Software and Collaboration in Higher Education: A Study of Open Source Software

July 26, 2006

Authors:
Paul N. Courant, Principal Investigator
Rebecca J. Griffiths*

Funders:
The Andrew W. Mellon Foundation, Carnegie Mellon University, Foothill-De Anza Community College, Marist College, Indiana University, the University of Michigan, Stanford University, the University of North Carolina, and The William and Flora Hewlett Foundation

***We are grateful to Matthew Rascoff for excellent field work, research assistance and extended discussions of this work, and we are grateful to Michael Carter and Kevin Guthrie for their time, expertise and valuable commentary on earlier drafts. Barnaby Gibson was a helpful guide to us in legal matters, and of course we take full responsibility for any remaining errors.**

Table of Contents

Executive Summary

As the use of open source software (OSS) has taken off over the past decade, there has been increasing interest in the potential of open source to address longstanding concerns in the higher education community regarding the cost and performance of commercial software products. A common view is that existing proprietary options do not have the features required or allow for cost-effective customization. Many administrators are concerned that academic institutions are ceding too much control for mission-critical tasks to an increasingly concentrated field of commercial vendors. OSS advocates argue that open source software can address these issues, and moreover that higher education has proven it can produce high quality and innovative software. However, adoption of OSS may well be hindered by uncertainty about future support for and improvements in the software. Furthermore, without coordination, there is likely to be wasteful duplication both of development efforts and of governance structures, and suboptimal attention will be paid to issues of interoperability.

In October, 2005, The Andrew W. Mellon Foundation and The William and Flora Hewlett Foundation convened a group of leaders in higher education to discuss the possibility that is the subject of this report – that the creation of an organization, referred to here as the "Organization for Open Source Software" (OOSS)[1], to coordinate and support OSS in higher education would be of value. The group did not reach an answer, but there was general agreement that the issue was worth exploring further. Subsequently, this study was launched to test the following propositions:

- There is a sizable community of users who are deeply dissatisfied with the cost and performance of currently available options, and this is a subject of concern at the leadership level of academic institutions.
- Open source software is a viable solution to this problem, and college and university leaders are receptive to it.
- There is significant demand for certain software products that is not being met, suggesting that there is a market failure and perhaps a need for some concerted action to address that failure.

We also attempted to envision what an organization could do to promote solutions to problems in higher education via the use of open source.

Our primary research method was conducting interviews with a broad range of constituents, including: senior leaders, such as presidents, provosts, CIOs and CFOs from a diverse set of institutions; project leaders and developers of open source software in and outside of higher education; commercial companies that are engaged with or compete against OSS; and assorted organizations involved in various aspects of IT in higher education. In total, we consulted with over 60 people and visited or spoke with staff at over a dozen colleges and universities across the country. In addition, we gathered and consulted information from a wide range of sources, including: surveys on usage of open source and IT spending in higher education, articles and papers chronicling the progress of open source, discussion boards of prominent OSS projects, and books exploring the movement's success by authorities such as Steven Weber and Eric Raymond.

We found a considerable amount of evidence attesting that many college and university leaders are dissatisfied with the cost and performance of software, and that this is a matter of significant concern to them. The areas of dissatisfaction can be grouped under three headings: (1) Cost. Many institutions have spent millions or even hundreds of millions of dollars implementing and customizing administrative systems, with significant costs incurred each time the vendor phases out old versions of

[1] We choose this name as a placeholder, in the fervent expectation that should such an organization be born, it will be graced with a nicer name.

the software. (2) Performance. Many commercial products are not well tailored to the needs of higher education, and because they are proprietary it is difficult and expensive to make the desired modifications. (3) Control. College and university leaders are concerned that consolidation in the sector may result in commercial software vendors having unfair pricing leverage in their negotiations with the higher education (HE) community. The areas of greatest dissatisfaction and concern to senior leadership were in the categories of administrative software, both those that are not specific to HE (financials / purchasing / physical assets / space management), and those that are specific to HE (student administration / financial aid / admissions / registrar / grants management). They are also concerned about course management systems, which are seen as core to the academic mission.

A very important question is why the existing competitive software market is not better able to meet the needs of this sector. One theory (which to us seems highly compelling) is that the problem lies in the distance between the software producers and users – developers working in the commercial world do not have a nuanced appreciation of the ways in which software is used in higher education. Indeed, this disjuncture between developers and users is common to many industries. Higher education, however, is different in a variety of ways. First, it is small relative to other large sectors of the economy, which may lead enterprise resources planning (ERP) vendors to produce to a broad marketplace that is quite different from higher education. The relatively small size of higher education may also make it especially vulnerable to monopolization. Whereas one vendor may find HE to be profitable, there may not be enough of a market to stimulate the entry that is so essential to effective competition. (This is especially troubling in light of the high initial investment and switching costs imposed on customers and the relatively low number of competing vendors in the software industry.) Second, higher education really is idiosyncratic and has many business practices that are unique and essential to the sector while being deeply puzzling to the corporate world.

In our discussion of the potential for OSS to address these problems, we drew an important distinction between the open characteristics of open source software (literally, the openness and accessibility of the source code) and the manner in which it is produced. We refer to the community-based volunteer model associated with the likes of Linux and Apache as “community development,” and note that it is also possible to produce software with open source code through a centrally managed (and often funded) process, which we refer to as “directed development.” When we refer to “open source software”, we mean just that the software has open source code, without reference as to how it was produced.

We concluded that there is plenty of reason to believe that community developed OSS can be very effective, and moreover that it is possible for universities and colleges collectively to produce open source software that meets their needs as well as or better than commercial products. However, we observed that community development seems to work best when developers are also users of the software (as is the case for software deep in the hierarchy, or applications such as web browsers and email that developers use frequently). It is less clear that community development is appropriate for complex administrative functions such as payroll and HR, which were identified as the areas of greatest dissatisfaction and concern to HE leaders. It does not seem likely that such applications will spontaneously emerge from the developer community. Moreover, people on the demand side are more cautious when it comes to adopting community developed OSS built for mission critical applications. Managers are skeptical that a decentralized process can produce reliable software that meets their requirements. Some are also concerned about the lack of a number to call if a system crashes, though certain OSS products are very well supported through a robust market of commercial vendors.

Based on what we learned about the shortcomings of the current market and the potential of OSS, we devised some scenarios of what a desirable world might look like five years from now. The first picture is one that almost anyone in higher education would view as overwhelmingly successful.

Here, the HE sector is on a clear path to a set of effective administrative solutions that have reasonable initial cost, that are well-tuned to the needs of higher education, that are adaptable to changing circumstances and requirements, and that are well supported, either in-house or through external vendors. Some of the products are open source, some may be commercial, and they operate in a sufficiently diverse market ecosystem so that there is little threat of monopolization. There is a high degree of interoperability within and among suites of software products, and upgrades can be managed within reasonable cost. The number of contribution agreement forms and OSS licenses is minimized, and institutions face minimal financial, operational and legal risk with using or contributing to OSS products. The second picture is a more tempered picture of success. Here, the vast majority of implementations continue to be proprietary, but in at least a significant number of the important domains, there is a potentially viable open source product that is good enough to exert significant market discipline on potential monopolists. This is the minimal level of improvement to the status quo that could be viewed as a successful outcome.

What can be done to move the community in this direction? What we found was that the case for collaborative, directed development open source projects designed to serve the operations of institutions of higher education in the United States seems to be a powerful one. First, based on the sample of institutions we consulted, it is not uncommon for colleges and especially large research universities to consider building their own solutions to enterprise problems. They are driven to do this both because of the concerns we described earlier with commercial options and because many higher education institutions have significant professional resources that are relatively well attuned to the particular (even peculiar) needs of higher education. When one or more institutions are considering building their own solution to an enterprise problem, spreading the costs over even a fairly small number of partners can greatly reduce the cost to each (although costs of collaboration may rise more than linearly). Advantages should generally be spread much more widely, and it makes sense for many institutions to look over the shoulders of the few that will be at the center of any one development. These efforts can also benefit from a signature feature of the open source and open content movements – many eyes, and many keyboards, can in theory improve the quality of code through a process of continual review and testing.

We believe that software projects (both in higher education and more generally) work best when there is clear mutual understanding between the users and the developers regarding how the software is to be used and what is important for it to accomplish. The success of many community-based open source projects derives from just such a confluence. Our examination of a number of OSS projects, both community developed and those developed via directed collaboration, is consistent with the importance of detailed, substantive engagement between the supply and demand sides in development. Lack of such engagement is common in higher education and in many other industries, and is especially visible in higher education when commercial products are employed. The biggest potential payoff to OOSS would be in correcting the coordination failures between users and developers in a more systematic way.

There are, however, substantial organizational and legal barriers to effective cooperation of the kind that we think is most valuable, and there does not yet exist a proven organizational model for projects as they move towards sustainability. The propensity of colleges and universities to build their own software also illustrates the tendency for each institution to act independently. Sakai, DSpace, and (in prospect) Kuali all seem to be reinventing similar wheels. We have also observed the experiences of a number of initiatives and organizations that were created to pursue various forms of collaboration in the IT space among universities and colleges, but that have not succeeded in fulfilling their missions to the extent one might have hoped. In many of these cases, a key missing ingredient appears to be vigorous and active leadership from the highest levels of administration, including presidents. An important lesson here, we think, is that any structure, be it project by project or a version of OOSS,

must be commissioned and governed by an entity that has substantial authority, an effective governing structure, and a clearly agreed-upon sense of mission.

There was a general belief that some kind of coordinating body would be useful in facilitating the development of effective OSS solutions; however, there was very little consensus as to what such a body would look like. As one would expect, senior leaders with responsibility for building or buying systems (the demand side) were focused on addressing broader issues facing their institutions and on mitigating the risks associated with adopting OSS, while project principals and developers (the supply side) focused on what OOSS could do to help their individual projects succeed. So those on the demand side looked for help in coordinating their investments in software development with those of other universities and in gaining access to better information about the reliability and performance of OSS. Those on the supply side tended to look for technical, marketing and business planning services, assistance with organization and related tax and governance issues, and easier access to resources. We sought to find overlap in the expressed needs of these important constituents.

One critical area where both the demand and supply side agreed that help was needed was with legal issues. There are three sets of issues. (1) A major challenge for open source projects, given the number of contributors who may be involved and the common usage of pre-existing OSS, is ensuring that all of the software used in the project is appropriately licensed. In the case of contributors, this means contribution agreements to document that permission has been given for the use of the contributed code. Many contribution agreements must be signed not only by individual contributors, but also by their institutions and employers. Likewise, the number of different forms of open source licenses has increased, and licenses attached to embedded open source software may not be "compatible" with each other. The need to negotiate and enter into contribution agreements with multiple individuals and institutions and the increasing number of forms of open source license generates a real need for legal assistance. (2) Another area of concern is with potential institutional or personal liability. There are two aspects to this problem. The first aspect is the actual risk of liability, as there is a possibility that individuals and institutions associated with a project could be held jointly responsible if portions of the code submitted by one contributor are infringing or otherwise problematic. Limited liability is one potential solution to this problem. The second aspect of the problem is one of perception, not of substance, but it is no less real. Proprietary software firms have every incentive to try to exploit concerns over potential liability for infringement and business interruption costs to spread fear, uncertainty, and doubt – or "FUD" – among institutions that are considering open source alternatives to their products. (3) Finally, the issue of governance – establishing where ownership of the code should reside and who should be tasked with overseeing its development – was identified as a high priority. With the appropriate structure, an OOSS could play an important role in promoting use of more uniform licensing practices and managing liability risks, as well as providing assistance and best practices to help solve the governance question.

Convinced as we are of the value of collaborative directed open source projects for higher education, we are not settled on an ideal mechanism to support their creation and deployment. In increasing order of organizational complexity and scope, we see a continuum of activities that could be undertaken in service of the mission. At a minimum, the virtues of OSS should be actively promulgated in many existing forums to alleviate some of the concerns of senior leadership (what we call "Jawboning"). Some awareness-raising on the importance of careful attention to licensing and intellectual property issues, and the need to establish standard forms of contribution agreement and open source licenses would also be valuable. This set of activities might be made more effective by formally appointing someone to coordinate them ("Jawboning Plus").

