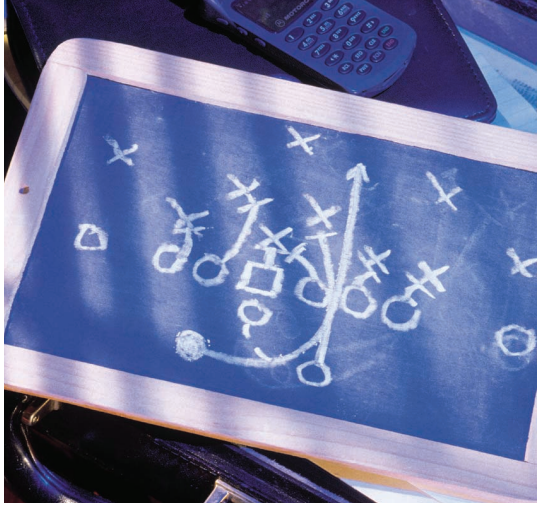


Implementers of a successful university ERP system share experiences and lessons from an IT perspective.

Ilya V. Yakovlev and Mark L. Anderson

Lessons from an ERP Implementation

In 1999, the University of Wisconsin-Superior (UWS) launched a project to implement an enterprise resource planning (ERP) system called PeopleSoft Student Administration (SA). The new system would replace the university's aging, home-built legacy student information system (SIS) and several third-party subsystems. The university's chancellor and provost envisioned that the SA system would integrate existing administrative subsystems, meet increased demand for student Web-based services, and provide prompt and accurate reports. The implementation was, by and large, a success.

BACKGROUND

UWS is a small liberal-arts college in Northern Wisconsin with around 2,700 on-campus students—one of 13 four-year universities in the system.

UWS built its legacy SIS on a Unisys A-Series mainframe in the early 1980s, with several major additions implemented in the 1990s. The system provided touch-tone registration, automated billing, and direct lending. It was integrated with a full, personalized, degree-progress-reporting tool called Darwin (Degree Audit Reporting System for Windows), a popular advisement tool on this campus developed by Miami University in Oxford, Ohio. The SIS performed well but only faculty, advisors, and administrative staff could access it. The system's monolithic, proprietary nature also constrained new development.

The SIS included several subsystems such as third-party financial-aid and student admission tracking systems. Remote vendors supported them, introducing delays in operations when problems occurred. UWS needed a new internal system to replace these products and to integrate student data management—from the tracking of prospective students to their graduation and introduction into the alumni population.

The university's Admissions and Financial Aid offices had third-party systems. The campus IT staff knew little about these systems and could not provide support. Integrating these third-party systems with the SIS was complicated and required review every time vendors upgraded one of them.

In 1998, the University of Wisconsin System purchased a systemwide license to use PeopleSoft's SA system. A steering committee that included leaders from main functional areas recommended replacing UWS' legacy SIS with an ERP implementation of the SA product. Motivating factors included the increasing student demand for Web-based services, the need to integrate several subsystems, and the desire to deliver reports to users (top management, faculty, and administrative staff in core functional offices) in a standard format. The new system promised to make available to students such services as Web-based registration, financial-aid information, billing information, unofficial transcripts, and degree auditing. In addition, the SA system would enable Web-based class rosters for instructors and degree audits for advisors. Users could design their own reports and run them against real-time data.

The provost formed an implementation committee involving key personnel from the Admissions, Registrar, Bursar's, and Financial Aid offices to address the implementation details for those areas. UWS implemented the SA system over a two-year period (see Table 1). The system is now live. The project manager had defined a successful implementation as one that delivers

1520-9202/01/$10.00 © 2001 IEEE

Table 1. Project timeline.

