

GARY SCHWARTZ - FROM THE OTHER SIDE OF THE COUNTER – LEADING A UNIVERSITY OPEN SOURCE PROJECT*

Ken Udas

This work is produced by OpenStax-CNX and licensed under the Creative Commons Attribution License 3.0[†]

Abstract

Gary Schwartz's contribution to the OSS and OER in Education Series. In this post, he writes from the perspective of an open source project manager.

NOTE: Author - Gary Schwartz, "From the Other Side of the Counter, Leading a University Open Source Project". Originally submitted October 17th, 2007 to the OSS and OER in Education Series, Terra Incognita blog (Penn State World Campus), edited by Ken Udas.

Like Forest Gump who found himself a shrimp boat captain, we find ourselves leaders of an open source software (OSS) project. It happens.

Our open source project is Bedework (pronounce it as you would beadwork), an open-source, enterprise calendar system for higher education designed to conform to current calendaring standards. The "we" are the Communications & Middleware Technologies unit at Rensselaer Polytechnic Institute of which I am the director.

Unlike some other contributors to this series, I am not a deep thinker on the topic of open source. Reviewing material for this posting, I came across a document I wrote four years ago for my management justifying our participation in the University of Washington's UWCalendar project.

"Whereas many university people enjoy a spiritual affinity for open source software, our interest is more pragmatic. As a campus-wide development and support group, technologies and products which have no license or usage fees are critical to providing solutions which can be deployed and reconfigured with impunity. Our web development foundation is largely built atop products and technologies which have no usage fees whatsoever, allowing us to deploy as many instances, servers, CPU's, etc as might prove to be necessary over time."

Recognizing that I would feel more comfortable if I had only one foot firmly wedged in my mouth, I continued,

"We are anxious to contribute to the project (UWCalendar) because:

*Version 1.4: Mar 7, 2009 6:21 am +0000

[†]<http://creativecommons.org/licenses/by/3.0/>

1. *We feel our work will make the product more attractive to other universities, hopefully resulting in many more of them using and developing this software.*
2. *The University of Washington has done most of the work which we have benefited from. Reciprocating is the right thing to do.*
3. *Rensselaer relies heavily on and benefits mightily from open source software but seldom contributes to open source. We believe this contribution will enhance Rensselaer's reputation in the area of software development."*

Our four year foray into the world of open source, two years working with the University of Washington, and the last two as leaders of the Bedework project, have had a profound impact on my views about open source. I agree with much of what Pat Masson and Rob Abel have said in this series. I have come to appreciate the message of the Mellon Foundation's Chris Mackie ¹ on Cyber Infrastructure sustainability as well as the "fallacy of the field of dreams."

The perspective I have to share on open source software in higher education is that of trying to build a modest open source project to sustainability. In the process, we have learned a lot about ourselves and our own university.

I have struggled somewhat to find the right voice for this piece as it is intimately tied to our experience with the open source project we lead—Bedework. Whereas one of the lessons of managing a fledgling open source project is "always be closing," that is, trying to sell your project, bowdlerizing the content to remove all references to Bedework eclipsed my skill as writer.

1 The Back Story

Some years ago, our CIO tasked my unit to provide a public events calendar for our university. Although there were a number of calendaring/scheduling systems on campus, public events were announced and managed through e-mail, web pages, and print publications. There was no explicit budget for this project, so buying a commercial product was not a viable option. Our choices were to write it ourselves, use software already produced by someone else, or collaborate with other organizations to produce this software. We expressed the objective this way:

"The software should be used and developed by multiple universities. There are three dominant products in university calendaring today including homegrown. Many institutions of higher education have chosen to implement their own calendar systems, some of which are very fine. Unfortunately, as far as we know, no two schools use, or collaboratively develop, the same calendar software. Rensselaer is interested in contributing to a university-specific calendaring product but we already have too many projects chasing too few people. We would prefer to have circumscribed, intermittent calendar development projects rather than having continuous development and support duties. An open source project potentially allows us to meet these objectives."

We continued to enumerate the following requirements:

1. Implementation is consonant with our core competencies in Java/J2EE programming, XML, and web interface design and construction.
2. Open source – no license or usage fees
3. The ability to distribute administration and control to the event owners themselves is crucial in a university environment.

¹http://www.si.umich.edu/cyber-infrastructure/bio_mackie.htm

4. The code must provide complete, well-defined APIs which are scrupulously honored, with no local dependencies (authentication, policies, etc.) The packaging must allow competent professionals to easily install the package and to get a demo version running with minimal confusion and frustration.