The next level of activity, which we see as the minimal version of OOSS, is what we have called "Market-Maker Plus." Under this scenario, with respect to projects in prospect or in their early stages,

OOSS would provide expertise, informational and logistical support to pull together a project team and a set of sponsors, the latter generally being higher education institutions. With respect to mature open source projects, it would provide a set of "Consumer Reports" –like functions regarding aspects of the software itself (including ease of modification and use) and the quality of commercial or noncommercial support of the software. It could also audit the licensing practices of open source projects and identify areas where further work is needed. (It should be noted that there are potential conflicts between the facilitating and evaluating functions.) A major weakness of this version of OOSS is that it has little leverage to effect collaboration as well as no mechanism for branding or certifying the value of particular projects.

In a stronger version of "Market-Maker Plus" OOSS would act as the steward for the intellectual property deriving from directed development projects, and either distribute or license to a subsidiary the distribution of the software ("Conservancy"). Projects would still operate fairly independently but would compete to get into OOSS, which would provide support to the projects and offer a valuable imprimatur or brand. By establishing itself as such an "umbrella" organization, OOSS would help individual and institutional contributors enjoy the advantages of limited liability without incurring the redundant costs involved in establishing numerous separate organizations. If OOSS were to serve as a holding company for the intellectual property associated with open source projects, OOSS could also help protect the aggregation of software and licenses that comprises each open source project, and would be in a better position to provide legal advice to the projects (though it is important to note that conflicts of interests and confidentiality might arise).

We also considered briefly a more ambitious version of OOSS, in which OOSS would be a subscription-based software development corporation. There was a fairly strong consensus that any advantages were outweighed by concerns about imposing too much top-down control, stifling innovation, allocation of resources, and incurring high costs.

In evaluating options, a key question is the level of engaged commitment obtainable from university leadership, both from the larger universities that are likely to provide personnel to write code, and from smaller ones, which have demands of their own. If the best outcomes are to be realized, there would have to be persuasive commitment from the senior leadership of a fairly large set of institutions both to support OOSS and to attend seriously to advice that it would give. Provosts and CFOs, in collaboration with their CIOs, would have to engage with OOSS positively and as a matter of routine when considering major projects and at times would have to make decisions that factored in the interests of the broad community in addition to those of their home institutions. Presidents would have to exert strong leadership in this regard to avoid the not-invented-here syndrome. We expect that such engagement would have the added benefit of improving the quality of collaboration between academic and technical leadership on campus. Without such a powerful commitment, the most valuable models that we have identified cannot succeed. Our own view is that with such a commitment, either the Market Maker Plus or Conservancy would be worth trying, with the chances of success, and also the risks and costs, greater for the latter.

Section I : Introduction and Framing

The use of open source software (OSS) has grown rapidly in the last decade, and the growth has been accelerating over the period.[2] OSS is widely used in many domains, most notably in the operation of computers and computer networks. Among the best-known applications are the Apache web server, which runs 65% of active websites,[3] and Linux, which is estimated to operate on 20% of servers.[4] There are also many open source solutions, in use for web browsers, e-mail, instant messaging, file sharing and other applications.

Use of OSS has also taken off in higher education. Linux and Apache are widely used, as are many other programs. A number of users and producers of information technology in higher education have raised the question of whether OSS could serve the sector more powerfully. Specifically, it has been suggested that institutions of higher education could operate both more effectively and more efficiently if they were able to cooperate in the development and use of OSS designed to serve the administrative and operational (as distinct from research) computing needs in higher education. This suggestion provides the motivation for this report, which addresses the specific question of whether the higher education sector in the U.S. would benefit from the establishment of an organization whose purpose was to promote and support the development of OSS in its operations.[5]

We note that as a general matter, the quality of administrative and related software is not an important domain of competition for colleges and universities. If administration can be accomplished more effectively and at lower resource cost, more resources will be available for the core missions of teaching, research and service, and, in principle, all or almost all institutions can be made more effective and valuable as a result.

In October 2005, The William and Flora Hewlett Foundation and The Andrew W. Mellon Foundation convened a group of leaders in higher education (See Appendix D for a list of participants) to discuss the possibility that is the subject of this report – whether development of an organization to coordinate and support OSS in higher education would be of value. The group did not reach an answer, but there was general agreement that the issue was worth exploring further. A group of institutions, including Carnegie Mellon, Foothill-De Anza Community College, Marist College, Michigan, Indiana, Stanford, UNC, and the Hewlett and Mellon Foundations subsequently contributed funding for this study. The issues – for, against, and in-between – that emerged in the October meeting framed our study initially and proved to be robust as we conducted our research. Thus, before getting into the methods, findings, and policy options that are the heart of this document, it is useful to review the principal arguments and questions that emerged in that initial discussion.

---

[2] For a compilation of data on the growth of open source, see David A. Wheeler, "Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!" available at http://www.dwheeler.com/oss_fs_why.html (accessed May 7, 2006).

[3] As of the April 2006. Apache vastly outranks Microsoft, whose various server products together have only a 26% share. See the Netcraft April 2006 Web Server Survey, online at http://news.netcraft.com/archives/2006/04/06/april_2006_web_server_survey.html (accessed May 7, 2006).

[4] In 2006, according to estimates by IDC and Citigroup Investment Research. See Brent Thill, John Reilly Walsh, "Red Hat: Early Innings in Linux and Open Source = Sustainable High Growth," Citigroup Investment Research, April 24, 2006. Gartner estimates that Linux has 21% market share in 2006, and forecasts that this will rise to 26% in 2010. Jeffrey Hewitt, "Linux Making Strong Inroads in Server Market," Gartner Research Paper ID Number: G00126977, April 4, 2005.

[5] For an early proposal for such an organization, see Ira Fuchs, "Needed: an 'Educore' to Aid Collaboration," *Chronicle of Higher Education*, September 24, 2004. Also available online at http://chronicle.com/weekly/v51/i05/05b01901.htm (accessed May 7, 2006).

1. Opportunities and Concerns re OSS in Higher Education

It is fair to say that a substantial majority of college and university presidents and provosts are not very satisfied with the market for software that they use to administer their institutions. Three sets of related concerns emerge repeatedly: (1) Commercial products are often not well tailored to higher education. For example, financial software often does not recognize that in a university there will be dozens or hundreds of entities authorized to make purchases; HR software does not easily accommodate multiple appointments paid for from multiple resources over different periods of time; HR and student administration must deal with the complexity that many students are also employees, and vice versa. (2) College and university leaders are concerned that consolidation could result in commercial vendors having excessive leverage to raise prices for the software used in higher education. The recent acquisitions of PeopleSoft by Oracle, and WebCT by Blackboard give credence to this set of concerns. (3) Commercial software tends to require frequent and costly upgrades. Where users have customized the software to meet their own business needs, these customizations need to be repeated with each upgrade. Even when universities and colleges are able to run the "vanilla" form, the vendors require (by withdrawing support of old versions) upgrades that are more frequent than would have been chosen and that have functionality that differs from what the customer would most value.

The result is that many in positions of leadership in higher education find that they are spending what seems to them a lot of money, both initially and over time, relative to what they get. Strikingly, and diagnostic of the extent of the problem as perceived by university leadership, a significant number of the institutions profiled in our sample have chosen to "build" rather than to "buy" significant portions of the enterprise software they run.[6] In these cases, the potential payoff to having a number of institutions share the cost, using OSS as a mechanism, is plausibly very high.

A key point is that there is likely to be a payoff to close coordination between users and developers, as would be facilitated by directed development projects among universities and colleges. Indeed, we argue below that many of the most successful OSS projects have been developed by the same people who would be the users of the software, making the coordination automatic. Such coordination would improve the quality of information going into upgrades as well as original design, and hence the quality and value of the upgrades themselves. Additionally, it is in the nature of OSS that customization of the software is easier than with proprietary products, as the code is open and users are free to change it.

The preceding considerations suggest that OSS produced through both directed development and community development can be valuable in higher education, and we will examine (generally favorably) both of these propositions in more detail later in this report. Even so, it does not necessarily follow that OSS in higher education would be well served by a formal organization dedicated to promoting its development and support, nor is it obvious how such an organization would best be configured. One argument in favor of such an organization derives from the fact that a number of OSS projects (Sakai, DSpace, Kuali), of just the kind that are of most interest to university leadership because they meet the concerns with commercial software raised above, are establishing independent "foundations" to support their activity. Everyone agrees that interoperability of these and other products will be of value; without an organizational structure pressing interoperability, it is likely to be subordinate to the immediate concerns of each project. Similarly, having a plethora of

[6] In our sample, the University of Texas, the University of Phoenix, Carnegie Mellon University, DePauw University, Michigan, Indiana, and Marist College had all built significant modules of administrative software (and in some case much more) in house. Sakai, Kuali, and the prospective collaboration around student information systems are all founded on software built within academic institutions.

"foundations," each with different governance structures, licensing protocols, and the like, will cause failures of interoperability in legal and administrative realms, as well as a good deal of wasteful duplication of organizational effort and resources devoted to governance.

Potential Problems. Three sets of problems regarding OSS in higher education were raised at the initial meeting, and are also discussed at some length in this report. A number of leaders, especially from smaller institutions that do not have extensive IT staff, are concerned about support for noncommercial products used in their production systems. When the payroll program fails, or course registration is put on hold, timely and effective support is an urgent requirement. Commercial vendors generally promise (and sometimes deliver) this kind of support.

Meanwhile, many managers and leaders see community developed OSS as insufficiently businesslike (informal, voluntary, even ideologically anarchic) to be used for activities that are essential business operations of the university. In many contexts, OSS is used in a very businesslike way: Apache and Linux are part of the operating fabric of thousands of commercial (and noncommercial) enterprises. An increasing number of firms are making it all or part of their business to provision and support OSS. IBM, among many others, contributes both employees and financial resources to the development of open source projects. That some open source products can be relied upon for essential operations is well established, but in order for specific products to be deployed successfully in higher education, managers will have to be sure that those products are reliable and well-supported. One of the possible roles of an organization to support OSS in higher education would be to assure, either through third parties, commissioning, or provision, adequate operational support for OSS products.

A second set of concerns is more philosophical. The information technology industries in the United States are robust, active and innovative. Without a compelling characterization of market failure – of why the private market cannot deliver well to the higher education sector – it is hard to see why a new, nonprofit organization would be necessary to deliver services that look at first blush to be potentially profitable. Is higher education so small or so different that the marketplace cannot serve its needs well?

Finally, the establishment of an organization to promote and support the use of open source software in higher education raises questions of mission, governance, and organizational form. Who would own it? What would it be called upon to do (and not to do)? Who would pay for it? How would it be governed? What would be its relationship to existing organizations, and to existing and future projects? How could it coordinate projects effectively without stifling innovation? In summary, what do we really have in mind when we consider such an organization?

2. The Organization of the Report

This report is largely an evaluation of the potential costs and benefits of establishing OOSS as a cooperative venture of institutions of higher education and of other institutions whose mission involves the support and improvement of higher education. The primary audience for the report is the leadership of U.S. higher education, as this group will need to be the locus of activity and support for such an organization (moreover, the study itself was commissioned by leaders from a small but diverse group of institutions). At the same time, we recognize that the outcome of this report could have a more direct impact on the day to day lives of principals and developers involved in open source projects, and that the support of this group will be essential to the success of any new endeavor that might materialize as a result. Thus, they also constitute a critical audience.

In order for OOSS to be of value, it must be the case that OSS itself has a realistic promise of improving the efficiency and effectiveness of college and university operations relative to products in

the commercial marketplace. We will argue below that such promise will generally derive from directed development projects involving several institutions, responding to clearly articulated goals of the academic leadership. UPortal, Sakai, DSpace and Kuali are each in their own way useful models, although all are quite different from one another in material ways. Crucially, by its very nature open source facilitates collaboration, both in initial development and over the life cycle of projects. But as we have noted above, establishing the value of directed open source projects does not in itself make the case for OOSS. We will explore some of the organizational, legal, and mission-related hurdles that must be overcome in order for OOSS to succeed.

The remainder of the report is organized as follows: Section II provides a discussion of how we went about our research, who we talked with, and what sorts of information we sought. Section III is the heart of this report, containing a detailed summary of what our various informants had to say and what we learned about possible products and possible roles for OOSS. Section IV provides two descriptions of what the world would look like in five years if open source were successfully employed in higher education. One describes a "most desirable" outcome, while the other is somewhat less so, yet would still constitute success. The implication is that there is a good deal of upside potential.

