I have wanted to put my fingers on the keys about this subject for a long time. The whole idea of an administrative information systems implementation makes my heart beat fast and my stomach churn. But these are not the feelings that come with the anticipation of a new love. They are the feelings that come with the anticipation of something I dread. And each new implementation—or, even, each substantial upgrade to a working product—brings this sense of dread.
In this article, I will not be using the term ERP (enterprise resource planning) systems. I am specifically focusing here on administrative information systems. Certainly, these big systems are ERPs, but we implement other ERPs too. For example, on my campus, our course management system has become an ERP. I regard our e-mail and calendaring systems as ERPs. It is the administrative systems implementations that I dread. The others are much easier.

As you can tell, this article is an opinion piece. It is not based on a data-collection effort. It is not based on a literature review. It is not based on a discussion with the Gartner Group or the Burton Group or the EDUCAUSE Center for Applied Research (ECAR). It is based on personal experiences in organizations where implementations were part of the job and on discussions with friends and colleagues who are part of organizations similar to mine. I do note, however, that two ECAR studies helped me to understand that it is not only me and the colleagues with whom I have spoken: colleagues nationally struggle with administrative systems implementations. This tends to be a generic problem.

I became part of administrative systems implementations too many years ago to count. In the first implementation in which I was intimately involved in a leadership role, I was very naïve and not very smart. I made some serious mistakes. “Go-live” was a disaster, and the disaster was the result of poor planning and a lack of understanding on my part. It was an IT fiasco. Eventually, IT fixed the problems. This was the easiest administrative information systems implementation that I have ever been a part of.

During later implementations—and these were in the hospital sector, not the higher education sector—it became clear to me that if you are in a leadership role, work successfully in our settings. So, no matter how well this job is done, there will be post-implementation pain.

I don’t want to imply that all implementations are this painful and difficult. I may be overstating the case. A couple of my male co-workers believe I am. My female co-workers, however, are not sure I have described enough of the pain. I’m not sure why this is so; perhaps it has something to do with the EQ of those involved or their Myers-Briggs scores or simply their personal pain tolerance. In any case, I am speaking from a limited experience—from my work in very large and complex organizations in which decision-making is highly distributed. And in this setting, some of the administrative systems work we do is indeed easy and routinely successful.

The question remains: Why are many of the administrative information systems implementations so dreadful? It is not the software. Most of the software products we are implementing are mature. This doesn’t mean that they are bug-free or that they meet all the requirements we impose, but the base products work. As we face our implementations, we can usually find colleagues who are in similar institutions and who have implemented the product we are implementing. We can even share information and some of the necessary interfaces and pieces of code that support essential “bolt-ons.”

And we know how to make the hardware work. We have staff who keep the hardware up and running, who develop hardware maintenance schedules, and who configure hardware for great throughput. Of course, sometimes the system goes down during testing because the hardware is not configured right, but that’s what testing is all about. And maybe the system wasn’t installed on the “right” hardware platform. We know how to fix that too, although there may be a cost associated with it.

It’s not the software, and it’s not the hardware. What are the problems? Here are the things that I most dread about the implementation of administrative information systems:

1. The implementation budget estimate will not be accepted.
2. The sponsoring organization and the financial leaders will not understand that the estimated budget is an estimate.
3. A sponsoring organization will not view the IT organization as a partner in the implementation but rather as a technical “body shop.”
4. A sponsoring organization will believe that it is better at managing the IT technical staff than are the IT leaders and project managers.
5. A community of users will not be sure of their requirements until they see a product, and then they will know that the product is not what they want.
Financial leaders who are looking at the bill almost always challenge this preliminary estimate and ask the IT leaders to reduce it.

6. The implementation will not go according to plan.
7. Organization leaders will not be public champions for this work.

I am going to talk about each of these dreads, acknowledging again that my statements are personal opinions and the results of discussions with colleagues.

