

# JA-SIG Keynote

Delivered by Ira H. Fuchs on April 29, 2008

---

Thank you Jonathan for that very generous introduction and thank you to the JA-SIG Board for inviting me here today. It is a great pleasure to be here and to see firsthand the degree of interest --- I should say excitement --- that community source has generated. Those of you that have been involved with uPortal know that you have played a very significant role in the success of this movement. I was delighted to see that Educause recognized uPortal with last year's Catalyst award for being the most widely used Portal in higher education and for being the groundbreaking project that led to all of the community source projects of today. I was also thrilled to see that uPortal recently announced the availability of Version 3. You all deserve a great deal of credit for creating and maintaining a very valuable software project and you can be especially proud of everything that uPortal has spawned.

I want to talk today about some of the legal, economic, organizational, and psychological impediments to widespread acceptance of community source and what might be done to overcome them. Then, because talking about obstacles can be rather gloomy work, I will conclude with a guided tour of the current status of the community and open source software projects funded by my program at the Mellon Foundation. That tour contains more than enough good news to cheer us all up again.

As you know, community source software has begun to benefit a significant number of institutions worldwide, yet those institutions still constitute only a subculture in global higher education. Why doesn't every campus use open source for services

beyond the Internet and Linux? Why don't more institutions participate in community source development?

Please know that my goal isn't to depress you: there is a tremendous amount to be proud of about your community source projects, including the quality of the work they contain and the quantity of installations they have achieved. But as we come together here to celebrate the successes of community source, in a community of like-minded individuals, I think it's useful for us to take a moment to look realistically at some of the challenges that remain—and in particular, to take a hard look at the way our projects look from the outside. After all, if we're going to continue to grow these projects into the kinds of long-term successes that we all want them to become, we need to understand, not only why people choose to adopt community source software, but also why they don't.

Of course, before we talk about that, we might want to ask why we should care if more institutions want to adopt community source? After all, many of the communities represented here are already self-sustaining. I think we should care, for at least two sets of reasons. First, communities are living things, and like all living things, when they stop growing they start to die. Not all growth is measured in terms of adoption, any more than the growth of living things is only measured in terms of size, but for right now at least, we want to see these projects continue to attract new members, and to work together to strengthen the community source “brand.”

Second, I don't think there's any question that the network effects arising from growth create benefits for the entire community. Most of us are familiar with Eric

Raymond's statement of Linus' Law, that "given enough eyeballs, all bugs are shallow," but that's only one of many positive network effects from project growth. For instance, more usage identifies opportunities for enhancement more quickly, thereby improving the product faster. More usage also generates more and better design input. And more usage creates all kinds of synergies in the community. For example, as the community grows...

- Developers have more people around them to turn to for help.
- End-users also have more peers, which helps them directly and also reduces the support costs that their institutions must bear.
- Funders, including foundations such as the Mellon Foundation, find that their investments in the community generate larger social returns.
- Contributors find more reputational and ego benefits from contributing to the project.
- And finally, vendors are attracted in greater numbers, and find more ways to serve the community.

All these factors work together to make thriving projects more attractive to potential adopters. So growth begets growth, in a virtuous cycle that continues to spread and thereby reduce the costs and risks of software development for all of the institutions in the community. Consequently, I don't think we can rest on our laurels just yet; I think we need to encourage adoption.

What, then, are the obstacles to growth?

Much of what I'm about to say you have heard before. Often, you've heard it from people in the community who declare that these issues aren't really obstacles at all! That may be true from where we all sit, but I'm going to take a slightly different approach. As I was thinking about this talk, I spoke with a number of IT colleagues around the world whom I know are not currently involved with any community source

projects, and I asked them what the community source world looked like from their points of view. I would encourage you to take what I have to say as a description of the kinds of world-views that we have to acknowledge, take seriously and respectfully, and find ways to address if our projects are to keep growing.