In Section V we examine in some detail different models of open source development, focusing on what is required to develop large-scale collaborative projects of the type that compete with commercial enterprise-wide administrative software. This is the domain where the case for collaboration is strongest *a priori.*

Section VI provides policy options, ranging from something close to laissez-faire to establishment of an OOSS organization that would own and distribute software, as well as assist in the commissioning of new projects.

Section II: Methodology

In designing the OOSS study, a primary objective was to talk to a cross section of people from different constituencies and types of organizations and with a variety of perspectives on the use of open source software in higher education. In total we interviewed over sixty people. One important group consulted were senior leaders – presidents, provosts, CIOs, CFOs – as their support would be critical to any new organization's ability to have maximum impact and to gain financial backing. Understanding their perspectives and goals is especially important because, although this group has so much at stake in this discussion, it often seems less well represented in the discussions about open source than developers and principals of open source projects.

The needs and perspectives of academic leaders and managers have a large degree of variation depending on their employer's size, mission, and access to resources. We therefore targeted institutions across a number of segments, including: large research universities, state schools, wealthy liberal arts colleges, less well resourced institutions, community colleges, and for-profit institutions. We consulted with administrators at two major state systems, Texas and California, and with other institutions that have played leading roles in open source projects such as Indiana, Michigan, MIT, and Stanford. A complete list of people interviewed for this project is provided in Appendix A.

The principals and developers involved in open source projects are also key stakeholders. Understanding their motivations is vital, as the long term sustainability of many (if not most) open source projects in higher education depends on their personal commitment. People consulted from

this group had a wide range of views about the potential for open source in higher education and what should be done to support it. As one would expect, these views tended to reflect the stage of development and needs of their particular project(s). We also spoke with people who were involved in open source projects outside the university environment, principally Apache. They offered a different perspective on what ingredients are necessary for open source projects to succeed, and how projects in higher education look different from other open source efforts.

In the course of this study we also spoke with staff at a number of commercial firms that provide support services for open source software (rSmart and Unicon), those that have major corporate investments in open source (IBM and Sun), and those that may compete with open source projects in higher education, such as Blackboard/WebCT and Oracle. Finally, we spoke to people at many of the myriad other organizations involved in some aspect of IT in higher educations, such as the Common Solutions Group, IMS Global Learning Consortium, Educause, the Alliance for Higher Education Competitiveness, OSS-Watch (UK), Ask-OSS (Australia), and the Open Source Development Lab's Higher Education Forum.

In addition to conducting interviews, we gathered and consulted information from a wide range of secondary sources. Surveys on usage of open source and IT spending in HE corroborated some of our more anecdotal findings. Articles and papers chronicling the progress of open source, as well as books by authorities on the subject such as Steven Weber and Eric Raymond[7], informed our understanding of the movement's evolution and success. Discussion boards of prominent OSS projects provided a picture of some of the issues of concern to the developer community.

We approached our research with our minds open to learning what people in these various roles see as their objectives with respect to software, what obstacles they believe stand in their way, and what solutions they believe would help to overcome these obstacles. At the same time, our motivation was to find evidence for or against a set of propositions that relate to the principal arguments and questions set forth in the October meeting. Thus, we sought evidence regarding the following assertions:

- There is at least a sizable community of users who are deeply dissatisfied with the cost and performance of currently available options, and this is a subject of concern at the university leadership level.
- Open source software is a viable solution to this problem, and university leaders are receptive to it.
- There is significant demand for certain software products that is not being met, suggesting that there is a market failure and hence the possibility that concerted action can effectively address that failure.

And finally, we looked for visions of what an organization could do to promote solutions to problems in higher education via the use of open source. During the course of our study, we discovered that legal issues around licensing and governance pose serious challenges for the open source movement, and thus we have dedicated a lengthy sub-section to discussing these issues.

[7] See Weber, Steven. *The Success of Open Source*. Cambridge: Harvard University Press, 2004. See also Raymond, Eric. *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly, 1999.

Section III: Findings: What People Told us and What we Learned

In this section we will review what people told us in relation to the central propositions described in Section II.

1. Dissatisfaction with Current Options

We found a considerable amount of evidence attesting that many university leaders are dissatisfied with the cost and performance of software, and that this is a matter of significant concern to them. This was particularly true in institutions that have recently gone through the process of implementing a major system. The areas of dissatisfaction can be grouped under three headings: cost, performance, and control.

There is a general feeling among university leaders that they spend a lot on information technology, particularly administrative systems such as finance and HR. These are the systems that have cost millions, and in some cases hundreds of millions, of dollars to install. It is worth noting that the bulk of that cost is in implementation, not licensing costs per se, due to the need for customization and training and the difficulty of getting various systems to interoperate.[8] Many of the costs have to be repeated when the software is upgraded. One concern conveyed by several presidents and provosts is that it is very hard to know how much they should be spending. They expressed frustration with the process of making decisions about what systems to install, as they do not have a good understanding of the cost drivers or even know what questions to ask in looking for ways to reduce costs.

What is especially aggravating is that, after spending all this money, many people are not happy with the results. It was noted that software provided by vendors like PeopleSoft and SAP is adapted from other industries and often does not work well for educational institutions, which tend to have idiosyncratic processes and ways of managing resources. Because the software is proprietary and the source code is inaccessible, it is expensive and difficult to customize these software packages to an individual institution's requirements.

A third concern expressed by a number of university leaders is that a small number of commercial vendors hold a great deal of market power in areas of core operating or strategic importance. This power comes into play in two ways: first, the market for administrative and learning management systems is dominated by a small number of companies, so choice is limited to one or two providers for a given type of software. The acquisitions of PeopleSoft by Oracle and of WebCT by Blackboard were frequently cited as sources of unease. These concerns were heightened by reports that each of these companies has plans to offer more integrated systems that might extend the community's dependence on them. Oracle's Fusion strategy aims to connect all levels of enterprise technology — database, middleware, and applications. Blackboard plans to expand into administrative software, enabling its course management software to integrate with related systems that manage student information. One could imagine a rather bleak scenario in which universities and colleges feel compelled to purchase monolithic systems with even less flexibility than they have now to optimize on individual modules.

The second dimension to the supplier's high level of power derives from the enormous switching costs of moving from one provider's platform to another. Expensive customizations must be repeated and interfaces with other systems rebuilt. Furthermore, staff are highly resistant to learning to use a new

[8] For an informal study of this point, see Jim Barry, Wayne D. Powel, "Post-Implementation Evaluation of an ERP/SIS," presented at the Educause conference, October 19-22, 2004, available online at http://www.educause.edu/LibraryDetailPage/666?ID=EDU0457 (accessed May 7, 2006).

system (this was a comment made repeatedly with respect to faculty and course management systems). Most CIOs said they expect administrative systems to last 10-15 years, which seems like an eternity in the software world but provides some indication of how reluctant users are to switch products. One CIO estimated that in a given year, his institution turns over 5-7% of its software. This mindset suggests that any change in the industry would take some time, as institutions tend not to make changes to their existing stock of software until a product is long in the tooth or otherwise performing inadequately.

It is not obvious, at first blush, why a vibrant and competitive software market is not better able to meet the needs of this sector. One theory (which to us seems highly compelling) is that the problem lies in the distance between the software producers and users – developers working in the commercial world do not have a nuanced appreciation of the ways in which software is used in higher education. Indeed, this disjuncture between developers and users is common to many industries. Higher education, however, is different in a variety of ways. First, it is small relative to other large sectors of the economy, which may lead enterprise resources planning (ERP) vendors to produce to a broad marketplace that is quite different from higher education. The relatively small size of higher education may also make it especially vulnerable to monopolization. Whereas one vendor may find HE to be profitable, there may not be enough of a market to stimulate the entry that is so essential to effective competition. (This is especially troubling in light of the increasing levels of concentration and the relatively high initial costs and switching costs in the software industry.) Second, higher education has a fairly robust tradition of building its own software, and many institutions have significant professional resources that are relatively well attuned to the particular (even peculiar) needs of higher education. With the exception of the IT industries themselves, higher education probably has the best informed in-house technical expertise of any industry in the country. And, third, HE really is idiosyncratic and quirky and has many business practices that are unique and essential to the sector while being deeply puzzling to the corporate world that is the principal supplier to the sector. (E.g., multiple appointments from multiple funding sources; simultaneous use of academic years and calendar years for pay calculations.)

For all of these reasons, HE is both relatively unattractive to potential entrants (indeed, the tendency has been for large vendors to merge with each other, which has just the opposite effect of entry) and has the potential to overcome substantive failures of coordination between producers and consumers by using its own staff. There is thus a conjunction of apparent market failure and plausible market opportunity.

Of course, it is possible that the problem lies in the higher education industry's practices. There is surely some truth to the notion that academic institutions are prone to indulging in their uniqueness, and that a more disciplined sector would focus on modifying its business practices rather than doing extensive customizations or even building its own ERP systems from scratch. However, we spoke to several leaders who had attempted to adapt to commercial ERP software in vanilla form and who described the process as highly disruptive and not ultimately very successful.[9] An interesting data point is that the University of Phoenix, which one would expect to operate with hard-nosed fiscal discipline, has chosen to build most of its own administrative systems. To be fair, we also met with leadership at a couple institutions that are quite happy with their commercial software and, reportedly though a combination of luck and management, were able to implement these systems with minimal pain. This point supports one of our general findings, which is that the needs in the community are far from homogenous, and any proposed solution will meet the needs of some constituents and institutions better than others.

[9] Here is a simple and telling example. Commercial entities cannot imagine the necessity for managing multiple appointments on multiple funding lines, something that is essential for a research university.

2. Products

Colleges and universities are complicated places, requiring the integration of many processes for both operations and policy development. In the course of our study, college and university leaders identified a number of areas in which better enterprise-level software could improve the quality of performance and reduce costs in some combination. The range, not surprisingly, is from the utterly essential (e.g., meet the payroll and register the students) to the optional but potentially valuable (e.g., create a template for faculty to report on their professional activity). The products discussed are described in detail in Appendix B, along with reflections on their level of importance and the perceived risks and shortcoming of the current options. The most salient point to raise here is that the areas of greatest dissatisfaction and concern to senior leadership were in the categories of administrative software that is not specific to HE (financials / purchasing / physical assets / space management), administrative systems that are specific to HE (student administration / financial aid / admissions / registrar / grants management) and course management, which is seen as core to the academic mission. An additional area of increasing concern is security, both of the network itself and of sensitive data about students and employees.

3. Is Open Source a Viable Solution?

We should reiterate here what we mean by "open source software." We use this term to refer to software for which the source code is made available for others to use, view, modify, and redistribute. The term "community development" is used to describe the distributed process that has been used for projects like Linux and Apache. Open source software can also be produced through a centralized, controlled approach, which we refer to as "directed development." (Once developed and in production, of course, its openness makes it amenable to decentralized, local changes.)

The success of community developed open source is quite well established. Apache has nearly two thirds of the market for web servers, approximately 10,000 contributors, and a sophisticated organization in place to oversee the software's development. Some form of Linux is used as the operating system in 20% of computers, and also has thousands of contributors and a highly evolved development process, led by its "benign dictator" Linus Torvalds. Moodle, a project from within the higher education community, has been installed in over 10,000 sites (including both secondary schools and higher education) and has nearly 200 contributors worldwide. None of these examples required significant up-front funding and all rely primarily on volunteers for code contributions.

These types of efforts have yielded enormous benefits for software users, and higher education clearly shares in them. A recent study done by the Alliance for Higher Education Competitiveness (A-HEC) found that 57% of institutions in the U.S. use some form of open source infrastructure software (operating systems, web servers, databases, etc.), and 34% have implemented open source application software (course management systems, web browsers, spam filters, and the like).[10] There is some variation in the perceived benefits to academic institutions of using open source. For schools with limited resources, the ability to acquire software without paying license fees is an important advantage. However, because these institutions also tend to have small IT departments, open source is only a serious option if it is very easy to install and maintain (Moodle appears to be meeting this test). A-HEC found that institutions with annual operating budgets of greater than $100 million were far

[10] The latter number includes proprietary products that are based on open source, such as SCT Luminis (based on uPortal). See Rob Abel, "Best Practices in Open Source in Higher Education Study: The State of Open Source Software," February 3, 2006.

more likely to be using open source infrastructure and applications than their smaller peers.[11] It is very possible that these schools would adopt less installation-friendly open source if support services were available (and at a lower cost than what they would pay for purchasing and installing proprietary software). On the other hand, the financial impact of open source software is notoriously hard to measure due to the difficulty of adequately accounting for all the ancillary costs of support, maintenance, upgrades, etc. CIOs of two large research universities commented that they expected the "total cost of ownership" (by which they probably meant annual operating budgets) for open source to be roughly the same as for commercial software. The main advantages of open source from their perspective are the added flexibility and control, and potential for increased longevity. Measured properly, these improvements in effectiveness and reductions in cost could reduce the total cost of ownership over any period of years.

