



**Final Report
Applications Software
Strategic Direction Assessment**

Yavapai Community College

Dickmeyer Consulting, LLC

*Request for Proposal #235, Software Consultant
June 29, 2006*

Nathan Dickmeyer, PhD, President
Fleur Eshghi, EdD, Executive Director
Dickmeyer Consulting, LLC
278 South Boulevard
Upper Grandview, NY 10960
(845) 358-5527
cell: (914) 441-7630
nate@dickmeyerconsulting.com
fleur@dickmeyerconsulting.com



Contents

Recommendations.....	Page 1
Background.....	Page 4
Recommendation Support.....	Page 9
RFP Required Elements.....	Page 19
Overview Cost Analysis.....	Page 26
Summary.....	Page 29
Appendix One: Definitions and Clarifications.....	Page 30
Appendix Two: Interview Findings.....	Page 33
Appendix Three: Online Survey Analysis.....	Page 37
Appendix Four: Design for a Demonstration of Functionality for Administrative Software Packages.....	Page 40



Recommendations

The rationale for each of these recommendations is given in the “Recommendation Support” section of this report. Further rationale can be found in the section that responds directly to the requests in the RFP. (Appendix one contains definitions and clarifications and should be perused before reading the body of this report.)

Enterprise Resource Planning (ERP)

1. End further development on the legacy system.
2. Issue a Request for Proposal to college enterprise system vendors, including Campus Management.
3. Accept the Campus Management proposal if the software price is \$350,000 or less for the student, finance/HR/payroll, alumni/development and housing modules.
4. Negotiate the Campus Management implementation price to \$530,000 or less by shifting or adding College resources to the implementation project, including these full-time dedicated positions:
 - a. Project manager
 - b. Two business analysts
 - c. Quality assurance administrator
 - d. Database administrator
 - e. Applications administrator
 - f. Technologies specialist and/or helpdesk analyst

In addition, a Web designer and a systems administrator should be assigned part-time to the job.

5. Replace the PeopleSoft HR system as part of the implementation.
6. The College should not purchase the Campus Management portals. The Blackboard portal can more easily be configured to serve students. (Nevertheless, vendor portals should be evaluated.)
7. Institute a Users’ Advisory Group, headed by a business analyst and staffed by the project manager, quality assurance administrator, the second business analyst, and applications administrator, to make procedural decisions during the implementation (and thereafter). The UA Group should include representatives from records and registration, admissions, accounting, accounts payable, academic administrative assistants, development, student billing, the Provost’s office, the Foundation, and housing.
8. Institute an Administrative Systems Policy Group consisting at least of the chief academic officer, the chief business officer, the chief student services officer, and the chief information officer. This group will make policy level decisions brought to it by the Users’ Advisory Group.
9. As part of the implementation, the College should examine and challenge every existing business rule defined within the administrative system following these guidelines:
 - a. Do not modify any code in Campus Management to suit what appears to be a special practice.
 - b. Design each business rule to tear down barriers to enrollment growth.
 - c. Assign maximum responsibility for decision making to faculty, advisors and students. Do not build control systems. Build systems that allow easy correction.

- d. Take the time to test the intuitive feel and communication of business rule design decisions. Do students understand the message? Can an ordinary user figure out what to do next? Maximize communication by using all available channels.
10. Make third-parties take more responsibility for providing service and record keeping (with electronic information exchange), or change the relationship, especially vendors like the bookstore, the bank, benefit providers, and Blackboard.

Organization

11. Change the way ITS (Information Technology Services) is organized to reduce the depth of the organization and to move from a specialty-based organization to a responsibility-based organization.
- a. Form a unified ITS division of User Support and Training.
 - i. Include in this division TELS (Technology Enhanced Learning Services), ITV (Instructional Television) support, PTSS (Presentation Technology Support Services), Help Desk and FRC (Faculty Resource Center—planned).
 - b. Form a second ITS division of Systems Management and Implementation.
 - i. Take advantage of the informal teams that now form for projects and larger challenges across technical specialty lines.
12. The President should appoint a Web Advisory Board. The board should be a permanent decision-making body, consisting of faculty members and administrators with representation from ITS.
- a. A Web Advisory Board should be charged by the president to take leadership in determining Web structure, control and ownership, policies and procedures, directions, and communications. (Our recommendations for each of these issues are given in the Web portion of these recommendations.)
 - b. Decisions, directions, changes, procedures and policies must be more strongly communicated to users. The advisory board must determine the information channels and the type of information to be communicated to users.
13. All IT advisory groups, including the Implementation Users' Group and the Web Advisory Board should be strengthened and given a strong charge by the president.