Let's start with the legal obstacles. Given the recent unpleasantness with Blackboard and other IP issues that still occupy a lot of our collective attention, I was pleasantly surprised to find that most of the people with whom I spoke are not too concerned about any absolute legal obstacles to participation. But there are still challenges. At a few campuses, community source is held to a risk-assessment double-standard, in which most other types of investments are evaluated by weighing the value of participation against the risks involved, but where community source is believed by some to represent "limitless liability" on issues like IP infringement. I submit that such a position is irrational or at least inconsistent, because empirically speaking, campuses routinely take risks that are objectively far greater than any risks yet to be demonstrated from participating in open source, and some of the same insurance practices that institutions use to shield themselves from other kinds of risk can mitigate their IP risks as well. Still, if this viewpoint becomes more common, there are some ways we can help; for instance, having software foundations, such as the Sakai or Kualu Foundations, hold the IP may, at least to a degree, shield participating institutions from liability by keeping them out of the direct line of fire.

I occasionally hear discussions of how the copyright laws should be amended to give not-for-profit institutions a kind of "fair use" exemption for participation in open

source projects, so that they are not subject to the same liabilities as a commercial project that infringes software copyrights or patents. It seems pretty clear to me that such an exemption is not in the cards—and anyway, I don't think it's necessary. As I've already said, even if the present situation worsens, we can still achieve effective protections with an intelligent combination of insurance and strategic IP ownership arrangements.

The second domain I want to address is the need for professional support. Overall IT capacity in higher education is likely to decline in the future as more institutions outsource key infrastructure like email and other collaborative services. Given that trend, if there's anyone here who still believes that vendors are unimportant to the long-term success of our projects, then it's time to lose that delusion. Vendors are essential, even for big institutions and especially for smaller ones. It's always startling when I meet a senior leader at a higher education institution who is familiar with open source, and even with community source, but believes that his or her institution needs to have substantial IT capacity even to consider participating. Roughly 80% of the institutional adoptions of Mellon-funded software over the last couple of years were vendor-supported adoptions to at least some degree, and most of them were wholly vendor-supported. In some newer projects, such as Quali Financials, the figure is virtually 100%. That's a statistic that should be on everyone's mind and everyone's tongue when we're talking with leaders in our institutions or in higher education broadly. Remember, too, that some of those institutions could have installed and run the software unaided; instead, they chose to work with vendors in order to mitigate the risks

involved with going it alone. That's a strategy we can expect to see more often as time goes by.

On a related point, it may seem obvious inside the community that open source is not "free," and sometimes it seems as though the rest of the world understands this as well. But I don't think we can assume that just saying "community source software is not free" exhausts our responsibility to help others understand what the total costs of ownership for community source really are. If we just end the conversation there, we leave others free to imagine—or to let their proprietary vendors insinuate—cost scenarios that have nothing to do with the real-world costs of open source.

Our projects need to take a lesson from how the community came together to prove prior art in the Blackboard situation. We must now provide the realistic, empirical evidence of what community source software really costs during the transition phase and then later, during steady-state operations. We're a little afraid of those numbers, I suspect, because some of the earliest adopters put in huge amounts of resources to make sure the projects succeeded. But we can't be afraid of the truth, and there are enough second and third-generation adopters out there now that we should be able to show clearly that costs are competitive. Even more important, getting that evidence now will allow us to track costs in a way that can help us to make better decisions about the projects going forward. Moreover, getting hard data on the costs may help us finally move beyond the naïve, direct comparisons to get at the crucial strategic issues on which the benefits of open source are much more difficult to assail; namely, that open

source affords institutions greater strategic agility and greater control over their own destiny going forward.

Then there are the perceptual issues that community source projects still face in the outside world. Over the years, we have heard open source development caricatured as an anti-commercial activity --- some observers have gone as far as likening open source development to some sort of communist plot. Few believe these simplistic sound-bites, but outside our community people still do have a hard time with open source projects that appear to compete directly with commercial products on a one-to-one, “me too” basis. When I ask CIOs about this issue, I hear that senior campus administrators and Board members either don’t want to go into direct competition with a for-profit enterprise, or they have a sense that “higher education should be run more like a business” and that somehow the use of open source software is not ‘businesslike.’ Their leaders, and especially their Trustees, don’t realize that community source projects are built by designers and developers who are every bit as professional as the development teams in many proprietary software companies. They also don’t realize that most businesses, including of course software and technology companies like IBM, themselves rely heavily on open source software. Even when they’ve heard about the third-party-vendor business model, they may be reluctant to trust it because they have no firsthand experience with it, or because they think of it only in terms of small, startup companies and not of industry giants like IBM and Sun. Our communications, both formally on our wikis and informally in our conversations with campus leaders, need to recognize these misperceptions and educate people—but without talking down to them

or marginalizing their views. After all, if we really were just competing head-to-head with effective commercial products, they would have a good point.