1. Developing the Budget Estimate

Very preliminary estimates may be floated before the selection (or before the detailing of specs for a build decision) of an application product to be implemented. People on campus will almost certainly talk to colleagues around the country to find out what others did and how much it cost them. Numbers will be tossed in the air to begin to frame ideas about what the costs should be. If some people in the larger community are bragging about how they implemented a big application at a very reasonable cost and if that's the cost that the financial leaders like and hold onto or that the sponsor thinks is right, there will be trouble from the get-go.

Initial budget development cannot take place until the product to be implemented has been selected. Hopefully, that selection will have been based on users' and technologists' specifications. Now the serious job of developing a budget estimate begins. The IT folks need to estimate the costs for infrastructure, training, applications development, data conversion, security, testing, operations, production control, backup, and whatever else is needed to implement and operate this new application. The sponsor departments need to project the costs for consultant support in their areas, for back-up for staff redeployed to support the implementation, for process and procedure development, and for end-user training. In my type of institution, the estimate for major administrative implementa-

tions usually has eight digits in front of the decimal point. This is where the first burp occurs. Financial and often sponsor leaders want to see a smaller number. Pressure is brought to bear on the IT folks to significantly reduce the cost estimate. And remember, we are still talking about a cost estimate.

Financial leaders who are looking at the bill almost always challenge this preliminary estimate and ask the IT leaders to reduce it. Certainly their position is reasonable. We are all challenged to manage our resources effectively. IT can explain that the budget can be reduced if the scope is reduced, but that is not what the campus partners want to hear. IT is challenged to accomplish all tasks at less cost. Sponsors and finance are hoping that IT has slack in its budget and can cover some of the costs without charging back. They want to hear that IT can do the work without IT project leadership. They want to hear that IT staff will work longer and harder and accomplish more in less time. The campus wants this to cost less, and because IT costs are the largest part of the budget, that is where reductions are sought.

I think this budget issue is where the first disconnect occurs between the IT leadership on the one side and the campus financial leaders and the sponsoring unit on the other.

2. Explaining the Budget Estimate

We don't really know what the actual costs of the IT development and implementation will be. We develop estimates down to the work hour for the programming and database administration staff and other IT staff. Those estimates hold reasonably well if the specs remain exactly as they were presented when the estimating began, and if the product works exactly as the vendor indicated, and if we understand all of the by-products that roll off onto other systems and applications (e.g., interfaces), and if no one becomes ill and needs time off, and if our project managers are really very good—and the list goes on. These estimates are extraordinarily detailed, but they are not science. The preliminary budget proposal is not a hard fact; it is an estimate.

The numbers prepared in this initial planning phase are as good as the skills and experience of the people preparing the numbers. And even the most skilled and most experienced people can't calculate the exact numbers. I believe that the preliminary budget estimate works as a good tool. It helps us understand approximate project costs if the project proceeds as planned. Any budget estimate needs a contingency allotment to manage mandated changes and unexpected occurrences. We do not know what we are going to encounter in our institutions as we begin this work.

Changes in cost usually result from changes in users' requirements or from a misunderstanding of how the selected system works. This misunderstanding can be a misunderstanding of the functions provided by the system or a misunderstanding of the technical capabilities (for example, ability to handle load). Sometimes misunderstandings result from a shortcoming in the vendor's product. Regardless, this unknown shortcoming will cost the project time and money, and those costs are not at all likely to be recovered from the vendor.

An effective budget for the technology side of the house would be one in which the estimate of the technologists was not discounted and in which a sizeable contingency fund was added to cover unknown events occurring during implementation. Interestingly, this could reduce the eventual cost of the implementation because leadership
would not be continually involved in budget battles and could spend that time supporting project staff. Staff might less frequently be asked to do the impossible, thus reducing sick time, turnover, and ineffective working-while-exhausted syndrome.