Training

14. Create a single Training unit within the ITS division of User Support and Training to support faculty, staff and students in gaining technological proficiency.
- a. Members of this unit should be skilled trainers first, technology experts second.
 - b. Members of this unit should be able to present training suited to specific user needs across a broad spectrum of technologies and software. Understanding the needs of users is more useful for trainers than deep product expertise.
 - c. Training should be tailored to user needs.
 - d. Trainers should follow up with users to provide additional help and to provide guidance to the trainers on designs for new training sessions.
 - e. There should be one training calendar.

Web

15. Web structure: The College website should follow an agreed upon unified look and design throughout, while maintaining the unique identities of each department.
 - a. Pages directly linked to the College home page should follow the home page design more closely than more distantly linked pages. First clicks should not startle.
 - b. All linked pages within the site should have at least one uniform navigation bar with an identical logo-link back to the homepage.
16. Ownership and control: Each page or group of pages should have an identified owner with a given level of control.
17. Production and maintenance: The initial production of Web pages should be outsourced. The producer should use agreed-upon Web authoring tools, compatible with the Yavapai Web environment and support structure within the unified design structure approved by the Web Advisory Board.
 - a. After initial construction of the site, individual departments will be responsible for the upkeep of their pages.
 - b. Upkeep should require minimal technical skills.
 - c. Departmental staff responsible for maintaining pages should receive training.
18. Content Management. Content XML—Cascade is a good solution for assisting departments in maintaining Web pages and should remain the standard. The College must engage in a more effective training program presented by the Yavapai training staff as recommended above.
19. The College should use the e-Portfolio capabilities already purchased with Blackboard. Faculty will find that this technology can assist them in creating personal Web pages. The technology should also be used as a tool for student development assessment.
20. The College should use the Blackboard portal system to help the navigational structure of the Website.
21. The College should use the Blackboard Content System in conjunction with the library's ERes Docutec. Each serves a distinct function and can be used to complement the other.
22. The College should plan to keep the Interactive Television (ITV) structure in service for another decade. New investment, however, should be limited to improving multicasting capability. The College should monitor the progress made in the development of online video streaming. We predict that this technology will become powerful enough to supplant ITV within the decade.
23. At the present time the College should not adopt strong standards for online courses.

Background

Yavapai College

As presented in the RFP that initiated this project: “The mission of Yavapai Community College is to provide affordable, quality learning opportunities to all people living within the county’s 8,000 square mile service area. Founded in 1967, the College currently enrolls over 14,800 students in credit and non-credit classes at numerous sites throughout Yavapai County. Although transfer and career training programs remain the College’s emphasis, many other services, from the Leadership Center to Small Business assistance, are offered. Officials at the Small Business Development Center serve an average of 450 businesses annually. Yavapai College Libraries provide service to 1,326 people daily, and 58,000 attend events at the Performance Hall each year. Over 8,000 people participate in Yavapai College Senior Programs.”

The county population will quickly grow past its current level of 170,000. The county is nearly as large as the entire state of Massachusetts, although sparsely populated in many regions. The full-time equivalent fall degree-seeking cohort numbers approximately 2,600. The College runs on a \$66 million budget, has 92 full-time faculty, and 285 full-time staff. 367 part-time faculty members are also employed.

Besides the full gamut of typical community college certificate and associate degree programs, the College has national leadership programs among community colleges in agribusiness, and film and video production, among others. It is also beginning to build programs integrated with the state’s universities in health science.