Another benefit emphasized by several interviewees is that open source can provide an alternative to commercial software by giving buyers more leverage. Institutions that use Blackboard and have no immediate plans to change are still very happy that Sakai exists. One CIO commented that Blackboard is already more responsive to standards than it was before Sakai. They hope that it will put pressure on Blackboard to lower prices. A provost commented that just the existence of open source administrative software (even if it were not quite production level) would make a big difference in their relationship with Oracle.

There is still an open question as to whether open source can be used to address the most pressing problems identified for higher education in the areas of administrative and learning management systems. Those on the demand side we interviewed are more cautious when it comes to adopting open source software developed through the community development approach for mission critical applications that are specific to higher education. Managers are skeptical that complex administrative systems can be developed through a decentralized, voluntary model that meets their requirements and reliability standards. A couple specifically stated a preference for dealing with commercial firms with a strong profit motive. When the alumni relations database or payroll system crashes, they would like to have a phone number to call (though this could, in fact, be possible through commercial support vendors). In the extreme, and a number of CFOs raised this explicitly, they would like to have someone to (threaten to) sue.

4. Services OOSS Could Provide

In each of our meetings we asked what the interviewee would like to see an OOSS do, if one were to be created. While there was near unanimity that some kind of an organization would be valuable, there was little consensus as to what it would look like. As one would expect, senior leaders were focused on addressing broader issues facing their institutions and on mitigating the risks associated with adopting OSS, while those engaged in projects focused on what OOSS could do to help their projects succeed. So leadership looked for help in coordinating their investments in software development with those of other universities and in gaining access to better information about the reliability and performance of OSS. Project principals and developers looked for technical, marketing and business planning services, solutions to governance issues, and easier access to resources. Appendix C provides a full list of possible activities for OOSS from both the user and developer perspective with commentary on the feasibility of each. Many of these desires will be reflected in the policy options put forth in Section VI.

[11] For infrastructure, 72% of schools with budgets of greater $100 million reported that they had implemented open source software, compared to 47% of schools with budgets of less than $100 million. For application software, 47% of the larger schools were using open source, as against 30% of the smaller colleges and universities. The smaller schools were also far more likely to report that they had not considered using open source software.

5. Legal Issues

The most consistent need expressed by both users and producers of open source software was for assistance with legal issues related to licensing and governance, thus we decided it would be worth describing these in more detail.

Licensing.

One major challenge for open source projects is ensuring that all the code in the software is appropriately licensed.[12] This has two aspects: First, ensuring that code contributions are adequately documented, and second, making sure that there has been an appropriate review to ensure that usage of any third party software complies with whatever form of license applies to that software. The first part can be difficult for projects in higher education, where on the one hand the institution is likely to have rights in any software written by IT staff, and on the other hand the institution is not likely to have in place extensive infrastructure for supervising this work. Furthermore, to the extent the institution has licensing expertise, it is concentrated in a "technology transfer" unit that is more typically tasked with maximizing the value of intellectual property created by the university in negotiations with commercial partners. One consequence is that it can take a significant amount of time and legal resources to negotiate contribution agreements with each contributing institution such that the agreements are consistent and fair to all.[13] The second part – ensuring that incorporating third party software is permitted – can be complicated by the sheer number of different licenses that OSS can be subject to, as well as by ambiguities in many of even the most popular forms of open source license regarding what is and is not permitted. While ordinary users of the software may not need to concern themselves with the details of these licenses, vendors that plan to redistribute the software in the course of providing support or other value-adding services, do. This work makes supporting open source software more expensive and less attractive to commercial vendors, and hence to users.

Another cause for concern is fear of liability associated with creating and distributing open source software. In open source software, code contributions can come from many sources, and it is not easy to verify that they do not contain any code that infringes the intellectual property rights of third parties. The risk that infringing code will creep into a project is, of course, not unique to open source software. Proprietary software vendors must also contend with the possibility that infringing code will find its way into their code base, exposing them to copyright infringement lawsuits. There are, however, two important differences in the way this issue confronts commercial software vendors and open source projects. First, commercial vendors indemnify their customers against liability for using software that contains infringing code. Users are thus offered some protection against the risk of damages associated

12 Each open source project must form two sets of licenses: "inbound" licenses with contributors and "outbound" licenses specifying the terms of use. Contributor agreements usually specify that the code being submitted is the contributor's own work, that it is not an unauthorized copy of code from another source, and that the contributor gives permission to the project to integrate his code into its software, distribute it to its users, and allow those users to modify it. Recently contributor agreements have started to include a limited license to any patents that may belong to the contributor to the project that would be infringed by use of the contribution. Open source "outbound" licenses generally give users permission to run the software on their computers, to read and modify the code, and to redistribute modified versions of the software, as well as disclaimers of any liability to users in connection with use of the software. There are now 58 different forms of open source license approved by the Open Source Initiative, most of which fall under the broad categories of "open closed" or "open open" licenses. Open closed licenses, such as the Apache Software Foundation and BSD licenses, allow users to distribute the code as part of proprietary, closed source software products. "Open open" licenses, such as the well-known GPL, require that any further redistribution or reuse of modified versions of the software be under the same open source terms under which the unmodified version was originally provided. These are sometimes referred to as "reciprocal," or pejoratively as "viral," because they spread their open source qualities to any piece of software that adopts part or all of their code.

13 Many extant projects have been relying on counsel at their host institutions or pro bono advice from unaffiliated lawyers, but this model is not sustainable, especially as the number of projects expands.

with using infringing software, although the indemnity would typically not cover possible business interruption and other indirect costs that may actually be far higher. Open source projects do not generally have the wherewithal to provide an indemnity, and typically disclaim any liability to end users. Second, by definition the source code of open source software is exposed, making any infringing code easier to detect by copyright owners and their allies. The more successful an open source project becomes, and the more it threatens commercial interests, the more likely it is to attract scrutiny. The announcement of a partnership between a U.S. firm called Open Source Risk Management and Lloyds of London to underwrite open source insurance for end user institutions, offering the functional equivalent of an indemnity, shows both that the risk of infringement is perceived as a serious issue, and that it is possible to evaluate and get comfortable with, or insure against, that risk.[14]

There is a real need to make sure that institutions whose staff have contributed to open source projects – especially lead institutions – are as protected as possible from infringement claims. At the same time, however, we believe that the risk that commercial firms will exploit the potential for liability and business interruption costs to spread fear, uncertainty, and doubt – "FUD" – among institutions that are considering open source alternatives to their products to be just as serious, if not more serious, threat to the long term success of open source in higher education. If FUD persuades academic institutions to take a conservative approach to the use of open source, these projects may never get off the ground.

Another noteworthy development is the creation of the Software Freedom Law Center (SFLC), a new non-profit law firm founded by Columbia Law School professor Eben Moglen to serve open source projects. SFLC provides its services at no cost to open source projects. Moglen is the former General Counsel of the Free Software Foundation and is closely involved in the development of the next version of the GPL license. His center has received funding from IBM, HP, Fujitsu, Red Hat, Hitachi, and the Open Source Development Lab (OSDL), because they are heavily invested in the future of open source and believe this problem is so important. In addition to legal services, SFLC recently announced that it is creating the Software Freedom Conservancy, which provides financial and administrative services as well as a formal legal structure to house OSS projects.[15] We were encouraged that Professor Moglen expressed interest in providing services to OSS projects in higher education.

With proper guidance, there are measures individual open source projects can take to mitigate the risk of infringement. Project managers and contributors must be educated to implement best practices from the beginning of each project to ensure their code is "clean." Contributor agreements can play an important role in creating awareness among developers of the importance of respect for intellectual property rights and the risks that can be created for projects if all contributions are not properly licensed.

Governance.

Almost all projects want help getting organized and finding a legal and organizational home. Their motivations can vary. Some have been developed and incubated inside a university (or small group of universities) and reach a point at which they feel they have outgrown this arrangement. They wish to concentrate responsibility and legal ownership in a separate entity, while opening the possibility of attracting resources from a larger community. Some see this as a useful way to transition from a

[14] See http://www.osriskmanagement.com/open-source-compliance-insurance.shtml

[15] See http://conservancy.softwarefreedom.org/

centralized, grant-funded project to one relying on a decentralized community of contributors. Several projects at this stage of their lifecycles expressed interest in transferring ownership and management to an OOSS. Others want to be independent, and are more than happy to live outside an OOSS. For the latter, launching new non-profit "foundations" or other type of legal entities seems to have appeal, perhaps in part because doing so can create a perception of stability and critical mass in the user community. However, if the trend to creating an independent "foundation" for each project could be forestalled, there could be considerable saving in both financial and organizational costs across the system.

Another motivation for creating an independent legal entity is to protect the managers of an open source project from liability risk. Incorporation concentrates liability in the legal entity and reduces potential liability for individual managers. Contributors remain responsible for their own actions, but the possibility that they will be found liable for infringing code provided by other contributors is greatly reduced. This has been a factor in the creation of legal entities outside higher education, such as the Apache Foundation (many of whose initial contributors made a lot of money in their day jobs). While none of the individual developers we consulted for this project mentioned concerns about their personal liability risk, this may be an important consideration from the perspective of universities and colleges that sponsor OSS projects.

Section IV. Measures of Success (or its Absence)

Having listened to presidents, provosts, CFOs, CIOs, project managers, developers, and users of software, we have developed long lists of things that people like and do not like about the current state of affairs and about prospects for the future. In this section we distill those lists into a broad agenda for improvement and two possible pictures of the world five to ten years hence.

The first picture is one that almost anyone in higher education would view as overwhelmingly successful. Here, the HE sector is on a clear path to a set of effective administrative solutions that have reasonable initial cost, that are adaptable to changing circumstances and requirements, and that are well supported, either in-house or through external vendors. Some of the products in use or in prospect are open source, some may be commercial, and all face sufficient market pressure so that there is little threat of monopolization. Should OOSS have contributed to this set of outcomes, it will have been one of the most successful cooperative institutions in the history of higher education.

The second picture is one of qualified success. Here, the vast majority of implementations continue to be proprietary, but in at least a significant number of the important domains, there is a potentially viable open source product that is good enough so that it can exert significant market discipline on the prices that commercial vendors attempt to charge the HE community. This is the minimal level of improvement to the status quo that could be viewed as a successful outcome. It is possible that a laissez faire environment (i.e. in the absence of an OOSS) could get us this far.

1. An Expansive Vision of Success

In the best of all possible worlds, all of the products discussed briefly in Section III (and profiled in depth in Appendix B) and more would be either operational or in prospect over the next several years. Moreover, the following characteristics would apply generally:

- Applications would last longer and be continuously adaptable. Many commentators lamented the fact that the life cycle of the products that they use is short, leading to both financial and learning expense when changes are made.

- The marketplace would offer a diverse set of responsive products.
- Ideally, there would be a very high degree of interoperability within suites of software products, while interoperability across suites would be fairly good and getting better. Failing that, it would be relatively easy for either local staff or commercial specialists to make the necessary adjustments and the trend would be towards greater ease and interoperability.
- Implementation and upgrades would be less costly and would not require as extensive and expensive a set of consulting resources as is often the case today.
- Users, from those in the trenches through CFOs, provosts, and presidents, would have better control over the features embodied in the projects.
- Academic and technical leadership would be closely coordinated with respect to choices, costs and opportunities in administrative software.
- For OSS, there would be a standard form of contribution agreement that will satisfy the General Counsels of a preponderance of academic institutions, and a more limited number of different forms of OSS licenses.
- Colleges and universities would face minimal financial, operational and legal risk associated with using OSS products developed for higher education.

2. A More Pedestrian Vision

The minimal level of success would be an attenuation of the risk that the private sector could develop and exploit monopoly power. In this case, OOSS would have contributed to sufficiently robust products in administrative and course management areas – where there is a substantial concern with monopoly and in which at least some institutions are comfortable building their own solutions – to provide some discipline in the marketplace. This may mean that OSS produced by HE does not quite reach production quality (though it is close enough to get there in a year or so, given the necessary investment), and it is only implemented at the relatively small number of institutions with substantial IT capacity. However, the existence of these applications still benefits the broader community by discouraging monopolization. The potential upside for HE could still be large. The sector spent $2.63 billion on administrative technology in 2005, of which 28% was spent on software and 30% on outside services.[16] Even a modest salutary effect on the marketplace would pay handsomely.