(With respect to the last point, it is clear, looking back, that high standards are not especially useful unless you can hold others to them.)

RPI took a look at the University of Washington's UWCalendar, whose mission statement, says, in part,

“UW Calendar will be a total calendaring and events system for institutions of higher learning. . . . UW Calendar will be open source and platform independent. It will use existing open standards. It will support integration with other systems and middleware, . . . such as uPortal and Shibboleth. It will be modular. . . and extensible . . .”

As the University of Washington's goals were consonant with RPI's, RPI joined the UWCalendar development team in June 2003. RPI's initial motivation was to deliver value locally to the RPI community while at the same time making UWCalendar attractive enough to other universities that they would adopt the software and contribute to its development. RPI had hoped that UWCalendar would eventually have a substantial user and developer community within higher education.

Rensselaer's initial efforts focused on restructuring some of the code to more cleanly separate the server (back-end) part from the web client (front end). This allowed us, among other things, to easily provide a “skins” capability. The UW calendar became more modular and amenable to using other client interfaces that other developers might care to build. Two of our goals going into the project were to leverage our expertise in Java, J2EE, web client interfaces, and to avoid becoming calendar experts, leaving that role to the University of Washington developers.

In December 2003, UWCalendar was made available at RPI. Over the next 18 months, we played an increasingly large role in UWCalendar development. The University of Washington really developed two versions of UWCalendar, the open source version which we collaborated on, and a local version, based on the open source version, which integrated with their locally developed portal, providing significant value to the UW community. Their obligations to the local UW version made it increasingly difficult for them to contribute to the open source version.

In 2005 we became convinced that UWCalendar would not achieve its ambitious goals and began development of a rearchitected, hibernate-based successor. After much soul searching, in September 2005, we told our colleagues that we would be working on a new version, and we announced a preview release of Bedework in December 2005, making us leaders of a new open source project. The first production version of Bedework, version 3.0, was released in March 2006.

Bedework's design goals and capabilities include platform independence (via Java/J2EE), database independence (via hibernate), internationalization, standards (RFC 2445, CalDAV) compliance, portlet (JSR168) support, no license fees or restrictions (BSD style open source license) fine-grained distributed administration, support for public events, personal calendars, and departmental calendars, easy to install code with complete, well-defined APIs, no local dependencies, support for external authentication (such as LDAP, Yale CAS, etc) via container authentication, full access control via the CalDAV model, XML and XSLT based web clients allowing for a number of capabilities, such as localization and multilanguage support and RSS syndication.

Bedework is probably in production use or in some stage of production deployment at about two dozen institutions of higher education.

2 IP – ours and others

As an independent open source project, we needed to decide early on how to handle the intellectual property issues associated with Bedework. The two pressing questions to be decided were the terms and conditions of the Bedework license, and the terms and conditions of the Bedework contributor's agreements.

Although the case could be made that Bedework was only the logical heir to UWCalendar and not a derivative product, we weren't sure what it meant to make such a case, or how much work it would prove to be to make such a case. Consequently, we decided to pretty much adopt and adapt the terms of UWCalendar, allowing the Bedework source code to be used for any purposes, including commercially, as long as acknowledge is given. Having to choose from the large number of open source licensing terms was not an appealing prospect anyhow.

When we were initially considering contributing to UWCalendar, we bridled at the notion of allowing anyone to make money from our work. This was clearly not a well-reasoned response as no one was exploiting UWCalendar commercially, or had shown any interest in doing so. We discussed the issue with the UW developers and they told us it was unlikely that their university had the resources or interest in policing a more restrictive license. Over time we have come to appreciate that the license needs to serve as an enabler to adoption, and that commercial adoption was perhaps a sign of success, not something to be feared.

The contributor's agreement is interesting with respect to the renewed interest in higher ed in exploiting their intellectual property commercially, and in protecting their IP. Specifics aside, it has become increasingly difficult at some universities to sign contributor's agreements in the wake of this very protective approach to IP. We would likely have more difficulty today signing the same contributor's agreement we signed four years ago.

We have received signed agreements from somewhere between six and twelve organizations, however.

3 Open Standards

Standards compliance is the key to Bedework's success - present and future. However, standards compliance is a double-edged, possibly triple-edged, sword.

In the name of standards compliance, there are potentially useful features we have not implemented because they would not be standards-compliant and would impede interoperability. Sometimes we simply have not brought enough ingenuity to bear on the problem, but in other instances there does not appear to be a way to have our standards cake and eat it too. And sometimes, we discover that we are not purer than Caesar's wife, and we are not quite as standards compliant as we have advertised.

