"Buy" Alternative

Background

The "Buy" alternative encompasses the purchase and deployment of a vendor student information system incorporating a relational database and web / portal interface. The SIS Planning Group held full day sessions with PeopleSoft and SCT, the two largest providers of "enterprise" student systems. The Planning Group also visited California State University, Los Angeles, an early adopter of the PeopleSoft student system.

Overview of Findings

In general, both enterprise software solutions offer very broad levels of functionality that provide most of the current capabilities contained in UCR's existing student system. Positive observations are as follows:

1. Both enterprise solutions provide for large amounts of self-service functionality, including registration, access to grades, degree audits, access to financial aid information, etc. Specialized self-service applications include a web based "transfer student articulation" module.

2. Both enterprise solutions provide functionality that is currently NOT deployed in UCR's existing SIS. This includes a "wait list capability", a student advising module, a financial aid repackaging module, on-line access to transcript processing, etc.

3. Both enterprise solutions provide for increased business / functional user "control" of key system processes. Examples include table / user driven control of course prerequisites and table / user driven control of auto-admission rules.

"Make" Alternative

Background

Adopting the "Make" alternative would result in C&C, the VCSA, and the VCA continuing the process of developing and customizing UCR's existing SIS+ system to meet campus needs. During 1998 and 1999, UCLA decided to pursue the "make" alternative. UCLA has a technical environment very similar to UCR's and the SIS Planning Group therefore conducted two "fact finding" conference calls with UCLA's technical leadership.

Overview of Findings

In general, UCLA feels (as does UCR) that the core SIS+ system provides more than adequate transaction processing support, as well as robust core Admissions, Financial Aid, Student Records, and Student Billing support.

However, in the late 1990s, UCLA faced challenges mirroring those currently facing UCR. Specifically, developing a web infrastructure that would meet student access needs, improving campus access to student system data, and providing for more rapid development of new student system functionality.

Proceeding with a "make" decision, UCLA's central IT staff created "middle ware" to allow web access to core SIS+ data and functionality. UCLA then re-wrote all the transaction processing web pages (heretofore accessed via Edify) utilizing the new "middle ware" capabilities. The College of Letters and Science developed the myUCLA portal using the "middle ware" to supply students with course enrollment data.

UCLA has funded a half decade long project to migrate SIS+ data to a relational database and to create a web based data warehouse for campus student information.
Technical Observations

1. Both systems make use of modern, relational databases and both are available on the Oracle platform.

2. "Out of the box", both systems provide scripts that automatically populate data warehouses from the transaction processing database (as a note, this functionality took one year of custom development in campus financial system).

3. Both systems provide relatively easy to use query tools for the casual user, and both systems have partnerships with firms like Brio and Cognos that provide additional query tools.

4. Both systems provide "out of the box" portal solutions that integrate well with their native applications and with Blackboard (the campus learning management system).

5. Both systems have industry standard solutions for "single sign on" via a portal and both have solutions for communicating with the campus LDAP authentication system.

6. Both systems have processes that "expose" the code / SQL behind business rules, edits, validation, etc. Both systems have a vision for incorporating XML into data transfer. These two capabilities greatly facilitate incorporation of external data into the system. This also facilitates the development of custom interfaces (web pages etc.) that incorporate native business rules, edits, etc.

UCLA has a very strong desire to "remain in control of its own destiny" relating to its student information system and thus pursued the "make" alternative. The following notes relate to this decision:

1. UCLA has a large "central" programming staff supporting SIS+ production, development, and web processing.

2. UCLA's central IT staff has received substantial funding to migrate from VSAM (non-relational database data storage mechanism) to a relational database. The funding is $8.4 million and the project is scheduled to be completed in 2006.

3. UCLA's central IT staff has received substantial funding to create a campus student system data warehouse. The funding is $1.4 million and a timeline for the project has not yet been established.

4. UCLA's technical infrastructure is VERY similar to UCR's existing technical environment. Almost all campus employees accessing the system use IBM "green screens", the system runs under the VSAM file system on an IBM mainframe, and there is currently no data warehouse available to the campus. The primary difference between UCR and UCLA is that UCLA has removed the Edify web infrastructure and replaced it with a more flexible system (utilizing the aforementioned "middleware" developed using Microsoft's COM architecture).
"Buy" Alternative

Customizations

In general, both systems have functional components that will not meet UCR's needs. Examples include "out of the box" class wait list functionality and navigation of some screens for rapid data entry (especially relating to Student Business Services payment processing).

The challenge relating to customizations is to perform them in such a manner that system upgrades do not become impossible (or extraordinarily expensive or lengthy).

In general, this implies that the customization process for an enterprise solution should follow the following guidelines:

A. Every attempt should be made to use the product as supplied by the vendor.

B. If a portion of the software can't be used "as is", every attempt should be made to reengineer the business process in question.

C. If a customization is needed, it should be done in a manner that does not impact native tables and uses objects that are already defined within the existing system framework (e.g. the customization should call the native, open code of the enterprise system for edit checks and validations).

D. If a customization to the core system is needed, every attempt should be made to have the vendor supply this customization and integrate it into the native application.

"Make" Alternative

Customizations

If UCR were to pursue the "make" alternative, it would develop NEW core functionality in various areas and it would continue to provide support for existing custom functionality. Neither UCLA nor UCR contracts with SCT for SIS+ maintenance or product support. Examples of new and existing functionality to be supported include the following:

Student Advising, Articulation, Auto Admit Process, Class Wait List, and increased web functionality in all areas of student interactions with SIS+.

Adopting the "make" alternative assumes that campus users will continue to access the system via IBM 3270 "green screens".

All customizations would be accomplished via COBOL programming without (certainly in the short term) any new tools to aid in RAD (rapid application development).

In order to support the "make" alternative, staff would have to be added to support the development, production, and web environments. UCLA's staffing levels are as follows:

Production: 18 FTE (Admissions 4 FTE, Financial Aid 6 FTE, Student Records 5 FTE, and Student Business Services 3 FTE).

Development: 5 FTE

Web Support: 4 FTE

TOTAL: 27 FTE (UCR Currently has 5 FTE in similar positions)

Under the "make" alternative, UCR would focus its immediate efforts on removing the Edify web access infrastructure and on developing a SIS data warehouse.