It is of course possible that improvements will take place in any case, with or without an organization specifically commissioned to enhance OSS in higher education. That said, to the extent that directed development of cooperative products can foster competition and control prices, efforts to foster such development could serve as valuable insurance for higher education as a whole.

[16] Gretchen W. Rigol, "The College Technology Review, 2004-05 Academic Year," Market Data Retrieval, 2005. Available for purchase at http://www.schooldata.com/mdrreports.asp.

Section V. Findings: What Will Get us There? Why (and When) does Directed Open Source Make Sense? – An Analysis of Supply and Demand

The traditional OSS community development model has many great strengths: costs are broadly distributed, tension among competing priorities is, for the most part, productively channeled, and the ethos of openness and meritocracy is usually well embodied in the formal or informal organizational structures and processes adopted by a given project. However, there are reasons to believe that we cannot rely on this model to organically/spontaneously address some of the most pressing needs in higher education.

One often-voiced concern with the traditional open source development model is that it is most suitable for software that is deep in the hierarchy ("stack") and is not specific to higher education or the operational requirements of any other particular industry. Its greatest successes, including Linux operating systems and Apache web servers, are exemplary of this argument. In apparent contradiction to this generalization, there are a number of examples of successful OSS applications developed through decentralized communities, such as Firefox and Moodle. We argue here that these are exceptions that prove the rule. A common characteristic of both sets of projects is that were all started by individuals who were scratching a proverbial itch – they had specific software needs and did not like the available options. A key feature of all these success stories is that developers are writing software in domains that they care about and know about – e-mail, web servers, operating systems, web browsers, etc. These are either low in the stack (which is the substantive expertise of developers) or are applications that they use and care about and understand.

Crucially, the supply and demand sides are present in OSS projects from the beginning via the developers themselves. This is a feature of the Moodle example, in which the founder, Martin Dougiamas, is both a programmer and an educator, and thus was able to execute his particular vision of what course management software should do. Spontaneous germination has not produced high quality applications in areas such as payroll, grants management, and space management, nor do we expect that it will. Similarly, it is unlikely that developers will be moved to solve the relevant technical problems of university management unless specifically commissioned to do so.

We are also skeptical about the practicality of producing industrial strength, enterprise level software for complex applications through a non-directed process within a fairly small community. Internet browsing is essentially a fairly simple function. Administrative systems, by contrast, are complicated and idiosyncratic, and require the inputs of a variety of constituencies. These products must also be able to support large numbers of users and process huge volumes of data (and be compatible with other systems). Linux and Apache may be able to meet these challenges, but they have the advantages of being lower in the stack, such that programmers themselves are also the primary "users" of the software, and of being general enough to attract thousands of volunteer contributors.

Fortunately, the progress of projects such as uPortal and DSpace suggests that open source software does not have to start with inspired individuals as the principal impetus. It can be seeded by institutional or grant funds and managed through a controlled, centralized process, at least during the critical requirements gathering and build stages. With this type of approach, the open source aspect of directed development software projects can be viewed as an enabling technology – the how rather than the what (though it may be very important for sustainability down the road, as we will explore further below). It is a good technology for collaboration, and it has good symbolic properties for the academy. Once implemented, OSS may be more adaptable, more sustainable, and more amenable to competition in the support domain. The most savvy IT folks we talked to, and the leaders closest to IT, stressed this point. Two speculated that a good chunk of the payoff may be later in the life cycle. This seems plausible to us, but we do not see any way to test that hypothesis at this point. Finally,

open source may help with interoperability. At least it allows users and developers to see the problem clearly.

We have good reason to believe that universities and colleges could collectively produce open source software that meets their needs better than commercial products. If the theory posited earlier is valid, that poor performance of some commercial products is due largely to the disconnect between developers in software firms and users in academia, then developers from within academia ought to be at an advantage. They understand their internal operations and needs well. Universities and colleges also have access to a large pool of skilled developers, particularly within some of the larger institutions. They have demonstrated the ability to produce high quality products (the course management programs Blackboard and Prometheus came from Cornell and George Washington universities, and the Mosaic web browser and NCSA web server – the ancestors of today's Firefox and Apache – originated in the University of Illinois), though whether they have the ability to produce enterprise level software is a subject of debate.

Finally, as mentioned earlier, a number of institutions have built home grown systems that perform well or are actively considering the buy vs. build decision. As institutions contemplate the build option, collaboration offers great appeal. The simple arithmetic of dividing by N as N increases from one to two to three can be a powerful economic driver. At the same time, coordination costs rise with N, probably more than linearly, so the optimal size of any such collaboration in development is still small – 3 to 6.[17] We also came across a number of institutions that had built valuable administrative modules that they would be delighted to share, were there a convenient mechanism for doing so. At DePauw University, for example, an enterprising software architect and developer in the IT group built student information, HR and development systems that appear to meet the institution's needs. DePauw would be happy to share this software with other institutions, but at present there is no obvious way to go about it.

It is worth reiterating here that administrative and course management software is not viewed as an area of competitive differentiation among universities and colleges. Collaborating with peer institutions in these areas thus makes a great deal of sense to the leadership, whose primary goal is to get the best performance for the least cost.

One element that is clearly needed to facilitate this kind of sharing and collaboration is better information about what others are doing. In making buy vs. build decisions, there is no systematic way for CIOs to know what other institutions would be interested in sharing the cost of building. Moreover, valuable software may be created at one university or college but never used elsewhere. Often this happens because no one else knows that the software exists, or because the software is not designed in such a way that it can be easily implemented at different institutions. Uncovering the hidden gems in the community and turning them into shareable open source software could unlock a great deal of value.

There are some early signs that this approach can be effective, though its long term sustainability is far from proven. DSpace has nearly 150 installations worldwide and a community of around fifty volunteer contributors. uPortal competed effectively against a variety of alternative products and is in production at over 80 institutions. Sakai has attracted funding from around 90 partners and is being piloted at 39 institutions.[18] These projects all aim to reach a critical mass of adopters and transition to the community development approach, in part because it has a proven sustainability model – once a

[17] Savings would be greater if added perspectives enhance quality, and less to the extent that there are coordination costs. As the number of partners goes up, the coordination costs will dominate, so the optimum number is likely fairly low.

[18] These installation figures are all as of April, 2006.

critical mass of individuals and institutions has adopted or invested in the software, they share an interest in seeing it live and grow. Mozilla is commonly cited as an example of a project that successfully navigated this transition – its base of contributors is now much larger than it was while Mozilla was perceived to be part of Netscape. uPortal has reached a point where some believe it could be pushed out to the community and survive. This approach is not yet proven in higher education, and we should all be watching efforts like Sakai closely.

We believe there are several conditions that seem to enhance the chances of success of directed development projects. One critical one is that they can build on existing code that works at least fairly well (approximately as well as a typical commercial installation, at least relative to cost), as was the case for Sakai and Kuali. Starting with a working code base precludes endless negotiation and debate over basic architectural decisions. It also increases the odds that the project can achieve early wins that demonstrate its potential value to the community and thus build momentum. Second, it is desirable that the principals of the project have functional expertise regarding the purpose and use of the software. Third, according to several people who have been involved in collaborative projects, collaborations should probably involve somewhere between three and six institutions. Two institutions may have too many ties when making decisions and produce software that is too idiosyncratic to be useful to other institutions. For more than six, the costs of collaboration are likely to become excessively high.

There are substantial organizational and legal barriers to effective cooperation of the kind that we think is most valuable, and there does not yet exist a proven organizational model for projects as they move towards production. Sakai, DSpace, and (in prospect) Kuali all seem to be reinventing similar wheels (though Kuali has followed closely in Sakai's footsteps). We have also observed a number of organizations and initiatives created over the years pursuing various forms of collaboration in IT, many of which seem to have achieved less than was originally hoped. One (perhaps obvious) lesson we take from these examples is just that collaboration in IT is very hard – institutions have difficulty compromising on requirements and priorities, individual developers have their own beliefs about the best ways of doing things, and incentives are difficult to align. Another lesson is that the success of collaborative efforts requires vigorous and active leadership from the highest levels of administration, including presidents. There must be commitment to the collaboration at each level, but the pressure to bring about that commitment must emanate from the top. The implication here, we think, is that any structure, be it project by project or a version of OOSS, must be commissioned and governed by an entity that has both substantial authority and a clearly agreed-upon sense of mission. (We do not view Sakai or DSpace as exceptions here. In both cases the relevant presidents and provosts, as well as CIOs, have attended closely to the work done by the developers, and have provided support both financial and in other forms.) Creating an effective governance and board structure is essential.

To summarize, we believe that software projects (both in higher education and more generally) work best when there is clear mutual understanding between the users and the developers regarding how the software is to be used and what is important for it to accomplish. The success of many community-based open source projects derives from just such a confluence. Our examination of a number of OSS projects, both community developed and developed via directed collaboration, is consistent with the importance of detailed, substantive engagement between the supply and demand sides in development.[19] Lack of such engagement is common in higher education and in many other industries, and is especially visible in higher education when commercial products are employed. The biggest potential payoff to OOSS would be in correcting the coordination failures between users and developers in a more systematic way.

[19] Reference case studies to be posted on Ithaka website.

Section VI: Policy Options and Conclusions

The case for directed development of open source projects designed to serve the operations of institutions of higher education in the United States seems to us to be an overwhelming one. The current market does not serve these institutions as well as it plausibly could, in no small part because the people writing the code are far removed from the operational needs of the people and institutions who are using it. This problem is not unique to higher education, of course, but higher education is in many cases more different from other sectors, and has more internal variability than most other sectors. And, with the exception of the information technology industries themselves, higher education has more internal capacity to provide for itself than do other industries.

Directed development has two especially valuable features. One is a signature of the open source and open content movements – many eyes, and many keyboards, can improve the quality of code through a process of continual review and testing. More important, however, is that when one or more institutions are considering building their own solution to an enterprise problem, spreading the costs over even a fairly small number of partners can greatly reduce the cost to each. At the same time, there should generally be advantages spread much more widely, and it makes sense for many institutions to look over the shoulders of the few that will be at the center of any one development.

Convinced as we are of the value of collaborative, directed open source projects to higher education, we are not settled on an ideal mechanism to support their development and deployment. In increasing order of organizational complexity and scope, we see a continuum of activities that could be undertaken in service of the mission.

Jawboning. Presidents, provosts, and CFOs are often far more skeptical about the value of OSS than is warranted, and are excessively skeptical of the extent to which such solutions can be sufficiently businesslike. The contrary point of view, drawn from experience and summarized in this report, should be promulgated actively in various forums, including AAU, ACE, and NITLE. The Common Solutions Group could also help make the case, and, indeed, more discussion between senior academic leadership and CIOs and CTOs would be helpful in any case. It will be especially valuable to involve proprietary firms that are seeking to support the use of OSS projects in higher education, as for many users the existence of such firms and commitments from them will be essential to adoption of open source products.

There would also be value in certain advocacy and education activities related to licensing. One would be educating developers about the importance of respecting intellectual property rights and making the work of the Software Freedom Law Center, which may be a useful partner in addressing licensing and governance issues, more visible to both academic leadership and associated project leadership. A more ambitious goal would be to seek agreement on a standard form of contribution agreement that will be acceptable to nearly all institutions, which would greatly reduce administrative costs and help ensure fairness and consistency between institutions, and also to try to encourage standardization around a limited number of forms of open source license.

Jawboning Plus. Making the case for the use of directed development open source would be facilitated if it were someone's job. Ideally, this function would be performed in an institutional setting that already has credibility with both academic and technical leadership in higher education. It is plausible, but by no means certain, that this activity, done well, would be self-sustaining, because a person in such a position would be well-situated to provide consulting services and to find good matches between open source projects, possible users, and commercial entities interested in selling related services.

Market-Maker Plus. Here we describe what we see as the minimal version of OOSS. It would have two purposes. With respect to projects in prospect or in their early stages, it would provide expertise, informational and logistical support to pull together a project team and a set of sponsors, the latter generally being higher education institutions. With respect to mature open source projects, it would provide a set of "Consumer Reports" – like functions, regarding aspects of the software itself (including ease of modification and use) and the quality of commercial or noncommercial support of the software. It could also audit and certify the licensing practices of open source projects, and help promulgate best practices for dealing with these issues. (There are potential conflicts between the facilitating and evaluating functions, an issue that we address later.)

