculty, program managers, and administrators now offered their functional, technical, and business requirements. Interestingly, many of the major LMS platforms grew out of higher education: ANGEL Learning was originally developed by Indiana University; Blackboard came partly out of Cornell University; Sakai was developed as a consortium of higher education institutions; Moodle evolved from within Curtin University; and WebCT grew out of the University of British Columbia. Perhaps these
origins are indicative of the “build versus buy” construct, when purchasing an out-of-the-box solution, such as an LMS, was not possible and therefore required the internal development resources of institutions. As the LMS matured, economic opportunities emerged, capitalizing on the credibility of university-sponsored efforts while offering an established business model that removed local, and redundant, requirements to produce and manage a homegrown LMS. For those accustomed to commercial development and support—a contracted relationship with defined direction and support levels—the debut of product-specific companies and the consolidation of the market seemed attractive, promising stability, and a “single throat to choke” in case something went wrong. The January 2007 EDUCAUSE Center for Applied Research (ECAR) Bulletin “Considering Open Source: A Framework for Evaluating Software in the New Economy” recognized this transition yet offered a counterpoint:

The debut of product-specific companies and the consolidation of the market seemed attractive, promising continuity. 

Importantly, in addition to technical and financial evaluations—such as stability, pace of development, support, and costs—the bulletin emphasizes assessing the organization that produces the software: “Selecting a product is not only a matter of technical evaluation but also of due diligence on evaluating the company or the community” by considering “the experiences of others,” “the available support options,” and the “reputation of the company or community.” Clearly, the principles and practices of the organization managing the vision and direction, development life-cycle, and support levels are critical when making the decision to adopt any technology.

Barriers to Adoption: Open Content or Open Communities?
In 2006, the Alliance for Higher Education Competitiveness (A-HEC) polled 200 officials responsible for software selection on their campuses and published “Best Practices in Open Source in Higher Education Study: The State of Open Source Software.” This study included the “top reasons why open source has not yet been seriously considered.” Barriers to adoption, as identified by those who responded, included the following:

- Most organizations are satisfied with current, non-open source products (and therefore have no reason to change).
- Uncertainty over actual costs: “Many respondents to the survey . . . cited lower total ownership costs as their reasons for switching to open source software,” but others noted that with open-source software, costs simply “shift away from acquiring the software and toward hiring more IT staff.”
- The perception that open-source software isn’t available “out-of-the-box” and requires local customization in order to meet the campuses needs: “The biggest thing is it takes more physical labor to implement open source because it isn’t pre-packaged,” reported the study’s authors. “You have to have software developers that can make this stuff work.”
- Little confidence that open source would reach a tipping point within higher education, with many “taking a wait-and-see approach.”
A lack of vendor—that is, contractual—support: “Most in the survey said they would prefer to work with open source through a commercial vendor.”

- Ambiguity regarding the direction and stability of open-source technologies and tools: “It’s not quite ready for prime time.”

Despite these barriers, campuses did not reject open source—in fact, far from it. Since the time the label “open source” was first introduced, other open initiatives “outside the hacker culture” emerged (“unexpectedly”), including open access and open content in 1998, open courseware and open educational resources in 2002, and open research/science in 2005. A majority of campuses apparently perceived some potential value in open projects and presumably in the communities that facilitate their design, development, distribution, and support. According to the Campus Computing Survey from 2006, the same year as the A-HEC report: “Over half (53.9 percent, compared to 51.9 percent in 2004) agree ‘Open Source will play an increasingly important role in our campus IT strategy.’” So although concerns were expressed, these issues did not appear to detract campuses from recognizing the feasibility of open-source technologies or the viability of the communities that manage them. Indeed, today we see a continued comfort level with open source among campuses of all sectors. The 2009 Campus Computing Survey reported the percentage of institutions that agree “open source will play an increasingly important role in our campus IT strategy” has increased steadily since 2004 (when this question was first asked), averaging around 65 percent among public and private research and four-year colleges and universities. Campuses’ continued comfort level with open source, as indicated through self-reporting, and the variety of open-source applications within specific product categories (i.e., specific products within one domain: operating systems like Mandriva, Red Hat, SUSE, Ubuntu), as well as within the broader higher education market (e.g., learning management systems, content management systems, e-portfolios, e-mail, office productivity suites), would appear to bolster advocates’ claims that open-development practices have matured to enterprise class.