There's another perspective that I want to mention here because it's close to my own heart. During my career, I've been an advocate of the view that wealthier, better-endowed higher education institutions have an obligation to help less-wealthy institutions by participating in open and community source software initiatives that benefit them as well as the community. There's no question that the mission and core values of our community source projects align well with the mission and core values of higher education. Values like innovation and competition can be shown to be enhanced by our projects, and enhanced in ways that benefit the entire community, not just a few. I think we need to keep reinforcing that message, but we also need to realize that not every institution or leader shares all of our views—and where we run into someone who doesn't, we need to have more pragmatic arguments ready as well.

Brad Wheeler once said to me, “if tomorrow I got a new Provost who wanted to know ‘why are all these people taking my paychecks and sitting in my offices but reporting to people at Cornell or Michigan and delivering software that Indiana University doesn't own?’, I can't then respond by talking about the virtues of altruism or the glories of open source: I need to have a spreadsheet that shows exactly why it is in our institutional interest to invest in this way.” I think that's exactly right—and I think that we have to work as a community to make those spreadsheets and related evidence available to those outside the community, who might, on careful examination, find it exactly in their institutional interest to come inside.



Which brings me to my final set of obstacles to open source adoption; namely, perception and marketing.

One misperception that still appears to be quite common in the world outside our projects is that people still judge the risks of open source software as higher, and the benefits as lower, than would most of us inside the community. Some of that misperception is still rooted in simple ignorance, though proprietary vendors seeking to preserve their market positions nurture some of it, too. The only way to overcome this widespread misperception is through clear, persistent communication: anyone who wants his or her institution to be able to participate in community source must be able to articulate the value of collaboration and participation, not only to peers, but to superiors in the organization, and to do it over and over again, understandably and effectively, to every potential stakeholder, until a more accurate perception settles in.

This is where marketing principles become crucial. It's usually very easy to persuade functional specialists that community source will help them do their work better, and if you look on the various project wikis and listservs you'll see lots of prose aimed at those constituencies. Much of that prose is just the record of people in the community doing their everyday work, but it can also be considered marketing material, in that it provides a newcomer with information that he or she can use to assess whether the project meets an institution's needs. That's great, but especially as the projects mature, we can't just assume that our open communication processes provide prospective entrants with the right amount or right kind of information to encourage them

to join. Projects need to make sure that their public faces stay friendly to people who aren't already deeply embedded in the community conversation.

We also need to be just as effective, if not more so, when talking with campus leaders about why community source is the right alternative. Senior administrators usually don't know and don't much care about software functionality: they care about costs, risks, and in particular the organizational relationships that may be affected by community source participation. When they come to the project wiki, they're looking for very different types of information than are functional specialists—and I have to say, on too many of our project sites right now they won't find a great deal of information to answer their most pressing questions. I will be much more confident about the long-term growth of all of our projects when I can go to any project wiki and see sections dedicated to providing adequate information for senior executives wanting to learn more about the issues they care most about.

Last year at the Mellon RIT program retreat, the leaders of your projects had a serious and constructive conversation on the importance of marketing to the future of open and community source. The consensus was that we needed to do more, and do it better. Talking with those outside the community, it appears we still have a long way to go: the inadequacy of marketing materials and marketing efforts, understood broadly, is one of the most commonly cited reasons why community source has not penetrated more deeply into higher education so far. For the most part, our products still lack the spit-and-polish of commercial alternatives; some lag by wider margins than others, but all need better packaging and marketing materials. It concerns me that, for all that we

talk about the importance of the user experience and the value of things like demos and screencasts, most of our projects are still intimidating at first look, even for domain experts in the fields they serve.