Year	Spring	Summer	Fall
1998		Committee formation Project planning	Project management consulting Technical training Hardware selection Platform and operating system selection
1999	Hardware installation Student Records training Student Financials training	Admissions office training Admissions and Student Records consulting and setup Migration of historical student records	Migration of current student records Migration of current and historical admissions data
2000	Admissions live with current prospect entry and with fall 2000 applicant entry Student Records live with fall 2000 Web-based registration Financial Aid consulting and setup Additional technical training	Spring/summer 2000 prospect and applicant double entry Student Records double entry for spring/summer 2000 activity Student Financials setup	Student Financials live Financial Aid live Admissions live Student Records live Additional consulting in all areas Celebration

key functionality on schedule and within budget. Even though some features of the system are not yet implemented, the implementation teams of functional users and IT staff consider this project a success.

MIGRATION APPROACH

The goal in this implementation was to migrate core functionality to the new system on time and within budget. Numerous reports of failed ERP implementations illustrated the importance of adhering to the project schedule and following implementation rules such as leaving the delivered product unmodified and providing adequate training to implementers and users.

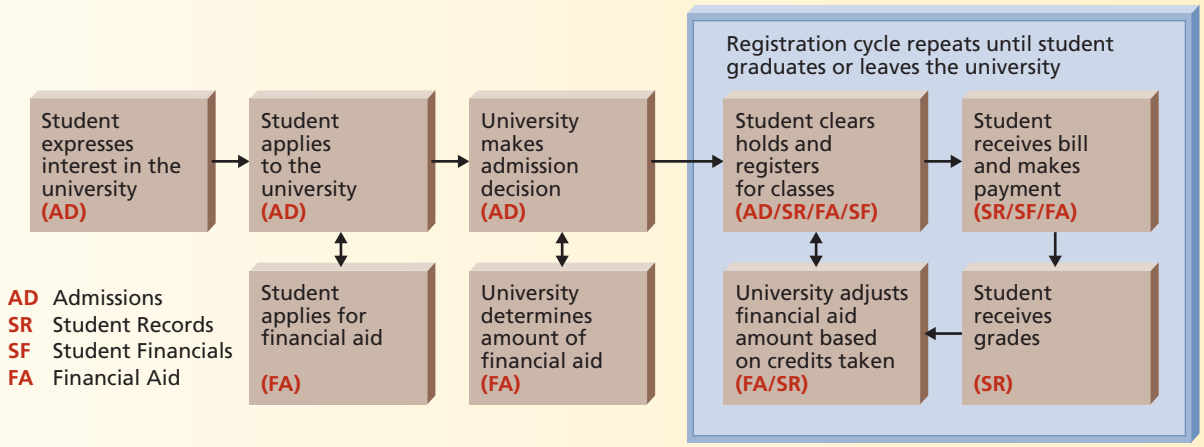
The provost appointed the director of Administrative Information Systems (AIS) to manage the project, provided project team staffing and training, and strived to maintain timely executive decision making. Top management also authorized offices involved in the implementation to hire part-time staff to help with the project. The campus IT department froze all noncritical new IT development and administrative changes to focus the implementers' efforts on this project. In addition, the implementation committee defined clear, realistic business requirements, including the requirement that IT staff make only minimal modifications to the delivered system to simplify maintenance.

The approach for the migration was a version of direct cutover, implementing all system modules chronologically close to one another—taking advantage of the school's small size. The direct-cutover approach had the benefit of minimizing the need to enter data in both the legacy system and the new system or to write bridges for migrating data between the new system and the legacy system. An alternative approach, such as parallel runs or phased migration, was not feasible. Functional offices were already overburdened with existing duties and could not handle double entry into two systems for an extended period of time. The campus IT department did not have enough staff to build and maintain bridges between the SA system and the legacy SIS. The ideal scenario would have been to implement all modules at once, but given the system's complexity and the differences in each module's business cycle, that approach was not feasible either. For example, prospect data entry for a new fall term starts in January of the prior year, whereas student data for early fall registration is needed in the spring of the same year (see Figure 1 for module interdependencies). The implementation committee decided to migrate legacy data at the beginning of each new business cycle. This approach enabled running final processes of the last cycle on the legacy system, migrating this data in its entirety, and starting entry for the next cycle on the new system.