Standards evolve and new standards come into existence. In our relatively brief history as calendar developers, the IETF began work on RFC2245-bis, an update to RFC2445, "Internet Calendaring and Scheduling Core Object Specification (iCalendar)," and published RFC4791 "Calendaring Extensions to WebDAV (CalDAV)," all requiring changes to our source code.

In his earlier posting, Rob Abel posited that "... Standards organizations are pretty much the only way to get a level playing field when it comes to new open source applications for learning - however, that won't happen unless the open source projects/communities are active participants." We are active members of CalConnect, the Calendaring & Scheduling Consortium, as are Mozilla, the Open Software Applications Foundation (OSAF), the Open Connector project, as well as about 20 research universities, commercial vendors and other companies. Although CalConnect is not a standards setting body itself, much of its work is devoted to standards development and interoperability testing. Active participation by both the open sourced developers and academia in these processes has benefitted both these communities and the resulting standards.

4 Building community, contributors = sustainability

Our open source leadership is still evolving, with room for improvement. We have incorporated contributions from some, and from others we have contributions which we have not yet incorporated, something we need to address.

In Scott Rosenberg's "Dreaming in Code," Rosenberg says that in Eric Raymond's "The Cathedral and the Bazaar," Raymond identified two key prerequisites ... and the rise of a cooperative ethos built around a leadership style like Torvald's that encouraged newcomers, welcomed contributions, and strove to maximize the number of qualified participants." Whereas Linux has a place in the open source pantheon that Bedework

will never assume, the ideal of the “cooperative ethos” described above seems to be worth striving for. As I said, we have much to learn in this regard.

We judge Bedework’s success not by whether it is the best calendaring product, whatever that might mean in a given context, but whether viable and growing user and developer communities within higher ed establish themselves. Both Bedework communities are growing but they have not achieved critical mass.

From the outset, we intended to develop Bedework with no RPI-isms or RPI branding. The name bears no relationship to our institution, nor does the code have any special awareness or consideration for the computing infrastructure at RPI. If we had not these objectives in mind from the beginning, I think it would have been very difficult to “sanitize” the code and/or design at some later time.

We recognized early on the world may not beat a path to your door if you build a better (or perhaps “good”) mousetrap. We have invited and hosted developers from other universities deploying Bedework, or thinking about deploying Bedework, and conversely we also sometimes invite ourselves to other universities to speak with them about Bedework. We also make ourselves available for consultation via telephone and e-mail. As there is no marketing, administrative, or support staff, the core development team assumes these tasks as well. For any number of obvious reasons, this is not really a very sustainable model long term, but I think it continues to be an important strategy now.

However, some very important signs of sustainable community are becoming evident. Users on the mailing list are starting to answer questions posed by other users, and others have developed, and shared back with us, solutions for earlier Bedework issues such as Oracle compatibility.

When and how to migrate to broader, more inclusive form of governance of the project is a question we will undoubtedly need to address sometime in the next twelve months. As the number of adopters has grown, the Bedework roadmap has become more explicitly influenced by the explicitly stated requirements of this growing community.

5 Staying on the right side of Dilbert

Although it is sometimes easy for those of us in academia to sometimes speak derisively of commercially produced software, over time any even modestly successful open source software project will be judged by the same standards as commercial software.

Despite our best efforts, we have missed almost every release deadline we have set for ourselves. In our December 2005 announcement of first preview release of Bedework 3.0, we stated the official release would be the next month, but in fact the official Bedework 3.0 was actually four months later, not the one month promised. We have subsequently improved our release performance, but vacations, illness, unanticipated local exigencies, difficulty choosing and honoring “freeze” points, and bugs found during final testing still contribute to missed release dates.

We do periodic Google searches on “Bedework” to ascertain who is saying what about Bedework and to learn who might be using Bedework (more on this point later). Among the things we have discovered is that we have been at least once accused of promoting “vaporware” and that Bedework was “primitive – just a fancy events calendar.” More gently, we were told, “I’d like to take that time to share some features that are a little clunky that you might want to examine for future upgrades.”

Undoubtedly there is a modicum of truth in most of the criticism we receive, sometimes more than a modicum, but as we view our open source work as the confluence of enlightened self-interest and altruism, it still stings.