3. Making IT a Partner
On my campus, we have one major functional area that views the IT organization as its strategic partner. This area and this major division jointly sponsor big administrative systems projects. What a delight! Our leaders work with leaders in this organization from the time there is the glimmer of a project possibility. And that glimmer could as easily arise from the IT staff as from the user staff. In working with this key partner, we build budgets together, we trust each other to assess our costs honestly and as accurately as we can, we work together to reduce requirements (not to do the same thing for less) if costs look too high, and we continue to collaborate throughout the life of the project. We have joint project reviews. We have successful implementations. We jointly announce our shared efforts and our successes. And we celebrate together. Ownership and pride are shared.

Other projects aren’t as easy. And projects haven’t been as easy in the other institutions where I have worked or at the home institutions of the colleagues and friends with whom I have spoken. The 2002 ECAR research study “The Promise and Performance of Enterprise Systems for Higher Education” explains some of the problems. This study suggests that the primary obstacles for successful administrative systems implementation relate to people, not technology. A key “people” problem is the challenge of establishing a partnership between the sponsoring unit, which will “own” the system, and the IT organization, which will implement and operate the system. Trust is a huge issue here, and trust comes from people and from the culture. It also comes as a result of keeping commitments. The cost model for systems implementation at any given institution also influences this partnership. If the sponsoring unit holds the budget, and trust is not the modus operandi, the sponsor may be able to demand to manage the IT staff as if the staff were hired contract programmers. Professional IT project leadership can be discounted in this situation. The leaders in the IT organization are removed from day-to-day knowledge about the project and are called on only when the project is not progressing well or when other resources in the IT organization are needed.

There are many reasons why this model of using the IT staff as contract programmers may be the preferred model for a sponsoring department. A good number of these reasons go right back to trust. The department may have had a bad experience with the IT organization in the past; many IT people have a touch of arrogance that doesn't sit well with those in user departments; IT staff use “IT speak” instead of “user speak” and thus are regarded as not understanding the situation; and IT may have a long-standing reputation on campus for overcharging for services. The more likely reason is that the leaders in the sponsoring department don't want to share control of their systems implementation. They don't trust the IT organization and its leadership, they want to do it themselves (often with the aid of external IT consultants), and they want to “own” the IT staff for the duration of the project.

This is often demoralizing for the IT leaders and any assigned staff. It also fails to take into consideration that the IT staff who are required to implement a significant application include not only the developers but also those with expertise in database administration, security, identity management, operating systems, production control, back-up, and operations. An arrangement to “body-shop” the IT staff routinely results in a less successful implementation and, later, challenges in the operations mode.

4. Managing the IT Staff
Even if the IT staff are not turned over to the sponsoring department, the sponsor's people working on the project often believe that they should be managing the IT staff assigned to the project. There is a desire to know exactly who spent what time completing which task. There is a desire to specify which IT staff member should be working on which task. There is a desire to find out why it took Sam four hours to complete task A when it took Sally only two hours to complete task B, with the implied questioning of Sam's competence and the competence of the IT project manager. This questioning is done without recognition of the complexity of the assigned tasks. It is done without acknowledgment that the specs for task A may have been changed by the user midtask or that the underlying product may not have done what it was supposed to do.

Such attempts by the user to manage the IT project demoralize and devalue the IT project managers and leaders and also ignore the fact that the time spent justifying who is doing what, how many hours it is taking, and why it is taking that kind of effort is time not spent supporting the IT project team members and managing the project plan. I do believe that the IT leadership owes good reporting of progress-against-plan to the sponsoring leaders. IT should be working closely with the sponsoring leaders throughout the project. The plan should be adjusted by IT and the sponsors together as necessary. A partnership in which the partners focus together on the big challenges of
the implementation would bring more satisfactory and satisfying results.

5. Changing the Requirements
I understand why requirements change. We can do the most detailed requirement analysis before beginning implementation, but it is not until results start to roll out of the project team that we can more clearly see what the products of the system will be and we can gain a new understanding of what will work and what won’t work. Requirements change because we have a better view of what is possible. Being able to adjust the implementation plan—and, of course, the project budget—to accommodate the changing requirements serves the community well.