TRAINING

Technical training on the new system began in summer 1998 and functional training in summer 1999. Early technical training gave the IT staff a head start on acquiring the hardware and installing the system. The functional

Figure 1. Student processing in PeopleSoft's Student Administration system.



Admissions is responsible for prospect tracking and student admission. Student Records handles registration, grade entry, and record keeping. Student Financials handles student payments and fees. Financial Aid manages financial aid awards and direct lending.

training involved learning how to enter instructor, advisor, and student information, and activate their records in the system. The project manager tried to schedule most training off site so that those involved would not be encumbered by their everyday job tasks. Some additional staffing helped IT staff and system users offload existing duties throughout the implementation process. The project manager also tried to send users and implementers to training no more than one month before they started working with the new system. Upon return from training, users would typically start configuring their module and inspecting migrated data as soon as possible, usually within two to three weeks. In 1999, PeopleSoft released a new version of the system. AIS did not implement the new version until 2000, to avoid having to send functional users and IT staff to additional training. In February 2000, a PeopleSoft consultant helped apply the new version. The project manager invited additional consulting help for each of the four functional areas to aid with the setup of each respective module.

PROJECT SCHEDULE

From the beginning, the campus IT department issued hardware and operating system specifications for user workstations. The hardware was to exceed PeopleSoft's recommendation in every category (permanent storage, processor speed, and so on) to lengthen the life cycle of the equipment and to minimize risks associated with new, potentially more demanding, releases of the system. For the operating system, AIS recommended adopting Microsoft Windows NT and Microsoft Explorer as a campus-wide standard combination for all administrative workstations.

Implementation teams installed four modules of the SA product:

- Admissions,
- Student Records,
- Student Financials, and
- Financial Aid.

The first module to go live was Admissions: Training occurred in summer 1999; prospect entry began in January 2000. Student Records was the second module to go live: Training took place in fall 1999; by April 2000, the implementers had migrated student enrollment, legacy and transfer credit, catalog, and class-scheduling data. UWS students registered themselves on the new system for the first time in April 2000. The legacy system still contained spring and summer 2000 term activity, which could not be migrated until the end of these terms. The campus IT staff avoided writing bridging programs between the two systems; doing so would have increased the project's complexity and put the project over budget and behind schedule. Student Financials and Financial Aid were the last two modules to go live. Like the Student Records module, they required finishing activity for spring and summer semesters on the legacy system and new processing of fall registrations on the new system.

The last major piece of the implementation involved writing an interface between the SA product and the Darwin advisement tool. Retaining this tool and integrating it with the new system were important to minimize the amount of change the SA product brought about. The degree audit

system provided unique functionality that was used for clearance of graduating students and for advisement of current students. The campus IT staff made Darwin reports available in paper form in fall 2000 and on the Web in spring 2001, to help advise students for fall 2001 registration.

DATA MIGRATION

The major challenges in migrating legacy data were in having to

- split several legacy data structures into dozens of linked tables, and
- add a date dimension to most student- and instructor-related data elements.

For example, the legacy system did not allow tracking of student or faculty name changes. Only the latest name was stored and made available. The SA product let us store multiple rows. Each row would track a student's or instructor's name as of a certain date. The more complicated cases concerned historical changes in administrative procedures. The implementers from the Registrar had to recreate some historical course information by entering the name that a particular course had several years earlier.

Any upgrade can affect existing functionality, and testing the areas most likely to be affected is crucial.

The campus IT staff was concerned with migrating too much data. Data going too far back chronologically would likely be less accurate and could significantly stifle the new system's performance. AIS recreated details of student and course data back to 1988. Any data beyond that point was migrated in summary form, without supporting detail.