We could, and some say we should, leave that spit-and-polish work to the commercial vendors. As important to our success as I think our vendor community is, marketing is too important to relegate it only to the vendors. Marketing needs to be a core part of every project's work. In a few minutes, I'm going to show you Zotero; a project that I think most of us would agree has handled its marketing responsibilities very well. Some might say that Zotero is an end-user project, and that marketing Sakai or Kuali is different. Of course they're different, but even enterprise projects can still learn important lessons from Zotero. One of the most important things Zotero teaches us is the role of end-users in driving institutional adoptions. Zotero is a quintessential end-user tool right now: it's a Firefox plug-in. But its popularity with end-users is so great that more than 80 institutions have adopted Zotero officially in the last year alone, based entirely on demand from within their own campuses. How many community source projects had 80 new enterprise adoptions last year? The demos and screencasts and podcasts that Zotero has built all help to empower end-users to represent their needs more effectively to their own campus leaders, allowing would-be users to do the selling for the project and to overcome any campus objections or resistance themselves, using the knowledge and information that the marketing materials convey.

What's particularly frustrating about the current state of marketing in community source is that we seem to be stuck thinking inside the box: most of us don't know much about marketing, and don't spend much time with people who do, so when it turns out to be harder and more tedious than it looks to make a good podcast or demo, much less to plan a whole marketing campaign, we just throw up our hands, declare that we don't have time to worry about it, and hand it to the vendors. Imagine if you took the same attitude toward something like debugging or security testing?

Let's come at the problem from a different direction. When we think about institutional resources to contribute to community source software projects, we tend to think in terms of developers, designers, and usability specialists. But most of you work in higher education institutions whose non-technical resources dwarf your technical capacities. Why shouldn't an institution contribute, say, a team of marketing grad students from the Business or Communications School, headed by a professor and perhaps advised by a board assembled from the community's institutions and vendors? Wouldn't it be reasonable to trade some faculty and student time, and perhaps some academic credit, for the opportunity to help shape the brand of a global, multi-lingual, innovative technology product with broad-spectrum corporate involvement? Shouldn't the community acknowledge and reward those contributions as just as valuable—and perhaps even more so—than the contribution of one additional developer to an already large team? Arrangements like these would take some thought and work, and the teams would need to gain the trust and approval of the community, along with its financial support, for any marketing campaigns they develop. But done properly, they could be

yet another example of how community source software generates win-win solutions for higher education.

I'm going to say a little more about this before I close, but for now, let's change the topic and look at some of the new and upcoming community source projects. Because of who and where we are, I'm not going to talk about the Mellon-supported projects represented here at this meeting, in order to allow more time to introduce to you some projects that you may not know much, or anything, about. Even among the projects that aren't featured at this conference, I'm going to focus primarily on the newer and less well-known. When I'm done, I hope you'll have a good understanding of the scope of the projects with which we're currently engaged, and some sense as well of what we see as the growth areas over the next few years.

This is a map of the projects that we collectively call "RIT-space." Don't worry if you can't read the map from where you're sitting, because I'm going to go through it piece-by-piece, and we also make it available on my program's website. Let me start with an overview of the map itself, which is produced using a software package called the Visual Understanding Environment, or VUE, developed with Mellon funding by David Kahle and colleagues at Tufts University. As you'll see shortly, VUE has some very interesting capabilities beyond concept and network mapping.

One of the most useful VUE features is that the software allows you to draw pathways through the network. Today, I'm going to use a pathway I created just for this talk, to highlight a few projects of particular interest. These pathways can serve as linear narratives within the complex map, suitable for use as a replacement for PowerPoint in

many teaching/learning contexts. You can create as many pathways as you like through the map; for instance, you could create a pathway to trace the life cycle of a research project through the various pieces of software that could help manage it, then create a different pathway to do the same thing for the life cycle of course content. You can also use the map and the pathways together, as I'm going to do here: starting with a map view, then using a pathway for a PowerPoint-like presentation, but bouncing back out to the map, to other applications, and even onto the Web at several points in order to present the information I want to share.

Not all of the Mellon-funded projects represented as nodes have been directly funded by my program, because some projects, like JSTOR and ARTstor, either predate the RIT program or they were funded separately, and others, such as FEDORA and ORE, were funded by my colleagues in other Mellon programs. Those projects are marked in gray, at the far right of the map. All of the colored nodes, plus Fluid, which is the white node slightly Southwest of the center of the map, are RIT-funded.

As you may or may not be able to see, the cluster of pale yellow nodes at the bottom-left are the Kuali projects plus uPortal and Sakai (which is in pink); in other words, the administrative and academic-administrative projects that were among the first projects funded by my program after its creation in the year 2000.