As Bedework is open source with no licensing fees, we found we do not have a reliable way of ascertaining who is using Bedework and how they are using Bedework. We have been surprised more than once when a Google search revealed a production installation of Bedework that we knew nothing about. We are aware of those who are active on our mailing list or who contact us off the lists, but at this early stage it would be useful in a number of ways to better understand how large the Bedework community is.

We have been invited to respond to RFPs by more than one university. We certainly did not anticipate this, nor were we especially well prepared to respond as we have no marketing, sales or other nontechnical

staff. We learned what you might have already guessed, that responding to an RFP is more enjoyable as preparing an RFP, but perhaps not a whole lot more enjoyable.

In the early 1980's, researchers at UCLA developed LOCUS, a distributed operating system,

“...that provided a very high degree of network transparency while at the same time supporting high performance and automatic replication of storage. By network transparency we mean that at the system call interface there is no need to mention anything network related. Knowledge of the network and code to interact with foreign sites is below this interface and is thus hidden from both users and programs under normal conditions.”

By the end of that decade, IBM had productized much of LOCUS in their AIX PS/2.

Bedework is not a descendant of LOCUS or AIX PS/2, but Bedework's alleged agnosticisms, DBMS, application server, authentication, internalization, portal (JSR-168), presentation, standards compliance, and scalability, remind me of LOCUS' attempt at true network transparency.

Like Virginia Lee Burton's Mike Mulligan, who had always said that Mary Anne, his steam shovel, “...could dig as much in a day as a hundred men could dig in a week but he had never quite sure this was true,” we had not been quite sure that our claims of Bedework's agnosticisms were as true as we intended. Over the last 18 months, the Bedework community have helped us understand where some of these objectives had not been fully realized, and in some cases, have worked with us to make the claims “more true.”

6 Higher Ed aware

We now refer to Bedework as “a calendar system for higher education” rather than as an “institutional calendar,” so it no longer sounds like it is a product for correctional facilities.

Emphasis on higher ed does not preclude other uses for Bedework, but it does mean we are cognizant of the needs and constraints of higher ed. Bedework has no licensing fees or other costs, no restrictions on usage or deployment, distributed, fine-grained administration, standards compliance, a public events component, JSR168 portal “friendliness,” and flexible authentication and access control, and the working assumption that Bedework be one of many different calendaring systems on campus.

On the other hand, there are other higher ed needs that Bedework does not yet easily accommodate, such as displaying building and facilities hours, or scheduling faculty office hours. Serge Goldstein at Princeton has written a very sophisticated office hours application that helped me appreciate the complexities and intricacies of addressing this issue.

7 We're only in it for ... the money?

We have gotten deeply involved, much more deeply, in Bedework than we anticipated when we started collaborating with Washington more than four years ago. However, our overall focus remains delivering value locally (to the RPI community) while at the same time making Bedework attractive enough to other universities that they would adopt the software and contribute to its development.

Earlier I stated that we view our open software work as the confluence of enlightened self-interest and altruism. The self-interest was to provide our university with a public events calendaring system, which we have done. Perhaps it was all enlightened self interest, however.

However, what we have gotten out of this project has transcended the calendaring system itself. Bedework and our participation in CalConnect has reconnected us the larger world and community of university software development.

Our open software project has allowed us meet, collaborate, and be influenced by so many talented people in higher ed around the world, an opportunity that probably would not have come our way if we had not engaged in an open software project. It is an opportunity to show the same kindness to others that the University of Washington showed us by welcoming us into their UWCalendar open software project.

It is an opportunity to continue the tradition of open software development of contributing according to our ability. It is an opportunity to reconnect with our own university by hiring a student to work with us on this project.

Ultimately, Joey “The lips” Fagan, the trumpet player in Alan Parker’s “The Commitments,” talking about what the band meant after it broke up, said it best, “You’re missin’ the point. The success of the band was irrelevant - you raised their expectations of life, you lifted their horizons . . .”

That’s what this open softwareproject has done for us professionally - it raised our expectations and lifted our horizons.

8 Comments

8.1 1. Ken Udas - October 19th, 2007 at 3:37 pm

Gary, First, thank you for this great posting. I believe that it is the foundation tram’sfor a very nice case study. I have a very broad question, so feel free to take it where you want to. Has your team’s involvement and leadership in Bedework had any noticeable impact on RPI (any particular part of the institution)? Ken

8.2 2. Patrick Masson - October 20th, 2007 at 10:51 pm

Gary, What I think is most spectacular about the development of Bedework is the development of Bedework. Many projects seem to first, form as a group looking to build a project, Bedework seems to be a project that is building a group: two models undertaken in the development of Moodle and Sakai as applications and communities.