Our strongest finding is that there is likely a good deal of benefit, both direct and indirect, to cooperative, directed open source ventures among institutions of higher education. Although there is a good deal of sharing of best practices and ideas at the technical level (CIOs and developers) and a good deal of sharing of problem strategies at the level of provosts and presidents, putting together a team across institutions, with committed executive sponsorship, continues to be very difficult. As one CIO – a fan of collaborative projects – pointed out to us, it is so difficult that even for fairly simple things, such as shared off-site storage facilities, the extra monetary cost of producing at suboptimal scale is often seen as small compared to the coordination costs and delays of shared enterprises. And the continuing story of NLR and Internet2 is a cautionary tale both about the difficulties of coordination and the importance of high-level executive engagement.

A potential solution to these coordination problems would be to establish an organization whose mission it was to solve them. This would require a director at a fairly high level, someone who had the confidence of CIOs, the ability to talk with Presidents and Provosts, and who was willing to stress the importance of agreement and engagement between senior academic leadership and senior technical leadership. Operationally, the market-making operation would look something like the following: With the help of staff, OOSS would develop expertise with respect to the desires and interests of academic leadership and of efforts made within HE to develop new products, and would seek good fits. (We would expect that OOSS would employ rules much like the Apache Software Foundation, requiring existing code and a kernel of institutions and developers.) Suppose, for example, that a number of institutions were working on student administration systems. OOSS would know who was working (both among the big universities and smaller institutions) and who was interested. It might call a group together, both demanders and potential developers, and have a "bake off," in which the group would settle on one or two approaches. OOSS would then help put together a manageable consortium to develop the project, and would also help with an interim governance structure and the identification of a larger community of interest.

Over, time, the market-making activity would also feed back to the supply side. OOSS would learn of some suite of products of interest to many institutions, and would be well enough informed about capacities to make judgments about how best to proceed. More generally, OOSS could share best practices on both sides of the market, as well as in developing successful collaborations. The expertise that would be essential in making markets for new products would also be valuable in performing the "Consumer Reports" function for existing products. It is not clear that OOSS could be sufficiently arms-length over time, as projects that it had helped to nurture moved from the development stage to maturity. Given that both the market-making function and the CR function are clearly valuable, and that similar (often identical) knowledge would be required to do each job well, it would be very appealing to put both functions in one organization if possible. We note that, as in the Jawboning version, it is plausible that OOSS would have valuable consulting services that it could sell, complementary to its core functions. An important advantage of this model for OOSS is that costs of failure or liquidation would be quite low, and it could still create value for the community along the way.

A weakness of the minimalist version of OOSS that we have sketched above is that it has limited leverage. No institution or set of institutions would be required to use it, and its advice would be just that – advice. Moreover, it would be limited in its ability to provide legal advice to projects (as it would not be a law firm, and ethical rules prohibit combining the provision of legal advice to third parties with other services) nor would it own any projects, and hence it could not impose conditions on projects, either substantively (e.g., with regard to interoperability) or legally (e.g., with regard to relatively simple and effective licensing arrangements). However, the existence of the Software Freedom License Center could greatly mitigate the problems here, provided that the firm had sufficient capacity or could develop such expertise in HE as might be necessary. Further, it seems likely that in this configuration successful projects would tend to each migrate to individual governance structures. Ideally, OOSS could encourage consolidations within the existing set (or a slightly expanded set) of "foundations," but there would be nothing to force such consolidations, absent powerful pressure from the leadership of the contributing institutions.

All of these limitations make it likely that higher education will fall short of the most successful set of outcomes that we discussed earlier. If the best outcomes were to be realized in this version of OOSS, there would have to be persuasive commitment from the senior leadership of a fairly large set of institutions both to support OOSS and to attend seriously to advice that it would give. In summary form, there would have to be a strong and engaged board, made up not only of presidents and other campus leaders, but others with direct experience in software development and related technologies. Presidents would have to be willing to use their standing and influence within their universities, especially in connection with attempts to standardize the terms on which universities participate in these projects as contributors. In order for OOSS to succeed, Provosts and CFOs, in collaboration with their CIOs, would have to engage with it as a matter of routine when considering major projects. We expect that such engagement will improve the quality of collaboration between academic and technical leadership on campus. This would be a mixed blessing for CIOs, who will benefit from closer alignment with the academic leadership, but who would also likely have less autonomy with regard to strategic decisions about administrative systems.

Conservancy. Some of the weaknesses of the preceding framework would be dealt with if OOSS held the intellectual property deriving from the directed development projects, and either distributed or licensed to a subsidiary the distribution of the associated projects. This configuration would require similar leadership and staff support. There would be an important gatekeeping function (implicit in the market-making function), as OOSS would require projects to meet certain conditions for entry, such as high quality documentation and licensing standards. Projects would still operate fairly independently but would compete to get into OOSS, which would provide support to the projects and deliver a valuable imprimatur or brand. Of course, those institutions would have had to agree in advance to this arrangement, but having done so, the ability of OOSS to coordinate at all levels and in both operational and organizational domains would be greatly enhanced. Legal issues would also be simplified, in that OOSS could have its own legal counsel, which could provide legal services directly to the open-source projects under its control without running afoul of the rules prohibiting mixing the practice of law with other businesses.[20] The number of separate governance structures could be sharply reduced, and one-time costs (incorporation, obtaining a letter ruling from the IRS, forming an advisory board) and recurring costs (tax returns and periodic corporate filings, convening board meetings) could be greatly reduced. (Note that OOSS would not represent the individual contributors

[20] While OOSS could review licensing practices and otherwise evaluate legal issues on behalf of the HE community for open source projects established as separate organizations, its counsel would be better positioned to provide candid advice while maintaining the confidentiality of that advice in a context where the projects are within the same organizational structure.

of code to the projects, and thus there could be confusion, especially for developers, about attorney-client relationships.)

A disadvantage with positioning OOSS as the holder of intellectual property is that failure could be messy. Liquidation of the organization could leave projects scrambling for a new home for their IP and other assets. It would be essential that there be reasonable provisions to dissolve the organization should that be desired. Additionally, it would be more difficult for this version of OOSS to provide the Consumer Reports functions, given its ownership of some projects. However, it would be valuable to figure out some way to take advantage of the complementary expertise.

The launch of the Software Freedom Conservancy both validates this vision for OOSS and presents potential competition. Some projects, given a choice, may prefer to place their IP with the SFC, which appears to offer a governance solution and some services with limited strings attached. That said, it is not clear what the SFC's criteria for admittance are, and we would hesitate to rely too heavily on an entity outside the community for an issue of such importance. If, over time, the SFC proves to be a robust solution, it could become an appealing option.

In the meantime, if presidents were willing to provide the level of sponsorship and engagement necessary to make the Market-Maker version of OOSS successful, it is but a small step to this version, which would be a much more powerful institution, better able to articulate and deliver on the mission. Again, the Board would have to include significant representation of university and college leadership.

Software Development Company Supported by Subscription. We also considered a model in which OOSS would employ developers and its leadership would determine which OSS projects to work on. It would own the projects, provide the same kind of branding as the previous model, and would be able to enforce – because of common ownership and design – interoperability at a high level. Because much of the development work would be in-house, the annual payroll would be several millions of dollars a year, perhaps as much as ten million dollars, in contrast to something like a million dollars in the Owner/Distributor configuration and somewhat less for the Market Maker.

There was essentially no support for this model, either from academic or technical leadership, in our interviews. University leaders do not have the appetite for creating a large, centralized organization in this space, and they are very wary of stamping out grassroots innovation. The more advanced OSS projects value their independence and would be reluctant to be rolled up into such an organization. A number of people expressed concerns about how resources would be allocated and how it could prioritize the needs of different institutions.

In weighing the more palatable alternatives, the key question is the level of engaged commitment obtainable from university leadership, both from the larger universities that are likely to provide personnel to write code, and from smaller ones, who have demands of their own. Without a powerful commitment, none of these models can succeed. Our own view is that with such a commitment, either the Market Maker or Conservancy model would be worth trying, with the chances of success somewhat greater for the latter.

In conclusion, we have found that there is good reason to believe that some market failure is taking place in the realm of administrative systems and that there is a real opportunity for the community to pursue a world with more effective and cost efficient solutions and a more robust marketplace. We are convinced that collaborative efforts to build open source applications can produce software that better meets the needs of partner institutions and also has the potential to benefit the broader community, should these efforts attract an expansive base of contributors over time. This belief is largely founded on the hope of overcoming the historical disconnect between producers of software and HE users, who have complex, unique, and poorly understood needs. It is also based on the fact that HE has a fairly robust tradition of building its own software.

Indeed, to some extent collaborative software projects are already taking place without any coordinated intervention. We believe, however, that the benefits of these activities could be accelerated and broadened by more systematic efforts to match up institutions interested in building administrative software and by providing some much needed information and services to both users and producers of that software. We have concluded that it would be worthwhile creating an entity to address these needs, but only if senior leaders are committed to being actively involved and if that organization has sufficient leverage to pursue its mission effectively. It will be necessary to consult with representative leadership to determine if this level of commitment is feasible. Without it, jawboning is the best strategy.

Appendix A: Interviewees

Art Pasquinelli	Group Marketing Manager, Global Education and Research, Sun Microsystems
Barron Koralesky	Academic Information Associate, Macalaster College
Barry Walsh	Executive Director, Kuali Project; Director, University Information Systems, Indiana University
Ben Hyde	Member, Apache Software Foundation
Bill Wrobleski	Director of Technical Information Operations Division, University of Michigan
Bob Carroll	Former CIO, University of Phoenix
Bradley Wheeler	Vice Chairman, Sakai Foundation; Associate VP for Research and Academic Computing and Dean of IT, Indiana University
Brian Hawkins	President, Educause
Brian Rosenberg	President, Macalaster College
Carl Jacobson	Former Project Liaison, uPortal; Director, MIS, University of Delaware
Chris Coppola	President, rSmart Group
Chuck Severance	Chief Architect, Sakai Foundation; Software Architect, University of Michigan
Daniel Updegrove	VP for IT, University of Texas at Austin
David Ernst	CTO, California State University; Board Chair, IMS Global Learning Consortium
David Foster	Dean of Advanced Programs, Devry University
David O'Connor	VP of Product Development, Higher Education, Pearson Education
David Wheaton	VP for Administration and Treasurer, Macalester College
Dennis Murray	President, Marist College
Dennis Trinkle	Associate Vice President for Academic Affairs, Chief Information Officer, DePauw University
Diane Michelfelder	Provost, Macalaster College
Dorothy Yancy	President, Johnson C. Smith University
Edward Walker	Former CEO, IMS Global Learning Consortium
Frank Prochaska	Executive Director, University of North Carolina Teaching and Learning with Technology (TLT) Collaborative
Gordon Freedman	VP, Education Strategy, Blackboard
Harry Williams	Director, Technology and Systems, Marist College
James Dalziel	Leader, LAMS; Chief Investigator, Australian Service for Knowledge of Open Source Software (ASK-OSS)
James Hilton	Associate Provost for Academic, Information, and Instructional Technology Affairs, University of Michigan (now VP and CIO, University of Virginia)
Jerrold Grochow	VP of Information Services and Technology, MIT
Joel Smith	Vice Provost and CIO, Carnegie Mellon University
Jim Farmer	Chairman, instructional media + magic, inc.
Jim Krailler	Manager of Instructional Support Technologies, Cincinnati State College
John Blakely	CEO, Unicon
John Etchemendy	Provost, Stanford University
John Gohsman	Director of Student Administration and Human Resources Management System Divisions, University of Michigan
John Meerts	VP for Finance, VP for Information Technology, Wesleyan University
Joseph Hardin	Board Chair, Sakai Foundation; Clinical Assistant Professor of Information, School of Information, University of Michigan