This purple node represents the OpenCollection museum project, which was funded by our Board in December to enhance and convert to community source a collections management system designed for museums and library special collections.

OpenCollection is the first stop on the pathway of new projects that I've built for the talk today.

OpenCollection is distinguished from most museum systems, which tend to be optimized for particular types of museums such as art museums or natural history museums, by its ability to handle any sort of object or metadata you can imagine, which may help to explain why it has already generated a strong multinational community comprised of large and small museums of many types, libraries, higher education institutions, and cultural heritage organizations, with core participants drawn from Canada, the UK, the Netherlands, Belgium, Germany, and New Zealand as well as the US. Under the Mellon grant, the Museum of the Moving Image in New York, in partnership with Toronto and Berkeley, will conduct a community design process as well as a build project. The objective of the grant is to help rebuild OpenCollection using SOA, expand its functionality to match that of commercial museum information systems, and develop a sustaining community. The first round of OpenCollection design workshops, to be held in Toronto and New York City, is already underway.

Just like PowerPoint, all I have to do is press a button or click a mouse to visit the next stop on the pathway. In this case, our next stop is VUE itself, the software I'm using to make this presentation. I've already told you something about VUE, and you're seeing it in action, so I will only stop here long enough to say that we have just funded another round of development for VUE, during which a number of exciting new capabilities will be added. For one, VUE will become Web-based; for another, it will be refactored so that its network visualization features can be used as display tools for a

variety of other projects, including the SEASR project, which I'll say more about in a moment.

Along with VUE, we currently fund another authoring tool, called Sophie. Sophie is an ebook authoring environment, built as a collaboration between USC and the Institute for the Future of the Book. What most distinguishes Sophie from commercial authoring environments costing tens of thousands of dollars is not its functionality, or even its price—like all Mellon-funded projects, it's open source—but rather its ease of use. With Sophie, a faculty member can receive very brief training and quickly learn to produce rich media content for courses and publications that is similar in quality to professional publications built using animation tools like Flash and typesetting programs like PageMaker or QuarkXpress. These rich-media documents can embed images, sounds, and even films, using automation techniques that make it easy to insert soundtracks, automate sections of a page or of the document, and so on. Again, what's remarkable is not that you can combine video and text, but that someone with minimal training can do it. I'm going to bring up Sophie now, taking advantage of the fact that VUE automatically embeds a URL into the pathway for any node that is attached to a Web resource.

Here's an example of a Sophie e-book in action. A high school teacher in New York City named Sol Gaitan created this book for her AP Spanish class. I'm only showing you an excerpt from the book, just to keep it simple. The whole book is 104 pages long, and Sol reports that creating it took about three weeks of working an hour or so every other day. That works out to about ten pages per hour—pretty impressive



for such a richly multimedia document! It's a book about an Andalusian poet named Federico García Lorca, and it combines text, images, audio, and video; there's even an Appendix that includes music inspired by García Lorca's poetry. In Sol's class, students first use this book as a text during the semester, and then, as a final project, create Sophie books of their own, drawing on work from García Lorca and other Spanish or Latin American authors and artists of their choice.

Sophie's commenting feature is particularly interesting because it's Web-aware. Your comments can just be regular annotations and marginal notes—or, if you connect your book to a Sophie Server, then your comments can be shared, real-time, with anyone else reading the book on the same server from anywhere in the world. Sophie's creator, Bob Stein, has a vision for the next generation of books that includes making real the metaphor of scholarship as a conversation. He's actually brought the conversation right down inside the book!

As you can see here, book pages can include videos, which can be set up to play automatically when you land on a page, or manually. They can also include audio soundtracks, which can run in a window or in the background, using something called a Timeline. I'm going to show you how easy it is to build rich media in Sophie by replacing the audio soundtrack for this book. First, here's how it works now: when you get past the front-matter and reach the first content page, a soundtrack plays and the next several pages auto-turn for you. I'm going to replace the soundtrack with a new one, which is just a matter of dragging a new file in, and resizing it on the timeline to tell Sophie how many pages I want to connect it to.

OK, now let's bounce back to VUE and continue our tour of the map.