I can remember, in 2003 while at UCLA, listening to Sakai conference calls, sitting in Sakai Conference sessions on Governance, Communications, Visioning, Strategic Planning and Collaboration, yet not allowed into “Sakai Core.” Also in 2003, I simply installed Moodle at the UCLA School of Dentistry.

To me, Bedework’s approach of letting folks discover the application and use it as they may need (or abandon it) seems more aligned with Raymond’s example of scratching that personal itch. Rather than homogenizing or neutering functionality to make an application palatable to all of those who have invested fup front, before development began based on a shared (arguable perceived) need, Bedework, and other needs-based projects will have more committed users who have adopted based on existing functionality meeting understood needs, yielding more focused development and, overall, a better application.

I think this is an important distinction as Higher Ed begins to accept Open Source. One of the often raised issues rejecting open source is the argument that running OSS requires a local developer. This attitude can easily lead to an “organize-first” approach, where senior administrators feel they must find partners, allocate resources and define objectives before a project can begin. This front loaded approach requires significant work to keep a project going, none of which is contributing to actual code development.

Consider successful OSS: how many of us that run Apache contribute code back, how about Linux? Imagine trying to get four major universities to define a server or operating system, then build it, versus slowly adapting one based on real-world needs by those who actually find it useful. This is the open source development model, this is how Apache, Linux and Moodle became successful: this is why Bedework will as well.

8.3 3. GarySchwartz - October 22nd, 2007 at 11:22 am

In response to Ken’s question: “Has your team’s involvement and leadership in Bedework had any noticeable impact on RPI (any particular part of the institution)?”:

The answer is “yes and no”.

With respect to the unit I manage, which has responsibility for many projects and services other than Bedework, it has been a little bit of a challenge to integrate the Bedework priorities, many of which are externally driven by installations at other universities, into our overall priorities. At RPI this has resulted,

to a certain degree, in an instance of the shoemaker's children going without shoes. It has been hard to find the time to deploy our own Bedework releases in production for our own users in a timely fashion.

Additionally, not everyone in our unit is a Bedework contributor. In that context it is important that Bedework not appear to be our favored or most important project. I can see how it might appear that way from time to time.

I am not sure that our experience with Bedework has had much impact on what we might call RPI's "institutional courage" to run open source software. We have the courage to run Bedework as our public events calendaring system, perhaps the courage to run an open source LMS such as Sakai instead of Blackboard, but not yet the courage to countenance even the thought of running Kualu, although we have the courage to run Banner on Linux, an open source OS.

In December 2006, the Bedework project was honored with a \$50,000 Mellon Award for Technology Collaboration (MATC) (see <http://matc.mellon.org/>)². This was significant in a couple of ways – as a very gratifying validation of the work we had done with Bedework, and it was the first award of any kind that our university had received from Mellon. Not surprisingly, the university is interested in parlaying this award into a larger relationship with Mellon, if possible.

Even though Bedework is RPI's public events calendar, many people on our campus do not know that nor do they care. So the 15 minutes of fame and minor celebrity that Bedework afforded us was lost on them. Additionally, in a research university context, a \$50,000 grant is a very small grant. At the provost level, I think there was some confusion about the fuss being made over \$50,000, and I can understand why.

We had a similar disconnect when we were directed to speak with one of our vice provosts to discuss calendaring. He was more interested in discussing a student developed calendaring widget, and suggested we ask them for their guidance.

RPI recently established the Rensselaer Center for Open Software (<http://undergrad.rpi.edu/update.do>). We do not really have any significant contact with this group nor do they look to us as experts or even people of interest concerning open source.

In some respects, this is not terribly surprising. The faculty and students are the soul of the university. It is their accomplishments, not those of the staff, that truly bring distinction to the university. Like Jerry Lewis before us, Bedework is more appreciated abroad than at home.

As I noted in my incredibly voluble original posting, the Bedework project has it raised our expectations and lifted our horizons. It reminds me very much of what I call the "Golden Age" of university computing, the 1980's, when RPI was a member of the MTS (Michigan Terminal System) consortium, with about 10 other universities in the US, Canada, and the UK. We had the privilege of collaborating with and competing with talented software developers from other universities, and that too lifted our horizons. The Bedework experience has been very positive and rewarding in much the same way.

8.4 4. Gary Schwartz - October 22nd, 2007 at 1:26 pm

Pat Masson's comments about the Bedework approach to building community and organizing our project may be over generous (but we thank him nonetheless) as it didn't really occur to us to go about it another way.