Another kind of requirements change is more frustrating. This occurs when users who have not been intimately involved in the project realize that their business practices will change with this new system. Resistance to new business practice grows like a thorny shrub grows in the spring. The IT staff then is directed to make the new implementation behave and deliver very much like the legacy applications being replaced. This does not serve the larger community well. It results in an implementation with excessive customization—customization that affects the life-cycle costs of the new system. It also means more work, more time, and more costs—none of which was budgeted during the implementation process.

6. Killing the Messenger
I’ve said earlier that systems implementation is not a science. The process of budgeting for and implementing large information systems involves a good deal of art. I have yet to see an implementation proceed as planned. I have seen some very successful implementations, and I have seen some disastrous failures. But even the successful implementations did not proceed as planned and budgeted. There are cost overruns. Remember, the estimated budget is usually reduced and then locked in stone. The IT folks probably provided an early warning that such a model would result in cost overruns. Critical milestones can be missed. Projects can take longer than expected. Project

Large systems implementations are more successful if senior institutional leadership is seen as being committed to the project’s success.
managers need to report that information to their leadership and to the sponsoring organization. Key project sponsors need to be informed. Key institutional leaders need to be informed.

Certainly there are times when the IT organization performs less than competently. But regardless of whether the suboptimal performance is a problem of competence, a problem of scope creep, a problem of budget overrun, or a problem with the ability of the system to meet the requirements, leadership needs to know. And that leadership is more likely to be informed if the people reporting the problems—the messengers—understand that they will be heard. This works even better if the messengers know they will receive some help in identifying corrective actions or next steps. Our messengers need to be brave. They also need to be treated respectfully.

7. Getting Support from Senior Institutional Leadership

Large systems implementations are more successful if senior institutional leadership is seen as being committed to the project’s success and to the improvements that the project is expected to bring to business processes. Large systems implementations are supposed to improve staff effectiveness in doing administrative work—work that supports the institutional mission of teaching, learning, research, and outreach. Large systems implementations are costly, and the money spent is money not available to be spent directly on core mission components. Therefore, these investments in new information systems should be undertaken only if they improve (or, in some instances, sustain) the staff’s ability to perform mission-critical activities.

The institution must enroll students, report grades, meet payrolls, fulfill state and federal reporting requirements, and account for grant funds. The more effective and efficient staff are at doing this administrative work, the more energy and resources will be available for the mission-critical activities. Systems implementation is much easier if institutional leaders champion these large implementation projects and the benefits they are expected to bring. If presidents, chancellors, provosts, chief financial officers, and deans regard the implementation as critical to the future of the institution, the campus community will be more willing to support the efforts, the costs, and the changes this implementation brings. It is not enough for the CIO and the director of the functional area to be the key project champions. The campus communities—communities of faculty and students as well as staff—need to see the recognized campus leaders as champions for this new system.

The work involved in large administrative systems implementations is very difficult and very complicated. This work is made more difficult by the fact that we have not learned how to identify, with enough certainty, what a project
The IT organization needs to demonstrate that it knows how to develop a good budget estimate and a good plan.

will cost the institution. And it is made more complicated by a lack of trust between the sponsoring organization and the IT organization, as well as a lack of obvious support from institutional leaders. I attended a Cisco CIO seminar this past year. John Chambers, the CEO of Cisco, talked to the group about how a solid IT environment was strategic to the success of his company. He said that his CIO has an unconstrained budget because Chambers is convinced that to ensure the company's ongoing success, Cisco needs to continue to invest in the administrative IT infrastructure. Clearly, at Cisco, the IT budget is adequate to the job, the IT organization is trusted, and the CEO is a champion for the administrative systems efforts.

I am not proposing unconstrained budgets or unfounded trust and support. The IT organization needs to demonstrate that it knows how to develop a good budget estimate and a good plan and how to implement to that budget estimate and plan. Institutional leadership needs to respond to this demonstration of competence and success by trusting the IT organization to manage the implementation project and by publicly supporting the organization's work in large systems implementation. In this scenario, IT would be more productive. Work would be more fun. And I would have nothing to dread.

Notes