The third scholarly tool project I want to show you is Zotero. You've already heard me praise Zotero as an example of how to do marketing right. In fact, Zotero does many things right, or else the marketing alone wouldn't help much. As many of you know, it's a Firefox extension that lets scholars store and manage web content and academic citations. Since its release in late 2006, Zotero has achieved more than three-quarters of a million active users: it's growing at a rate of about 70,000 new users per month, so it should top one million users in the next few months, and it was named one of the Best Free Software packages in both 2007 and 2008 by *PC Magazine*.

Some of you who pay attention to Zotero may also know that we have funded the Center for History and New Media at George Mason University, Zotero's creators, to develop Zotero Server, which will allow scholars and students to share citations and other Zotero data within and across institutions, with the whole world or with selected groups. And what many of you may not know is that, last December, our Board funded a very exciting collaboration between Zotero and the Internet Archive to try to solve the problem of the lack of permanence of scholarly resources on the Web. When this project is finished, you will be able to push a button in Zotero and have the Internet Archive automatically archive a copy of any static resource you're looking at—Web or personal. IA will store a copy and return to you a permanent URI for that item, so that you can cite it, share it with others, or do anything else you like while knowing that the object will be available permanently at that address. IA will also, at your option, make the object available through its own search interface for use by others.

Zotero is successful in part because it encourages innovation by others. Zotero is a plug-in, but it also has its own APIs so you can build plug-ins within Zotero. Here's an example of one plug-in, for videos, called Vertov, which extends Zotero to deal effectively with video content, a medium that Zotero doesn't recognize natively except as a blob. With Vertov, you can view and playback video inside Zotero, and you can also clip and annotate video. Combine that with Zotero's Internet Archive connections, and you now have a way to clip, permanently archive, and share and cite video clips. I'm going to demonstrate it using a clip from the EDUCAUSE award to uPortal. Suppose I don't want to cite the whole video, I only want to cite certain of the interviews? Here's how I do it.

The next project I want to discuss is SEASR: that's an acronym for the Software Environment for the Advancement of Scholarly Research. The project is the result of collaboration between IBM and the NCSA at UIUC, and it's a platform for rich-media analytics. If you perform research using quantitative data like statistics or economic modeling, there's a terrific open source environment called the R project that lets you analyze and visualize your data any way you like. But the R project isn't very useful in most arts and humanities disciplines, because instead of spreadsheets of numbers those disciplines use rich media, like text, images, audio, and video. SEASR is an environment that's intended to provide the same quality and variety of analysis and visualization for rich media that the R project does for quant data.

That's a tall order, given that the R project has thousands of collaborators and has been in existence for more than a decade. It's still early days for SEASR, which was

funded last year, will ship its first production version this summer, and is just holding its first training sessions for institutional installers and prospective scholars next month in Champaign. Still, SEASR already provides a component environment that integrates a powerful, flexible data model with extremely user-friendly features like automatic marshalling of Grid resources, a palette-based scientific workflow system, and Web services interfaces that let it connect to and from other projects easily. The existing SEASR components are particularly strong in the areas of large-corpus text mining and music information retrieval; in fact, here's a demo called "Son of Blinky," that shows how SEASR can use an assemblage of its machine learning components to automatically categorize a piece of music by genre and mood using any of several different classification algorithms.

Son of Blinky is essentially a head-to-head competition between several music classifiers, some of which look at the music genre—rock, classical, pop, and so on—and others of which look at mood: cheerful, wistful, and so on. It's designed to help computational musicologists improve their classification algorithms, so it's set up as a research tool—but you can easily imagine turning it into a more practical tool by adding a new component that polls the various classifiers and makes a summary judgment about the genre and mood, and then using that summary tool to auto-generate tags or metadata for a large music or video collection.

At the heart of SEASR is a user interface connected to a workflow engine, sitting on top of a very flexible data model and a services oriented architecture. The user interface uses a wiring diagram metaphor, so you can build sophisticated analyses by

stringing together building-block components. Each building-block is an algorithm: one component may open a data file for reading, while another strips endings off of nouns, another performs latent semantic analysis, and another visualizes the result. You can reassemble these algorithms into many different combinations depending on the kinds of questions you want to ask. SEASR takes care of all the nitty-gritty work, like making connections to the data, finding CPUs and allocating memory, and so on, using both institutional and Grid resources as needed.