Julie Walker	Project Liaison, DSpace; Senior Business Strategist, MIT Libraries
Kathleen McNeely	Functional Council Chair, Kuali; Interim Assistant VP for Finance, Indiana University
Kevin Hegarty	VP, CFO, University of Texas at Austin
Ken Klingenstein	Director, Internet2 Middleware and Security (Shibboleth); Former Director, Computing and Network Services, University of Colorado
Laura Patterson	Associate Vice President, Administrative Information Services, University of Michigan
Lois Brooks	Director, Academic Computing, Stanford University
Mackenzie Smith	Project Director, DSpace; Associate Director for Technology, MIT Libraries
Mark Armstrong	Vice President for Student Products, Oracle
Mark Shuttleworth	Founder, Shuttleworth Foundation
Martin Ringle	CTO and Director, Computing, and Information Services, Reed College
Michael Zackrison	VP, Product Management, Unicon
Mitchell Kapor	President, Board Chair, Open Source Applications Foundation; Board Chair, Mozilla Foundation
Neal Abraham	VP for Academic Affairs, Dean of the Faculty, DePauw University
Patrick Carey	Higher Education Leader, IBM Business Consulting
Paul Gray	Executive Vice Chancellor and Provost, University of California, Berkeley
Paul Mutone	VP for Business and Finance, CFO, Marist College
Peter Lange	Provost, Duke University
Philip Long	CIO, Yale University
Randy Ebeling	Associate VP and COO of Information Technology Services, University of Texas at Austin
Robyn Render	VP, Information Resources, University of North Carolina; Board Chair, Educause
Rob Abel	CEO, IMS Global Learning Consortium
Scott Siddall	Assistant Provost and Director of Instructional Technology, Denison University
Shel Waggener	Chief Information Officer, University of California, Berkeley
Sheldon Ekland-Olson	EVP, Provost, University of Texas at Austin
Stefano Mazzocchi	Member, Apache Software Foundation
Steve Midgley	Program Manager, Stupski Foundation
Steve Pappageorge	Director of Administrative Operations, Devry University Online
Steven Carmody	Security Architect, LionShare; IT Architect, Brown University
Stuart Sim	Senior Architect, Global Education and Research Group, Sun Microsystems; Founder, Education Commons
Ted Fine	Assistant Director for Networked Services, Information Technology Services, Macalester College
Walter Massey	President, Morehouse College

Appendix B: An Annotated Inventory of Products

For each of the items on the list below, we discuss briefly the importance of the area and the perceived risks and shortcomings of the current state of affairs. Where the area is important and the risks and shortcomings are substantial, there is a prima facie argument for change.

Financials/purchasing/physical assets/space. These are traditional business activities that are required of any enterprise, and the case could be made that higher education should be able to find good "off-the-shelf" solutions in the commercial sector. Two arguments militate against that simple case. First, as evidenced by the fairly recent home-grown system at Indiana that is a basis for the Kuali project, as well as the very successful home-grown systems at places as disparate as Texas and DePauw, a number of serious institutions find that commercial products do not serve them well, and have put their money where their mouths are. Second, and related, the dozens of institutions that implemented PeopleSoft at great expense are still frustrated by some inflexibility, and are concerned that the new Fusion product that will come from the acquisition of PeopleSoft by Oracle will itself be expensive to implement, be similarly inflexible, and will pose the risks inherent on being reliant on a single dominant vendor for an essential part of doing business. Smaller institutions also report that products aimed at their market segments are expensive and often inflexible. At the very least, higher education leadership would rest easier if there were more competition in this set of domains.

HR. Exactly the same set of considerations apply to Human Resources administration, although here commercial products are generally more adaptable to the HE setting.

Course management. Course management is a core activity of colleges and universities. The leading commercial product is Blackboard, which recently acquired its leading commercial competitor, WebCT. An open source project, Moodle, has been widely adopted, generally by smaller institutions. It is highly regarded but does not have sufficient functionality for the largest and most complicated places. This may change as the Open University in the UK, which serves nearly 200,000 students, recently made a decision to implement Moodle, and presumably any enhancements it makes to the code will be shared with the Moodle community. Sakai, which is currently being piloted at Michigan and Indiana, as well as 37 other places, is a more complicated and richer open source product. A number of commercial vendors are providing operational support for Sakai and for Moodle. It seems likely, but is by no means certain, that directed development within HE should succeed in this domain if it is to succeed anywhere. Flexibility in application and in upgrades seem to be key issues for users, while leadership is concerned about potential monopolization, especially at the high (complicated) end of the market. A number of related open source education tools (e.g. VUE) have been launched, and interoperability among these applications will be critical.

Student administration/financial aid/admissions/registrar/grants management. These areas are similar to financials in that users are concerned about cost, functionality and flexibility, and the potential for monopoly, especially following the demise of PeopleSoft. But, unlike financials, student administration is peculiar to the education sector, so it may be an even better candidate for collaboration in development. Several schools have expressed interest in an open source SIS, and Indiana University has recently received a planning grant from the Mellon Foundation to study how the University of British Columbia's single-institution system could be transformed into a directed open source initiative.[21]

[21] See http://chico.nss.udel.edu/jasig9/popAbstract.jsp?id=1525b198 for a brief description of the project.

Rights management. Copyrighted materials are subject to a myriad of rights management issues, greatly complicated by the fact that the permitted use of some materials varies by location of the materials, and the location and identity of the user. The result is that it is difficult (inter alia) for a faculty member to know what can be put up on a personal website or a course site. Software to help with this problem would be of great of value, probably uniquely to HE. This is a set of problems that is not going to get better on its own. We know of no efforts in this domain currently under way.

Automated CV. When asked what new software needs might arise over the next five years, one provost suggested a system for managing data on the professional activities of faculty. This software would allow faculty to keep their records updated and generate CVs in any format desired for offline purposes. The data would also be available in a searchable database for deans or provosts to use when making advancement and compensation decisions. As far as we know, there are no commercial products available to do this, though some departments have jury-rigged their own solutions.

Development. This is a peculiar area in that it is viewed as a source of competitive advantage for many institutions, but most continue to rely on poor quality, outdated systems. Although it fits our definition of a system where collaboration would make sense (requirements are unique to HE), it is not clear that universities and colleges would be open to collaborative efforts because this area tends to be so jealously guarded.

Institutional repository. This is a relatively new piece of infrastructure in many institutions, frequently under the purview of the library. The growth of institutional repositories has been stimulated by the recognition that a great deal of faculty research and research inputs (notes, datasets, etc.) are produced, stored and shared electronically, and that their host institutions might like to aggregate and preserve the intellectual output of the faculty in one place. (It is not yet clear that faculty see the value in this, so many institutional repositories are in need of content.) Several commercial (BE Press) and open source (DSpace, Fedora) products are currently available.

Security. The primary functions of security software are to protect the network from sinister infiltrators such as viruses and spammers and to identify members of the community. It is an area of major concern to IT administrators. Many open source (SpamAssassin) and proprietary (Symantec, McAfee) products are available for network protection. Shibboleth is an open source initiative of the Internet2 to create an authentication system.

Compliance. A subject that came up repeatedly in our conversation is the increasing difficulty of compliance, particularly due to Sarbanes-Oxley. All systems that generate reports for the federal government, notably financial aid and management of federal grants, must be modified each year to produce the appropriate data in compliance with new requirements. Oracle, PeopleSoft, and SAP, as well others, provide updated financial aid modules in a timely way. Practice with respect to financial administration and reporting on grants varies widely across institutions. Because these reporting requirements affect nearly every institution, it seems like there would be an advantage to pooling these efforts.

Email, Calendar, etc. All members of a university or college community rely upon basic applications such as email and calendaring. There are many products available, and MS Office is the most widely used of these. OpenOffice is an open source set of products that appears to provide basic functionality, but that probably would not meet the needs of most campuses. Chandler is an open source effort launched by the Open Source Application Foundation, which is funded by the Common Solutions Group and the Mellon Foundation. One of its objectives is to produce an open source shared calendaring solution, but this effort seems to have fallen behind schedule.

Portal. Campus portals allow users to create customized versions of the campus Web. They also provide community tools, such as chat, forums, survey, and so on, that can build relationships among campus constituencies. A university-led open source project, uPortal, is in production at over eighty institutions.[22] One remaining commercial competitor, now called SunGard Luminis, rebuilt its product around uPortal.

Library OPACs. These systems are used to catalogue library holdings. There are a number of commercial products available (Ex Libris, Endeavor), but consensus seems to be that these systems are clunky and outdated. One theory we heard is that vendors are reluctant to invest in upgrading these systems because the function of libraries is in such a state of transition, and it is not at all clear what activities the software will need to support five to ten years form now. A number of people speculated that an open source OPAC would make sense, though the same challenges would apply.

Sponsored projects. Federally sponsored projects carry with them a host of financial and regulatory requirements that are unique to the set of institutions, including higher education and freestanding research institutions that carry out sponsored research. The regulatory requirements are especially difficult when human subjects are involved, as research protocols must be approved by Institutional Review Boards, and actual practice must be monitored. It is easy to violate the letter of the rules, especially with regard to the timing of required renewals of approvals and the like, and the financial and legal liabilities associated with violations can run into the tens of millions of dollars. While a number of the ERPs have developed software for research administration, most institutions have found the commercial products to be far too generic to meet their needs, and the number and types of solutions (some much less automated than others) in use is highly varied. The payoff to a successful directed development project in this domain would be high; the challenges in producing one would be considerable.

[22] For a list of uPortal institutions, see http://www.uportal.org/who-prod.html (accessed May 7, 2006).

Appendix C: List of Possible Activities for an OOSS

In each of our meetings we asked what the interviewee would like to see an OOSS do, if one were to be created. While there was near unanimity that some kind of an organization would be valuable, there was no consensus at all as to what it would look like. We have put together a full list of possible activities for OOSS from both the user and developer perspective. We have also commented on the feasibility of engaging in these activities. They are presented roughly in order from least controversial to most controversial.

Legal services for open source software projects. The need for assistance with licensing and governance issues was the one common theme we heard from both users and developers. It is clear that provision of these services would be of great value. However, it is important to note that OOSS would not be able to provide legal advice directly to third party projects in combination with other activities. Unless the OSS projects were brought under the same legal entity, OOSS could only act as a referral service to a law firm or other entity dedicated to providing legal advice. These issues are explored in more detail in Section III.

Information clearinghouse about open source. OSS-Watch was created by JISC to promote awareness and educate the higher education community about open source software. It publishes reference cases and best practices to facilitate the adoption of OSS. There seems to be value in this set of activities, and we found some interest from both the demand and supply side in creating a U.S. version.

Broker collaborations among institutions interested in sharing the costs of building. We found a surprising number of institutions that have built or are contemplating building their own administrative systems. Assuming there are some basic commonalities in their functional requirements, there seems to be huge potential value in aggregating information about what institutions are doing and acting as a matchmaker for collaborative efforts. In some ways the Mellon Foundation has played this role in the cases of Sakai, Kuali and most recently a planning grant for a student information system. We can see great potential in performing this activity on a broader, more systematic scale, and found that many leaders would be interested in supporting it.

Consumer reports service. On the demand side, many interviewees identified a need for better access to information about both commercial and open source products. For OSS, which is perceived as higher risk, this service would be particularly valuable in certifying quality and licensing practices. The ability to access this information from a trusted third party would potentially give universities and colleges greater confidence in using open source and promote wider adoption. We uncovered two reservations regarding this service: one, other organizations/initiatives such as Gartner, Educause, Business Readiness Reading (BRR) and the Open Source Institute (OSI) already provide variations on this service. Second, conducting very detailed evaluations of software packages is quite expensive; individual institutions might prefer to use their resources to evaluate a specific application they need to implement rather than contribute these funds to a general pool. Even so, we found considerable interest in this service.

Incubator of OSS projects in HE This concept could be modeled after the Apache Foundation, but tailored to the needs of the higher education community. We heard a fair amount of interest in it from OSS projects that feel a need for better governance arrangements. The benefit of this model is that it would provide a legal home for open source projects and reduce the overhead head costs associated with setting up separate non-profit organizations for each one. It also ensures that ownership or at least a license to distribute the aggregated code resides somewhere safe. Moreover, membership in the incubator could send a signal of quality to the community and increase a project's adoption prospects. The tension is that the incubator would need to have a reasonable level of involvement in the management of the software projects, but most projects do not desire that level of supervision. There is also a concern that the projects most likely to succeed on their own would be less likely to want to join.