AIS developed a mapping document that related legacy data elements to the SA data elements, allowed narrative, and placed a deadline for migrating that particular table. The implementers expected the migration to be an iterative process. The mapping document tracked counts of valid and invalid records. As IT staff migrated a particular table to a test database, they wrote down the two counts. Whenever the count of incorrect or incomplete records was high enough, the administrative department responsible for that particular structure's data entry received an edit list so that the data could be fixed. IT staff, with the help of functional users, migrated every table several times until the count of invalid records was zero—or low enough, in the case of very large tables, which the IT department or the appropriate functional office fixed later. Meeting weekly with the users responsible for administering or operating each SA module, AIS helped with module configuration and clarified any questions on the data migration edits.

The SA product required a major change in handling duplicate records. Although the legacy system was somewhat tolerant of inadvertent multiple records for a student, the new system produced many pages of edits related to duplicates. Eliminating duplicates was difficult. The implementers' approach was to focus on current students, ensuring that no duplicate records pertained to them, and then fix older duplicate records later. The elimination of duplicate prospect records is especially complicated because such records typically do not carry a unique identifier such as a social security number. The implementers agreed to accept such duplicates at the prospect stage, catch many of them with exception reports, and catch all of them when these prospective students applied to the university.

PROCESS MIGRATION

One of the main drawbacks of ERP systems is that they require a business to change its practices to fit the mold of the product. Looking back at the project, more time should have been planned for reengineering existing processes to fit the mold provided by the SA product. The implementers retained many of the legacy practices and either configured or changed the SA product to fit those practices. Because much of the fitting was accomplished through the setup, this was not a major problem. Nevertheless, additional reengineering would have helped the implementation go more smoothly.

MAINTENANCE

The system's vendor is continually improving the system and correcting processes that do not work exactly as designed. In addition, some processes must be periodically updated to comply with changes in federal regulations. The Financial Aid module can require many such changes throughout any given academic year. IT staff reviews the necessary upgrades on a three-month cycle and implements them in a testing database. In some cases, the IT staff must immediately apply a critical update to fix an urgent flaw in a process or screen. After an upgrade is placed in a testing database, system users help test existing functionality and any new functionality that a particular upgrade provided. Any upgrade can affect existing functionality, and testing the areas most likely to be affected is crucial.

LESSONS LEARNED

For the most part, the new system is a welcome change on campus. However, involving students and faculty in the implementation process would have helped its acceptance. Having excessively large implementation committees was undesirable, but having under-represented committees is undesirable as well.

The direct-cutover approach is very effective for this type of system. This approach is aggressive and short-term

Table 2. Core implementation teams, in addition to the respective office supervisors and cross-functional participants.

Implementation team	Employees
Admissions	One from IT, three from Admissions
Student Records	Two from IT, two from Registrar, one from Advisement
Student Financials	One from IT, two from Bursar's office
Financial Aid	One from IT, two from Financial Aid

oriented. It was key to keeping this implementation on track in terms of schedule and budget.

The implementers underestimated the new system's impact on the campus. Future implementers can help reduce stress associated with the amount of change by sending key individuals from all pertinent offices to change management classes. The abruptness of the change is a by-product of the direct-cutover implementation approach, and implementers must account for its effects.

Functional users also had to think outside the box. Whereas the old system's users concentrated primarily on their own department's activities, an enterprise-wide system required them to consider how their work affects other offices. For example, much of the data entered during student admission is used later in enrollment, grading, and reporting. If an address is entered inaccurately or a unique ID number is omitted from a student's record, a report or mailing will later reflect this error. The campus IT staff wrote many exception edits to catch errors early in the data entry process. For example, one such edit ensures that all admitted students have a complete and reasonable address. Another edit verifies that students with a transfer indicator always have transfer GPA and credit data entered in the system. A very important edit, run periodically, reports on potential duplicate entries in the system. Prospective student records are especially prone to duplication because prospect data is often taken over the phone, and names and addresses can easily be misheard.