Extending the Open Model

If, as advocates state, the principles and practices behind open-source initiatives provide greater flexibility, fewer errors, more features, quicker pace of development, increased closeness to customers, lower costs, and greater stability, reliability, and continuity, why wouldn't other initiatives choose to organize in the same manner? One reason is the continuing debate over the feasibility of openness (i.e., the suitability of open-source products for higher education) and the viability of openness (i.e., the practicality of specific open principles and practices to sustain and advance those products). There is an important difference between an open-source product (e.g., uPortal) and the organization that produces and manages that product (e.g., Jasig). In an effort to provide both feasible products and viable development communities, a variety of governance models have emerged among open initiatives, with sometimes subtle and in other cases dramatic differences for operational decision making and developmental direction. For example, the uPortal Steering Committee, “the governing body responsible for the strategic direction and operational oversight of the uPortal project,” is composed of two members elected by the Jasig membership, two members selected by the uPortal developers, one member appointed by the Steering Committee, and two members representing the Jasig board. Another open-source portal project, Liferay, is led by a private corporation of founding partners “driving innovation for the benefit of enterprises around the world.” These two examples—representing a community versus a corporate approach—highlight what I believe to be a point of ambiguity and thus contention around the viability of open source: open-source governance and, fundamentally, openness itself. What is openness, and how have open communities organized themselves to deliver the quality of products required by higher education—whether that be open-source software or open educational resources?

Colin Currie tried to answer the question “What Is Openness, Anyway?” in a 2009 EQ article. He referred to Wikipedia, which offers a definition: “Openness is a very general philosophical position from which some individuals and organizations operate, often highlighted by a decision-making process recognizing communal management by distributed stakeholders (users/producers/contributors) rather than a centralized authority (owners, experts, boards of directors, etc.).” In addition, Currie pointed to “EDUCAUSE Values: Openness,” which states:

A central pillar of the academic community is its commitment to the free flow of information and ideas. This commitment to sharing is essential to scholarly discovery and innovation. It is also central to helping learners...
engage, absorb, and apply knowledge in order to advance personally and academically. Finally, this commitment to openness provides the foundation for leveraging resources, both within and among institutions, to strengthen the creation, transmission, and preservation of knowledge.

The academic—and, by extension, social—value of unfettered intellectual exchange finds expression in technologies, applications, and approaches that foster sharing, collaboration, and open access to knowledge and resources. By actively encouraging collaboration, customization, and experimentation, open environments—whether social, cultural, or technological—best facilitate the free flow of information and ideas on which discovery, innovation, and high-quality higher education depend.16

“Unfettered intellectual exchange” appears to be the central tenet for openness as a value for EDUCAUSE. I would suggest that these two statements respectively offer excellent definitions for “open” as a governance model and for “transparent” as an approach to communication. Transparent organizations promote unfettered intellectual exchange (i.e., communications) by providing access to all of the organization’s current discussions and activities and to the considerations that led organizational and operational decisions; yet authority for decision making within the organization, although influenced through collaboration, remains centralized and hierarchical—undertaken by “wizards or small bands of mages.” Open organizations go one step further and decentralize governance so that decision-making authority is distributed to the community. Open organizational administrators serve as facilitators who promote the principles and practices that enable openness, rather than as leaders who direct policy, development, and operations. All open organizations are transparent; not all transparent organizations are open. This distinction is important, with practical implications around the processes for identifying needs, analyzing resources, assessing value, setting priorities, defining outcomes for success, and carrying out all of the other obligations associated with governance. A definition that negotiates between open as a governance model and transparent as an approach to communications provides potential participants with a framework for evaluating “openness,” ensuring that their expectations (and opportunities) are aligned with their organization.

If we agree, then, for the sake of this discussion, that the benefits ascribed to open initiatives are feasible, what are the principles, values, and practices that make openness viable, and how should open organizations be organized and operated? Table 1 (see page 54) lists bazaar-style and agile principles,27 along with the values and practices that can be derived from these principles, to provide some evaluation criteria for the assessment of organizational openness.

