Avoiding Problems in Implementing Administrative Systems

When software implementations go bad, the temptation is to blame the software. Some faults do indeed lie there, but many others lie in our administrative decisions. This article examines four aspects of administrative systems implementations that administrators need to understand to avoid implementation pitfalls.

by **Joel M. Smith**

The September 24, 1999, article in *The Chronicle of Higher Education,* entitled "Delays, Bugs, and Cost Overruns Plague PeopleSoft's Services," reported on the frustrations that administrators at a number of institutions have experienced in implementing a new generation of administrative software from PeopleSoft. In fact, similar frustrations have plagued institutions implementing commercial student administration, financial, and human resource systems from the time they became a viable alternative for colleges and universities to locally developed software.

As the article noted, a number of structural features of that class of software, often known these days as enterprise resource planning or ERP software, can lead to cost overruns and dissatisfaction with the results of its implementation. Those features include the software's complexity, the difficulty of deciding about initial configuration options, the dangers of choosing to customize the software, and the realities of coping with bug fixes and updates to the software. However, the article did not give details about how to avoid or at least mitigate the problems it described.

Even worse, without more detailed information about the challenges of implementing ERP systems, the article might lead an administrator new to dealing with this class of software to the incorrect conclusion that he or she must simply choose the right software, that is, something other than PeopleSoft, to avoid the problems described. That conclusion would be unfair to PeopleSoft, a company that has worked closely with higher education to produce high-quality, innovative products for our purposes.

More importantly, the notion that there is a product from any vendor that will work right out of the box with few dangers of cost overruns or of dissatisfaction with the results is simply mistaken. Administrators who reach that conclusion will make bad decisions for their institutions. Many of the dangers described in the article are inherent in the nature of this genre of software. Installing a complex administrative software system that is sufficient to meet the needs of a modern college or university requires more sophisticated knowledge and tactical decisions from us than have been required of college administrators in the past.

While it would be nice if we could depend on software vendors to educate us about the prerequisites and ramifications of implementing, maintaining, and using their products, we cannot. Salespeople say what we want to hear. For example, they often emphasize the very features of the systems—for example, ease of customizing the products to the potential buyer's current business practices—that will lead to serious problems down the road. They know we want to hear that we really don't have to change our ways to use their software.

Nor can we depend on the consulting firms that make their living implementing administrative software to give us the complete story about deploying the systems. After all, by far the biggest cost in any such implementation is often the exorbitant fees we pay consulting firms to help us set up and customize the software. Although they can be valuable partners, their interests do not always coincide with ours. Even the use of consulting firms at the beginning of the process cannot guarantee good cost estimates or risk assessments. The actual costs of implementation will not emerge until a detailed analysis is performed of

how well the "vanilla" system fits your needs. This is usually a part of the implementation process that can take weeks or even months. Vendor or even third-party estimates of costs made without this information are often gravely mistaken. Ultimately, we have to depend on ourselves to know about the strengths and weaknesses of the software we are considering and about the inevitable risks and high costs of the process.

There are a number of good software systems, including PeopleSoft, that institutions can purchase. The problem is that it is all too easy to implement any of them using strategies that will lead to both short- and long-term problems. To prevent the problems, administrators need to understand the complexity of the systems; the dangers of customization; the critical nature of documentation; and the real costs of institutional staff time that must be devoted for the project, training, and the loss of key personnel.

System Complexity

First, the software packages are complex systems. Changes made to computer code or database structure in one part of the system can affect other parts. That is both good and bad news. It makes fixing some problems easy. One community college in California solved performance problems throughout its PeopleSoft system by making fairly simple changes in the programming commands that put data in and retrieved data from the underlying Oracle database. But changing code to fix one part of the system can produce problems in another part.

Knowledge of that aspect of large-scale software should result in some concrete administrative strategies. Changes to the software must be made serially, be heavily documented, and be tested carefully for unexpected consequences. If the staff of a consulting firm or the institution's own information technology (IT) staff is allowed to operate in any other way—for example, to make many changes at the same time or to fail to document changes carefully—unexpected problems and cost overruns are likely to occur.

Dangers of Customization

Second, implementation decisions must be made with future maintenance in mind. Failure to understand that fact is the most serious mistake administrators can make in implementing such software. For example, PeopleSoft allows the customer to customize its software or to create new applications to use alongside those that the company has developed. All commercial administrative systems either permit or require customization. Meeting your own business and student service needs is very likely to lead you to choose to customize the software.

Customizing a commercial application creates significant difficulties when the vendor releases a new version.

However, customizing a commercial application creates significant difficulties when the vendor releases a new version. That version might contain new features that conflict with the changes you have made or that remove structures you have depended on in your customization of the product. If you have customized the software, your IT staff will have to spend a great deal of time evaluating the relationships between your customizations and the vendor's changes before you can proceed with any upgrade. The more changes you make, the more time it takes to go through the process every time you upgrade your software.