The training schedule was right on track in some cases, and called for premature training in others. The compromise was to send users to training no earlier than one month before they would start using the product. It was important to at least start the setup of their particular module upon their return to ensure retention of the knowledge learned in class.

Another major lesson was the need to focus on migrating a smaller set of data. IT staff and functional users chose approximately a 20-year window of student activity for detailed migration. A better approach would have been to migrate only 10 years' worth of student and course data. A smaller set of data would have decreased the likelihood of introducing incorrect or incomplete data in the new system—garbage in, garbage out—to better safeguard system performance.

This implementation placed a tremendous load on the IT staff, as well as on functional users. Unloading as many existing duties as possible from key individuals so they could work with the new system on a full-time basis was crucial. Also important was finding quiet areas where the implementation teams could work on the new system with few interruptions. Table 2 gives the compositions of these teams.

Making very few changes to the vendor's SA system design and accepting some areas of functionality that did not work as planned was the right decision. The vendor later corrected most of these deficiencies and provided the appropriate patch to implement the fixes. The IT staff and functional users must now review every product change whenever they apply a minor patch or a major upgrade to the system. They must implement the delivered component, thoroughly analyze it to learn if its underlying logic has changed, and reapply and document any prior modifications.

Functional users had to accept several changes to their business practices. In retrospect, the implementers should have discovered them earlier in the implementation process and should have planned for them better. In any event, many changes to the business practices were drastic. For example, the implementers had to restrict the campus' highly flexible billing rules to fit the SA mold. Because the product represents the best business practices of several other (typically large) schools, future implementers can expect significant changes to the way they do business and should plan for them well before the implementation starts.

Like all ERP systems, the SA product is complex. To implement it well, IT staff and functional users had to learn many of the intricacies of delivered screens and processes. Many implementers engaged in long-term arrangements with consultants to help with the implementation. Such arrangements generally were not cost-effective for the campus. Functional users and IT staff typically preferred to implement those pieces of the system that were straightforward and invite consultants for short visits to help solve more complicated problems. This approach helped exercise better control over the implementation and retain much of the knowledge gained while solving problems. It also let the project stay on schedule and helped minimize consulting costs.

ERP projects are notorious for staff turnover. Even

though the management strived to retain all the implementers, one key systems programmer and one financial aid specialist resigned during the project. Future implementers should review and improve staff retention practices so as to retain staff throughout the project. Cross-training is another key to ensuring that the project does not fail when a key implementer leaves.

Research on ERP implementations in universities is scant. Yet, such implementations involve mission-critical systems, and their failure can be devastating to a school. Further research is needed to determine the critical success factors of ERP implementations in higher education and whether such projects have a higher risk of failure than those in industry.

Another important future research issue is how to modify existing system functionality or add new features without affecting the delivered product, while enabling rapid upgrades. One example is replacing a deficient screen to add missing functionality or replace deficient functionality. Another example is creating a custom-written report to provide a particular set of data in a certain order.

This implementation of the SA product was challenging but also rewarding. The campus now has a better system that provides accurate integrated reporting from many functional areas of the university. Most student, advisor, and instructor services are now Web enabled. Most third-party systems have now been replaced by the new system, allowing internal support and better response time. ■

Ilya V. Yakovlev *is a senior programmer and analyst in the Administrative Information Systems Department at the University of Wisconsin-Superior. He is also pursuing his PhD in the School of Computer and Information Sciences at Nova Southeastern University. Contact him at ilya@acm.org.*

Mark L. Anderson *is chief information officer at the University of Wisconsin-Superior. He also directs the Administrative Information Systems Department. Contact him at manderso@facstaff.uwsuper.edu.*

This implementation's success is due to the hard work and dedication of the implementers and was made possible by excellent support from the university's administration and from the University of Wisconsin System.

For further information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.