**Enabling Openness**

Today, most would agree that Internet-based communications, social networking, and collaboration tools have empowered independent, diverse, and distributed individuals and groups, allowing them to find and work with one another at a pace never before possible. The P2P Foundation notes: “The key organizing pattern of our global culture is shifting from a top-down hierarchical pyramid to a distributed, self-organizing network.”28 Indeed, peer-to-peer co-creation appears to be the primary motivation for the growing number of organizations choosing openness. In “Open Source 2010: Reflections on 2007,” Brad Wheeler noted: “Open source’s greatest appeal is the leveraging of resources of the partners and the community for shared value creation.”

The confluence of technology provides greater opportunities to find relevant tools and services and like-minded communities. In fact, many of the open-source initiatives familiar to those of us in higher education capitalize on the convergence of such technologies. Distributed (participants are geographically dispersed), decentralized (authority is not confined to within one organization), and remote (development and support is not dependent on local resources) are no longer organizational or operational barriers. Sakai, for example, was founded as a consortium of several universities (distributed) and is led by a diverse group from the participating campuses as well as commercial partners (decentralized) with development support, while its funding derives from educational grants and from academic and commercial partners (remote). In addition, as noted earlier, the new development processes known as agile software development promise greater responsiveness to users through iterative development, in which requirements and solutions evolve through collaboration among self-organizing cross-functional teams. These two factors—technology convergence and new approaches to production—provide a potential platform for the needs identification and development required for openness.

As articulated above, and demonstrated through adoption rates across higher education, many have come to recognize the value, the feasibility, of open
TABLE 1. OPEN PRINCIPLES, VALUES, AND PRACTICES

<table>
<thead>
<tr>
<th>Bazaar Principle</th>
<th>Agile Principle</th>
<th>Values &amp; Practices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Every good work of software starts by scratching a developer’s personal itch.</td>
<td>Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.</td>
<td>Bottom-up</td>
</tr>
<tr>
<td>To solve an interesting problem, start by finding a problem that is interesting to you.</td>
<td>Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.</td>
<td>No heroes</td>
</tr>
<tr>
<td>Good programmers know what to write. Great ones know what to rewrite (and reuse).</td>
<td>Working [technology] is the primary measure of progress.</td>
<td>Evidence-based</td>
</tr>
<tr>
<td>Release early. Release often. And listen to your customers.</td>
<td>Deliver working [technologies] frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.</td>
<td>Frequent testing</td>
</tr>
<tr>
<td>Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone.</td>
<td>Our highest priority is to satisfy the customer through early and continuous delivery of valuable [technologies].</td>
<td>Iterative and incremental development</td>
</tr>
<tr>
<td>Treating your users as co-developers is your least-hassle route to rapid code improvement and effective debugging.</td>
<td>Business people and developers must work together daily throughout the project.</td>
<td>Transparency</td>
</tr>
<tr>
<td>If you treat your beta-testers as if they’re your most valuable resource, they will respond by becoming your most valuable resource.</td>
<td>The best architectures, requirements, and designs emerge from self-organizing teams.</td>
<td>Openness</td>
</tr>
<tr>
<td>Smart data structures and dumb code works a lot better than the other way around.</td>
<td>Continuous attention to technical excellence and good design enhances agility.</td>
<td>Honesty</td>
</tr>
<tr>
<td>The next best thing to having good ideas is recognizing good ideas from your users. Sometimes the latter is better.</td>
<td>Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.</td>
<td>Emergent design</td>
</tr>
<tr>
<td>Often, the most striking and innovative solutions come from realizing that your concept of the problem was wrong.</td>
<td>At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.</td>
<td>Maturity</td>
</tr>
<tr>
<td>Perfection (in design) is achieved not when there is nothing more to add, but rather when there is nothing more to take away.</td>
<td>Simplicity—the art of maximizing the amount of work not done—is essential.</td>
<td>Just enough</td>
</tr>
<tr>
<td>Provided the development coordinator has a communications medium at least as good as the Internet, and knows how to lead without coercion, many heads are inevitably better than one.</td>
<td>The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.</td>
<td>Collaboration</td>
</tr>
</tbody>
</table>


...
Open Governance in Higher Education: Extending the Past to the Future

Patrick Masson (pmasson@umassonline.net) is Chief Technology Officer of UMassOnline, the online consortium of the University of Massachusetts.

© 2011 Patrick Masson. The text of this article is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License (http://creativecommons.org/licenses/by-sa/3.0/).