We believed that to be successful our project needed to transcend local objectives and local requirements, be standards-based, and provide enough obvious value that institutions would be motivated to deploy it without having to be "sold". This doesn't mean that we thought the community would build itself, but we felt the community should select itself, albeit sometimes with our guidance.

8.5 5. Ken Udas - October 24th, 2007 at 4:51 am

Hello, Very interesting stuff, and really important insights. Where do we look within the "academy" to see what type of impact our activities in OSS and/or OER might have? It seems to me that the impact of projects like Bedework might ultimately be through creating community outside of the institution, to which

²<http://matc.mellon.org/>

the institution can later refer. For example the stated goal of the Rensselaer Center for Open Software³ (this link points to a PDF):

This is the primary goal of The Center: to provide a creative, intellectual and entrepreneurial outlet for students to use the latest open-source software platforms to develop applications that solve societal problems. Moreover, the Center expands upon our commitment in The Rensselaer Plan to provide "... an undergraduate experience that surpasses all others, combining theory and hands-on experience as the means to educate tomorrow's leaders for technologically based careers."

is predicated on the cumulative efforts (and occasional courage) of folks like you and many others to take up the lead on OSS projects in environments that might not see the inherent value of such efforts beyond the instrumental contributions it is making to their home institution. Obviously though, on some level, OSS activities strike at an important value within RPI in terms of the Centre's mission, which ties together OSS and support of civil society.

The mission of the RCOS is to develop and adapt open software platforms for knowledge and information management in the context of promoting civil societies, both here at home and across the globe. (also from the Centre announcement)

Here is a sort of pragmatic question, has anybody, students or faculty, at RPI or outside, shown interest in contributing to the Bedework effort as part of their academic responsibilities (class, research agenda, internship, etc.)? Would that be seen positively by the Bedework project team at RPI? That is, I am wondering to what extent an "administrative" calendaring project (representing any OSS project) could also directly serve the academic mission of the host university.

8.6 6. GarySchwartz - October 24th, 2007 at 3:40 pm

To Ken's question of whether we would welcome the participation of RPI faculty or students in the Bedework project, the answer is a resounding "yes". There are some barriers to participation, which I will address, but one of aspects of community we were looking to address with Bedework was reconnecting with our own local community at RPI.

As an administrative unit with responsibilities for running centralized services, we do not provide direct end user support. Consequently it is sometimes difficult to feel connected with the academic life of the university. We seek appropriate opportunities to work with students, such as on Bedework, as it draws us back in to the primary mission of the university. Working with students has been a very positive experience for us.

Impediments to wider academic participation in Bedework include administrative policies with respect to funding of graduate students, as well as a university-wide effort to provide additional opportunities for undergraduates to participate in faculty research programs, which are for credit.

In some respects our project might be less appealing to students than some other opportunities on campus. Bedework is an enterprise calendaring system in the J2EE environment. It is a little harder to make a contribution immediately in this environment than perhaps with a desktop application, for example.

There are other programming opportunities on campus which are less constraining than working on Bedework. Bedework, exists, has an architecture, an implementation, and an implementation team already in place. The Rensselaer Center for Open Software (RCOS) ask student to propose their own projects, and essentially to manage their projects themselves. The Rensselaer Union, which is student run, and the student government also initiate sprogramming projects for students which are student managed. Some students work for companies in our incubator program (<http://www.rpi.edu/dept/incubator/homepage/>), and others program as part of their co-op assignments.

³http://www.rpi.edu/dept/cct/apps/undergrad/resources/PDFs/RCOS_Announcement.pdf

Last year we just missed the deadline (our “bad”) for Goggle’s “Summer of Code”, which would have afforded us another opportunity to work with students, albeit not necessarily RPI students. We do have an undergraduate working with us now, for money, not academic credit, and we have been approached just recently by a graduate student who was interested in Bedework. As I noted previously, his participation would likely be informal.

As I reflect on our current situation, I think it is possible that we might have been more successful bringing people from our own campus into the project had we concentrated less on trying to build an external community for Bedework.

8.7 7. Ken Udas - October 27th, 2007 at 11:52 pm

Gary, This was a very enjoyable post. Although I have occasionally had overall responsibility for IT departments, I have never directly managed an IT service unit. As a general and program manager, have always been supported by IT groups and have depended on their ability to meet program and organizational needs. It sounds to me that the experience that you have had with Bedework (a successful OSS project) could improve ones ability to better support internal projects and work units. Thank you! Ken