Over the next few years, we'll most likely keep funding new components and new capabilities into SEASR, in the hope that it will attract the kind of critical mass that has made the R-project so successful in its own field. SEASR has already drawn the interest of US and foreign governmental funders, so we hope before long to see modules showing up in SEASR that were not necessarily supported directly by Mellon.

The last project I want to talk about today is Bamboo, which was only approved for funding a few weeks ago at our last Board meeting. Describing Bamboo is actually difficult right now, because the project will begin with an extended period of self-definition. Bamboo hopes to develop a community to provide shared technology services to support arts and humanities scholarship. There are several ways to understand that goal, and for the time being Bamboo is deliberately avoiding choosing any one of them, in favor of spending the next 18 months in a community design process that is intended to discover what services will be of greatest value to artists and humanists and developing a plan for delivering the most widely and urgently needed services first.

One way to understand Bamboo is as an arts and humanities analog to the cyberinfrastructure initiatives that have become commonplace in the sciences. In that guise, Bamboo has attracted a lot of interest from government funding agencies here and abroad, who see the humanities and arts as a large and important gap in their current cyberinfrastructure funding. By providing such an environment in a community source model, Bamboo could make cyberinfrastructure substantially more affordable by more higher education institutions, allowing smaller and less wealthy universities and colleges to set up virtual research organizations more easily and cheaply. Because it will be built as infrastructure, rather than as a sideline to a research project, we would hope that Bamboo will deliver cyberinfrastructure in a truly sustainable form. Along the way, Bamboo would almost certainly build strong ties into the Sakai, SEASR, FEDORA, and Quali Research Administration communities.

Another way to understand Bamboo is as an institutional project—a classic community source project—that hopes to solve the problem of delivering technology support consistently and affordably to the arts and humanities on large and small higher education campuses everywhere. As many of you know, right now most “digital humanities scholarship” is accomplished by means of custom solutions: technologies that seldom cumulate beyond the borders of the project for which they’re developed, and that usually dry up and blow away as soon as either scholarly interest or money runs out. By creating an SOA environment for scholarly support, Bamboo could reduce the costs of new custom solutions considerably, make it easier to connect them together for cumulative benefits across projects, and make it more affordable to sustain them

after the initial grant runs out. In this way, Bamboo could end up being a publishing environment for scholarly content and scholarly research technology as well. I can also imagine Bamboo building strong ties to Kuali Student, both to borrow SOA insights and technology and to connect student learning to scholarship in new and innovative ways.

Bamboo may also be viewed as a technology environment that will provide end-to-end support for managing the life cycle of scholarly content, from collaboration through publication through archiving. Here, Bamboo would connect with Mellon-funded projects like MIT's SIMILE and the FEDORA repository, as well as with SEASR, Zotero, VUE, and Sophie, and perhaps with upcoming library systems projects as well. Some of the work my colleagues at Mellon have been funding on standards-based annotation systems would likely also come into play here.

If Bamboo is successful, then in the long term it could become one of the more important nexi in the higher education technology network. The first round of Bamboo workshops is underway today, in Berkeley, and three more offerings of that initial workshop are going to be held over the next several weeks in Chicago, Paris, and Princeton. See [www.projectbamboo.org](http://www.projectbamboo.org) if you would like more details

That concludes my tour of the Mellon/RIT projects, but before I take questions, I want to return to the idea of why growth is important, and why we need to be realistic about the various obstacles that I mentioned earlier and work hard to overcome them. We can talk about broad categories of legal, economic, organizational and other obstacles facing our projects, but the reality is that each institution is different, and on each campus the issues will come up in slightly different forms and with slightly different

emphases. To succeed in those diverse environments, *all* of us must become at least basically competent at educating and marketing these projects as well as developing the software.

At my program's grantee retreat this year, Clifford Lynch gave his usual terrific closing summation. Among the many things he said that I found valuable, one observation seems particularly apt. Clifford noted that until recently, every community source project served an established purpose like financials or learning management. These established niches have associated budget lines at most institutions, so obtaining resources and institutional commitment was in a sense straightforward: institutions merely needed to reallocate resources already reserved for that purpose.