Provide support services to projects, such as low level infrastructure, collaboration tools, documentation, business planning and marketing advice, organizing conferences and seminars. The OSS projects were enthusiastic about having access to these types of services. However, it is not clear to us that providing technical and documentation services would significantly improve their chances of success – projects generally figure out ways to meet these needs on their own if they have to. We did not hear a strong argument for creating a non-profit to take on tasks that are merely tedious (as opposed to requiring scarce expertise). Moreover, university leaders are not particularly interested in funding these activities. We do, however, see more value in providing business planning and marketing advice, as this is an important kind of expertise that OSS projects may not have access to internally. We heard several times that each project has to "reinvent the wheel" in all of these areas, and providing these services would allow them to focus on more value-added areas. We have also heard that members of the open source community and IT staff benefit greatly from attending OSS project conferences (some Sakai partners said this was the main value they got for their membership fees). If organizing these conferences proves to be a major distraction for these projects, this may be a useful service to provide.

Provide support services to institutions. Several OSS developers expressed a concern that they would have difficulty attracting commercial support providers before a market for their software is established. They thought it would be valuable for an OOSS to provide support for their products, at least during an interim period, and argued that OOSS could do so more economically than commercial providers because it would not need to earn a profit margin. However, there are several persuasive reasons why OOSS should not provide this service. First, no one on the demand side was interested in subsidizing an organization to provide support services to themselves. Second, commercial VARS, which are a necessary part of a healthy open source ecosystem, might be less likely to enter a market for services in competition with an organization funded by their prospective customers. Third, it is not obvious that a non-profit entity could provide services more efficiently (and at lower cost) than commercial vendors in the absence of monetary incentives.

MacArthur-type grants for open source developers. This suggestion came from a couple people on the project side who would like to see more financial support directed at OSS, but are wary of administrators trying to exert control over the projects. An argument in favor of this approach is that the success of projects like Apache, Moodle and Linux is often attributed, at least in part, to exceptional leadership. It stands to reason that people like Brian Behlendorf, Martin Dougiamas and Linus Torvalds merit financial support, and that their managerial and technical decisions should not be second-guessed. However, university leaders would prefer to fund their own "stars" rather than pooling and relinquishing control of their resources.

Centralized organization for developing, maintaining, and supporting software. This model has several advantages of the proprietary software model: it would avoid some of the pitfalls and coordination costs associated with collaborative projects, it would provide users with the comfort of knowing where to turn if its products break, and it could enforce interoperability among projects (the inability to do this is viewed as a major shortcoming with laissez faire). Some would argue that this level of centralization and control is essential to producing enterprise level administrative applications. These advantages would, in theory, be combined with certain benefits of open source, such as many pairs of eyes producing better quality code. Several people consulted for this study thought that such an entity would be great, if it were feasible. However, our interviews have persuaded us that university leaders do not have the appetite for creating a large, centralized organization in this space, and they are very wary of stamping out grassroots innovation. The more advanced OSS projects value their independence and would be reluctant to be rolled up into such an organization. A number of people expressed concerns about how resources would be allocated and how it could prioritize the needs of different institutions. Overall, the feasibility of this model appears to be low.

Appendix D: Attendees at October 2005 Meeting

Chair:
John Hennessy, President of Stanford University

Attendees:
William G. Bowen, President, Mellon Foundation
Molly Broad, Chancellor, University of North Carolina – Chapel Hill
Mary Sue Coleman, President, University of Michigan
Paul Courant, Professor of Public Policy and Economics, University of Michigan
Jackie Ewenstein, Assistant General Counsel, Mellon Foundation
Ira Fuchs, Program Officer, Mellon Foundation
Kevin Guthrie, President, Ithaka
Rebecca Griffiths, Manager of Strategic Services, Ithaka
Mark Kamlet, Provost, Carnegie Mellon University
Martha Kanter, Chancellor, Foothill-De Anza CC
Pat McPherson, Vice President, Mellon Foundation
Mike McPherson, President, Spencer Foundation
Michael McRobbie, CIO, VP for Research and CIO, Indiana University (now Interim Provost and VP for Academic Affairs)
Dennis Murray, President, Marist College
Don Randel, President, University of Chicago (future president, Mellon Foundation)
Larry Ricciardi, Trustee, Mellon Foundation and Ithaka
Judith Shapiro, President, Barnard College
Chuck Vest, Former President, MIT
Don Waters, Program Officer, Mellon Foundation

Appendix E: Existing Initiatives and Organizations Related to IT and OSS in Higher Education

Organization	Mission	What it does	Governance and support
Alliance for Higher Education Competitiveness (A-HEC)	Develops and disseminates best practices through research focused on innovation, transformation, and effectiveness in higher education	Conducts studies and publishes research. Recently issued a report on open source	Nonprofit, governed by a small board. Founder/president recently appointed to run IMS (see below)
Apache Software Foundation (ASF)	Provides organizational, legal, and financial support for a broad range of open source projects, most notably the Apache web server. Offers an established framework for intellectual property and financial contributions	Umbrella support organization for over 30 open source projects. Sets norms for decision-making and governance by merit. Incubates new projects	Membership elects a nonprofit board of "committers," respected members of the Apache community whose contributions are highly valued
Australian Service for Knowledge of Open Source Software (Ask-OSS)	Serves as an information clearinghouse for open source in Australia	Modeled on OSS-Watch UK. Publishes news, case studies, and information on licensing and intellectual property. Gives advice to OSS projects	Supported by the Australian government's Department of Education, Science, and Training
Business Readiness Rating (BRR)	Establishes standards for community rating of open source software and evaluates OSS using those standards	Publishes Consumer Reports-type studies of open source projects, assessing them for functionality, security, support, community, and other quality metrics	Sponsored by Carnegie Mellon West Center for Open Source Investigation, O'Reilly CodeZoo, SpikeSource, and Intel. Steering committee is composed of industry representatives and chaired by the executive director of CMU West's Center for Open Source Investigation
Common Solutions Group (CSG)	Helps IT leaders at elite research universities identify and understand strategic information technology, adopt common solutions to common problems, and develop a map for the future of IT in higher education	Organizes conferences for participating CIOs. Supports some research and open source projects financially (e.g., Chandler/ Westwood)	Membership organization that elects a coordinator and treasurer. Members are institutions of higher education and a limited number of higher education IT consortia

Eclipse Foundation	Serves as a home for an open source community whose projects aim to provide an extensible development platform and application frameworks for building software	Like Apache, an umbrella for several open source projects. However, Eclipse projects are more closely related to one another than Apache projects	Governed by a board with four classes of membership. Led by corporations such as IBM, Nokia, CA, Intel, HP, SAP, SyBase, and Red Hat
Education Commons	Supports an open and transparent system of communication between diverse groups committed to advancing the state of education worldwide	A new non-profit founded by Sun to promote standards and open source software in higher education and thereby advance teaching and learning	Supported by Sun
Educause	Advances higher education by promoting the intelligent use of information technology	Professional association for higher education IT staff. Also provides benchmarking data and publishes research to inform IT decision-making	Governed by a board of higher education CIOs. Receives financial support from commercial vendors
Eduforge	Provides an open access environment for sharing of ideas, research outcomes, open content, and open source software for education	Akin to Sourceforge for education (not just higher education). Hosts forums, blogs, and wikis, as well as open source software	Funded by the New Zealand Tertiary Education Commission's New Zealand Open Source Virtual Learning Environment project
Free Software Foundation (FSF)	Promotes computer users' rights to use, study, copy, modify, and redistribute computer programs	Author and advocate of the GPL open source license. Publishes a directory of free software projects	A six-member board, including Eben Moglen and Lawrence Lessig, governs the FSF, which is led by Richard Stallman. It is a nonprofit organization
Globus Alliance	Links organizations and individuals who are developing fundamental technologies behind the "Grid," which lets people share computing power, databases, instruments, and other on-line tools	Among other activities, Globus provides the Globus Toolkit, an open source software toolkit used for building Grid systems and applications. Globus also incubates new projects	Governed by the Globus Management Committee. Most funding comes from the federal government – especially the NSF and Department of Education. Control over software decisions is in the hands of "committers"

IMS Global Learning Consortium (IMS)	Supports the adoption and use of learning technology worldwide	Standards-setting body for industry and academic IT	Non-profit membership organization, supported by elite universities, software firms, publishers, and other industry players
Internet2	Works in partnership with universities, industry, and government to develop and deploy advanced network applications and technologies	In addition to providing connectivity to a fast backbone network, Internet2 funds and provides an organizational home for open source middleware and security software	Led by a board of trustees, mostly university presidents and some CIOs
Java Architectures Special Interest Group (JA-SIG)	Serves as an association for Java developers in higher education	Organizational home for uPortal and a few other open source projects	Non-profit board. Major support was provided by Sun. Funding also comes from conference fees
.LRN Consortium	Advances the adoption, improvement, and development of .LRN software; convenes a global community of innovative people and organizations in educational technology to share knowledge and applications using open source principles	A single-project focused open source organization, devoted to the .LRN application, a tool for rapidly developing web-based learning communities	Non-profit organization governed by a board of directors that consults with members
Mozilla Foundation	Promotes choice and innovation on the Internet. Provides organizational, legal, and financial support for the Mozilla open source software project, and governs the actions of Mozilla Corporation	Produces the Firefox web browser, using a centrally-directed approach that integrates the contributions of developers. Advocates for standards and partners with industry	Non-profit Mozilla Foundation owns the for-profit Mozilla Corporation. Governed by a small board that is chaired by Mitch Kapor. Funding is provided through industry partnerships and some donations
Ohio Learning Network (OLN)	Networks Ohio's colleges and universities using technology to enhance distance learning	Among other activities, OLN is developing a point-to-point shared hosting and support model for colleges and universities in Ohio. They are piloting Sakai and other open source products	A program of the Ohio Board of Regents

Open Source Applications Foundation (OSAF)	Develops next-generation inter-personal information management software	Employs full-time software engineers who are developing a desktop PIM application code-named "Chandler", and a server code-named "Cosmo"	Non-profit organization whose funding comes from Mitch Kapor, The Andrew W. Mellon Foundation, and the CSG. Governed by a small board that includes Mitchell Baker and Mitch Kapor of Mozilla
Open Source Development Labs (OSDL)	Accelerates the deployment of Linux, marshals resources from industry, and provides advice to vendors and end users on open source software	Higher Education Forum targets higher education community with the goal of advancing the use of Linux	OSDL is supported by vendors like IBM, HP, and other firms, many of whom have business models built around open source. The Higher Education Forum has attracted a handful of schools to become paying Academic Affiliates. It is led by a board of directors, most of whom come from the software industry
Open Source Initiative (OSI)	Certifies open source licenses. Publishes the widely-accepted definition of “open source”	Reviews open source licenses to determine whether they fit the approved standards. Helps reduce the proliferation of licenses. Advocates for open source software to the commercial world	Nonprofit board composed of a mix of open source leaders and industry representatives
Open Source Software Institute (OSSI)	Promotes the development and implementation of open-source software solutions in government and academia	Administers the National Center for Open Source Policy and Research. Facilitates communication among various constituent communities	Non-profit governed by a small board from industry, and with a representative of the University of Southern Mississippi, where OSSI is hosted. A larger advisory board is beneath the governing board. Three levels of membership, for corporate, government, and academic members. Major funding from HP and other software firms
OSS-Watch	Provides an information clearinghouse and source of unbiased advice about open source for UK further and higher education	Organizes conferences, publishes educational materials, monitors the open source community, and communicates advances	Supported by JISC, the British government's higher education information technology funding group. An advisory committee drawn from the education sector guides the organization

Sakai Foundation	Provide Sakai developers, adopters, and users with a common place to coordinate their efforts	The Sakai Foundation oversees the development of the Sakai software and administers the Sakai Partners Program, which collects funds from participant colleges and universities (as well as commercial affiliates) to support development	The Sakai Foundation is a non-profit membership organization whose board is elected by members. Funding provided by the Andrew W. Mellon, The William and Flora Hewlett Foundations, partner schools and commercial affiliates
Software Freedom Law Center (SFLC) and Conservancy	Offers legal advice to protect and advance free and open source software	SFLC provides selected open source projects with free legal services. These services include help with IP and licensing, litigation support, and training. The Conservancy provides a corporate entity for free and open source projects that seek an organizational home	A non-profit law firm, SFLC is supported by commercial open source vendors like IBM, HP, and Red Hat, who allow SFLC significant discretion in selecting clients. It is governed by a small board of directors drawn from the free and open source community
SourceForge	Hosts more than 100,000 projects on a vast open source software development web site	Provides free web hosting, project management, and collaboration services to open source projects. Also markets an enterprise version to support development inside corporations	For-profit owned by OSTG, Inc. Requests donations, though it is not a charity. Omidyar Network among others have provided support