Applications (Administrative) Software History

Several decisions that were intended to change Yavapai College’s applications software strategy were either reversed or reduced in scope. The underlying strategy has evolved over time, but much of the basic approach remains. In the late 1980s the College began developing a unified approach to applications that ran in the COBOL/OpenVMS environment on (at that time) Digital Equipment Corporation VAX Alpha hardware. The two basic applications software modules were (from the RFP):

“**AMIS** - Admissions, Registration, Student Financial, Financial Aid, Curriculum Development, Course Catalog Development, Textbook Management, Advising, Housing, Room Scheduling, Grade processing, Transcripts processing and Facilities Tracking.”

“**Financial Systems** - General Ledger, Accounts Payable, Student Accounts Receivable, Fixed Asset Management, Budget and Bank File Maintenance & Reconciliation.”

In 1998, the institution decided that operating advantages could be gained by purchasing fully integrated applications software from a vendor with an existing base of community college clients. The software was the CARS system from Quodata, and an enticing number of function improvements made the purchase inviting. Unfortunately, after partial installation, many of the promised improvements had not been released by the vendor, and the College cancelled the contract.

The College continued to search for a feature-rich administrative system and selected the “best-of-breed,” PeopleSoft, as vendor. Again from the RFP, “Yavapai College completed the implementation of the PeopleSoft Human Resources/Payroll subsystem and fundamental components of PeopleSoft HR/Payroll/integrated student system 7.6 in July 2000. In 2004, the College de-coupled from student and converted to the Government version of 8.8 for HR-Payroll.” A 2003 study by the Gartner group showed that beyond the completed purchase of software, maintenance costs, the assistance of consultants in implementing the full suite of software modules and staffing necessary for installing and running the software would cost the College a further \$2 to \$4 million.

During much of this time, the College continued to upgrade the capabilities of its COBOL/VMS software. In the initial stages of the project, the College’s developers, led by Rick Giardini, invented procedures that presaged the object-oriented programming of today. Specifically, they systematized into a modular format input and output transactions from the database. As a result, the team produced a number of Web services not usually found in COBOL/VMS environments. The RFP describes these as:

“**Electronic Student Services** - Web Admissions, Web Registration, e-Cashier Payment Processing, Online Grading, Transcript Generation/Order, Attendance Accounting, Course Search/Lookup, Housing Application, Scholarship Application and Financial Aid Status Lookup.”

In addition, as pointed out by the RFP, there are “eighteen departmental applications ranging from a large Oracle-based Transaction system (e.g. Blackboard Transaction System) to small single server or single workstation applications (e.g. AdvisorTrac, Telephone Billing)” coupled with the legacy system to varying degrees.

After the decision to halt PeopleSoft implementation, a more efficient method of upgrading the legacy system was sought. If properly chosen, this method could lead to a full shift off of the VAX hardware. As the Digital Equipment Corporation has been chewed up in acquisitions and blurred by corporate mergers, the commitment to support the hardware by each owner of the VAX product line has shifted. While the urgency to move off of the VAX equipment and write a new system has gone up and down, the institution never formally committed to a project to develop replacement software in-house.

Under the leadership of Rick Giardini, Information Technology Services (ITS) began a search for an appropriate database technology, development environment and development programming language. After much research ITS chose to experiment with the Microsoft SQL Server database, to develop within a Microsoft .NET environment, and to use one of the newer languages for this environment, C# (C-sharp).

To make the experiment meaningful, they chose to mirror exactly the COBOL/VMS database in SQL Server and develop applications using real data (in a development environment outside of the active, operating environment). The first major process they chose to explore was “course selection.” This is the search an online user would use to find and select courses for registration.

Sufficient resources for the experiment and “proof of concept” were allocated to ITS, but these resources were not at a level that could sustain a project to replace the COBOL/VMS system in a reasonable amount of time. Developers occasionally became involved in patching the COBOL/VMS system in emergency situations. Building sufficient framework (security and database entry modules, for example) pushed the length of the experiment to three years before the course selection section was implemented.