Even though PeopleSoft provides sophisticated tools to help with the process of comparing your software with the new version, upgrades of extensively customized systems can take months. That is true for all of PeopleSoft's competitors too. If you don't have enough IT staff members to perform the upgrades, you will have to pay for high-priced consulting help.

Here again, knowledge of the details should lead to concrete strategies. You should change your business practices to match your software instead of customizing the product. That is going to be uncomfortable for many staff members, but not as uncomfortable as not being able to upgrade or patch the software because you don't have the resources to update a customized product.

Any good software will include ways to tailor it to your needs that don't involve customization. For example, with PeopleSoft you can write your own self-contained subsystems that don't cause the difficulties described above at upgrade time. In any system, you can use the report-writing tools to create custom reports that extract just the information you need from the system without customizing the software.

Documentation

Third, documentation of set-up decisions and changes is critical to a successful implementation. That may sound obvious, but the reality is that neither consultants nor information technology staff members like documenting, so it seldom gets done well, if at all. Poorly documenting the implementation of a complex administrative software system leaves the institution at the mercy of IT staff members, who are notoriously difficult to retain these days. Even worse, failing to create clear, usable, comprehensive documentation means that the

software cannot be upgraded without figuring out how it was set up in the first place, which takes time and money.

Real Costs

Fourth, it is easy to misestimate the amount of time that the institution's own staff will have to devote to changing from a legacy to a commercial system. When you are paying consulting partners millions of dollars for the implementation, you might think that will be almost the entirety of the personnel costs for the project. The reality is that internal staff both from the IT groups and from virtually every office in the college or university will have to work side by side with the consultants in the implementation. Many will need to devote between 25 and 90 percent of their time to the project for varying periods of time. Temporary staff must be hired to backfill the internal staff who are working on the implementation. The result can be a significant cost overrun. The strategy should be to develop painfully realistic projections of the amount of time institutional staff will spend and the real costs (including recruiting and training costs) of backfilling their positions.

Training for both IT staff and users of the system will exceed your initial estimates. The strategy of training only a few staff with the expectation that they will return to train their fellow workers succeeds only if the returning staff are good trainers. Since it is seldom the case that they were hired for this talent, it is unlikely that many will be either good at or comfortable training their peers. This means that far more people will have to be sent away to the vendor's training than originally expected. A training plan for everyone on campus who will either support or use the new system should be part of the original cost projections.

Finally, any such project is vulnerable to the risk of losing key personnel. This means that plans for redundancy for everyone in the project—from the project manager to the database manager to the person in charge of posting progress reports on the Web—should be built into the plan and the costs of the plan from the beginning. It's cheaper in the short run not to create redundancy; in the long run, depending on your luck, it can be far more expensive.

When software implementations go bad, the temptation is to blame the software. Some faults do indeed lie there, but many others lie in our administrative decisions. We expect complex software systems to work right out of the box. We fail to arm ourselves with an understanding of the details of the systems we have chosen. We train our staffs insufficiently or incorrectly. We choose the comfort of customizing software to the way we've always done things over the difficulties of using the basic, easily upgradable product. We let staff members get away with poor documentation. We turn too many tasks over to consultants so that our staff members are lost when the consultants leave.

The technological sophistication required to implement administrative software is greater than that to which academic administrators are accustomed. But no piece of shrink-wrapped software alone can provide the functionality we need to serve students who live in the information age. We have to develop the more complex strategies required to implement and manage the tools of that age.

Joel M. Smith (joelms@andrew.cmu.edu) is director of the Office of Technology for Education at Carnegie Mellon University.

This viewpoint was adapted for the EDUCAUSE Quarterly *from an article that appeared in* The Chronicle of Higher Education, *October 22, 1999, B12.*

NETWORK

continued from page 45

provide/ensure campus coverage and potentially to share in risks and rewards. The key is to remember how quickly technology changes and that vendors come and go quickly as do substantial revenue streams.

The good news is that there is nothing mysterious about preparing a college or university for a networked future: a campus network can develop and grow according to well-understood principles in an orderly, well-coordinated process of planning and implementation. Today's campus leader, while having little need to be an expert in network technology, must nevertheless take personal interest and provide attention to ensure that this happens.

Endnote:

1. This article was adapted by the author from a chapter he authored in Mark A. Luker, ed., *Preparing Your Campus for a Networked Future* (San Francisco: Jossey-Bass Publishers, Inc., 2000). The book is available from EDUCAUSE (for ordering information, see http://www.educause.edu/pub/pubs.html#books) as well as from the publisher (see http://www.josseybass.com/). The material will also be included in a chapter of *College and University Business Administration*, to be published by the National Association of College and University Business Officers.

Philip E. Long (philip.long@yale.edu) is director of academic media and technology in Information Technology Services at Yale University.