However, with the arrivals of some of the new projects that I've just shown you, Clifford commented that community source has entered a new phase, in which new projects are attempting to deliver products that have no direct commercial analogues, and for which no institution has yet carved out a budget line. If you thought the conversations with college leaders about a community source financial or learning management system were exhausting, wait until you try to explain both a whole new software concept *and* the idea of community source in the same conversation. That's the challenge faced by projects like Fluid and the just-funded Bamboo, and that's when we'll really need the marketing materials, the clear understandings of how leaders think strategically about software, and the communications skills that I've been talking about today.



The time to develop and demonstrate marketing and communication skills is now, on our existing initiatives, so that the new crop of projects will have models for how to engage effectively and constructively with higher education leaders and the community. Every one of you has a stake in the prosperity of these new projects as well as your own. That stake is easy to see for something like Fluid, which touches so many projects: if you want Fluid to continue to help you make your own projects easier to use and more valuable to your institutions, then you need to help Fluid create a sustaining community of its own, too. That kind of synergy is just as important for Bamboo and SEASR. These projects each bring new parts of the higher education establishment into the community source sphere, and by doing so they create whole new constituencies that will be predisposed to support your projects as well. This conference is a living example of the kinds of synergies that can develop when we all remember that there is a meta-community of community source projects, and that it is in our enlightened self-interest to help all of our meta-community members succeed.

To succeed in growing these projects, you're going to have to talk effectively with campus leaders as well as functional specialists about your own project and perhaps others as well. Don't be daunted by the idea of educating your Provost or even your Trustees. I would suggest that you keep two simple principles in mind. The first is that (in spite of any doubts you may occasionally harbor) your campus leaders are fundamentally rational people, who approach any question like 'should we go with a community source project?' using a very simple calculus. They take the benefits, and

subtract the costs and the risks, all expressed as numbers. If the result is positive, they're usually agreeable to moving ahead.

In other words, assuming that in fact your case is good, your primary responsibility in a conversation with your leaders is to give them numbers they can trust. Imagine for a moment that you're in a meeting with your Provost and your general counsel, and they want to know the economic benefits and costs of joining a community source project. If you are not able to communicate both the value of your institution's participation in a community source project and the risks of participation, using real numbers where possible and justifying your assumptions where real numbers aren't available, then you should not be surprised if your Provost does the rational thing and decides to steer clear of such a venture.

The second principle involves *how* to give your leadership numbers that they can trust. When educating your leaders, you must assume that their concerns are different from your concerns, and you must learn to see the world from their perspective, not try to force them to see it from yours. Talking about SOA or agile programming with them isn't just unnecessary; it's also usually a mistake, because while you're telling them about the latest tech wonder, you're also implicitly communicating to them that you don't understand what *they* care about. That damages trust, so they're less likely to believe you even when you finally deliver the information they actually want. Instead, learn what matters to your leadership, learn to see the world their way, and then provide them information in a form they can trust because it is understandable to them and meets their needs.

If you happen to be at an institution where the CIO is not a fan of community source, the same rules apply: learn to see the world through his or her eyes, find out what's important from that perspective, and assemble the information your CIO feels he or she needs to reach the decision that's right for your institution. If, as is increasingly the case, your CIO has just come over from the corporate sector and thinks higher education should be run more like a business, then use project case-studies and other materials to show her why joining a community source project is a great business decision. If he's risk-averse, learn from Ted Dodds and Kuali Student how to explain why community source is a lower-risk approach than relying on a proprietary vendor *or* going it alone to build a home-grown solution. If she's concerned about vendor support, prepare a list of vendors already engaged with the project and show her what a purchasing requisition for open-source services looks like. In short, meet your leaders where they are, not where you wish they were.

It's really that simple, but like many simple things, it's not necessarily easy; in fact, most of the people in this room have seen firsthand how challenging it can be. Still, there's every reason to be encouraged. The projects are thriving, the communities are vibrant—if anyone still doubts, a visit to this meeting should set their mind at ease. We may have picked some of the low-hanging fruit in terms of institutional adoptions, so finding new institutions to join us in the next wave may be more challenging. But you have the people and the tools you need to meet those challenges, provided you approach them in the same spirit you've approached all the challenges so far: together,

with a commitment to doing what needs to be done at a very high level of quality and diligence. Thanks very much for your time and attention.