Quite recently, the leadership of the College turned over. The president, chief financial officer and provost are new in their positions. This new administration realized that the College had to set its applications software strategic direction with firmness. The choices facing the College were to:

1. Continue upgrading the legacy system in a Microsoft .Net environment;
2. Move to a full conversion project in the Microsoft .Net environment;
3. Re-start the PeopleSoft installation; or
4. Purchase another full suite of applications software.

Dickmeyer Consulting, LLC was hired to assist the College in making this choice in a way that registered the firmness of the new administration’s determination to move ahead.

Project Methodology

We began the project by requesting and reviewing a number of documents available from the College and from the College website. These included service surveys, College and departmental goals and strategic plans, departmental strength, weakness, opportunity, and threat analyses, accrediting self-studies and focused reports, organizational charts, job descriptions, and departmental budgets.

Nearly two dozen short telephone interviews were conducted to search for issues, system strengths and weaknesses, and Yavapai College faculty and staff ideas on the needs for existing function updates and the addition of new functions. Respondents were called at the Prescott, Verde Valley, Prescott Valley, Chino Valley, and Sedona locations.

The first site visit was conducted from February 6, 2006 through February 10, 2006. The visit allowed interviews with 13 groups of from three to fifteen people and meetings with 13 individuals. Over 100 people contributed during this phase of the study. In addition, several open forums were held on the Prescott campus with an ITV link to the Verde campus. Participants in the interviews included staff and faculty, and system users as well as ITS staff. Most meetings focused on the functionality of the current administrative computing system and the need for new system capabilities. Appendix two lists the function requests as they were developed for the website.

The intent of the project and the current findings were maintained on a project website linked from the Yavapai College homepage.

The second site visit was conducted from March 27, 2006 through March 30, 2006. Six individual conversations were scheduled as well as meetings with eight groups, consisting of 47

staff members and a Foundation board member. This visit focused on gathering information on current administrative computing efforts and establishing the necessary level of funding to set up and complete the .Net project. Members of the administration were asked about their perceptions of the value of high functionality and the costs of system instability. Many of the meetings also allowed individuals to add to the functional needs listed. The project website was updated with this information.

The final day of the visit included sessions with software vendors in Phoenix.

To further involve the Yavapai College faculty and staff in the process of choosing a direction for administrative computing and Web effectiveness, notice was given of the availability of an online questionnaire. The survey asked respondents about faculty and staff perception of administrative computing and website development priorities. The results of the survey are given in Appendix three. One hundred faculty members and two hundred staff members answered the survey online. Although students were invited to participate, only four filled out the survey.

At this stage we requested cost estimates from vendors for free-standing financial and human resources modules. Software, implementation and maintenance cost estimates were also requested from several integrated college and university administrative system vendors for the entire enterprise system (for an institution the size and complexity of Yavapai), but with each module priced separately.

ITS leaders and consultants were contacted regarding new software developments for community colleges and other development projects. Barbara Viniar, Executive Director at the Institute for Community College Leadership at Cornell connected me with community college information technology officers that she relies on for technology information. Art Szu-tu at Occidental Community College (New York) did not know of a community college that was undertaking in-house ERP development work. Other contacts gave me the same information. At Foothill Community College (California) President Bernadine Chuck Fong told me of Foothill's move to join the Sakai open source instructional support project, despite in-house development of the ETUDE online course management system since 2001. Even after licensing ETUDE to 50 other colleges, they were not finding continued development financially supportable.

At NACUBO, Sue Menditto, Director, Accounting Policy, told me about NACUBO's support of the Kualu open source development of a university ERP with activity centered at Indiana University. She described progress with the finance system. Work on a student system had not begun. John G. Robinson at RSmart, consultants for open source installation headquartered in Phoenix, told me of the Kualu project's interest in testing the scalability of Kualu with community colleges. San Joaquin Delta Community College was then (and still is) the only community college consortium member. I did get the impression in discussions with him and with other observers of the project that, while the open source software would be free, consulting costs for implementation would be comparable to those necessary for major ERP implementations. At the University of Texas at Austin, I talked with Dan Updegrave, Vice President for Information Technology. He said that UT was doing quite a lot of development, but using Natural Programming Languages and Environments, not C# and .Net.

Using the costs estimates from the College and from vendors (see table one below) and using the ideas from the College’s leadership about the cost of system downtime and the value of high functionality, we employed decision science techniques to assess the range of options open to the College. The value of each of the options was based on the costs, estimated probability of down time, and estimated probability of high or low functionality of each. We also tested the sensitivity of the options’ values to possible changes in cost estimates, downtime probabilities, and resulting function-level probabilities. The decision tree is attached as a separate Microsoft Excel worksheet.

Table One: Estimated ERP Costs from Vendors and Consultants

Vendor/system	Cost estimates	Modules included
Gartner/for PeopleSoft	\$2.3 million to \$4.1 million	Financials and student system for implementation
SunGard/SCT/Banner	\$1,100,000 software 1,550,000 implementation 200,000 annual maint.	Student, Finance, Oracle DB
SunGard/SCT/Banner	\$2,500,000 software 2,500,000 implementation 500,000 annual maint.	Student, Finance, Oracle DB, HR, Portal/eLearning, Financial Aid, Info. Access
SAP/Osprey—Finance	\$200,000 software 200,000 implementation 40,000 annual maint.	Finance only

The options and values derived from the costs, probability estimates, and functionality value estimates were shared with the College in a series of interviews conducted from April 24, 2006 through May 5, 2006. Four locations were visited. We met with eleven groups totaling fifty-three individuals and individually with thirteen people. During these meetings we showed how the choice had narrowed to: 1) continue the .Net project, purchasing financial software for around \$200,000 or less, and keeping HR running on PeopleSoft, or 2) purchasing Campus Management enterprise software with the option of replacing the PeopleSoft HR module. The decision logic showed how the choice must be based on more than financial grounds. The choice had to be made based on the functionality of the Campus Management software and the failure risk inherent in either option.

During the week from May 1, 2006 to May 5, 2006 Dr. Fleur Eshghi visited the College to assess (from the RFP) “Web direction in relation to online classes, online registration, and functionality to formulate recommendations.” We met with six groups totaling 19 individuals and singly with 16 individuals at five locations. These interviewees also consisted of faculty and staff, including users and ITS staff.

Following the final visit, using the functionality needs specified by College users, we undertook a preliminary assessment of the functionality level of the Campus Management software. We also contacted current users of the software.

Recommendation Support

Enterprise Resource Planning (ERP)

1. *End further development on the legacy system.*
2. *Issue a Request for Proposal to college enterprise system vendors, including Campus Management.*
3. *Accept the Campus Management proposal if the software price is \$350,000 or less for the student, finance/HR/payroll, alumni/development and housing modules.*

The cost difference between a lease/purchase of the Campus Management (CM) system and Yavapai itself building an enterprise system using .Net technology over five years is not sufficiently large that a choice may be made on finance alone (assuming either a five or seven-year lease for the CM software and installation and assuming that each project could make use of certain funds supporting the legacy system, see Table Seven, page 25, in the Overview Cost section below). The cost estimate from CM for the CampusVue Collegiate software, based on 2,611 FTE enrollment, included software licensing fees for these modules: Student Information System (\$210,000), Financial/HR/Payroll (\$80,000), Alumni Development (\$35,000), and Housing (\$20,000). The Web Portals module (\$70,000) is not included in this estimate. The choice of the CM system rests on functional and risk assessments, as follows:

- A) The Campus Management system appears to have the potential to satisfy the requirements of the College and the procurement process. The hallmark of the system is its high degree of flexibility.
 - B) The Campus Management system is one of the few built on the same modern platform as the internal project. This platform was developed specifically for Web-mediated operation.
 - C) The College cannot wait for the end of an internal development process to present true Web services to students, faculty, and staff.
 - D) Turnover in new IT professionals has been high. An internal project requires consistent staffing.
 - E) The search for qualified people for the project would be very challenging, as it is for faculty in the area. Housing prices make it difficult to attract qualified people.
 - F) The project's success rests on one person.
4. *Negotiate the Campus Management implementation price to \$530,000 or less by shifting or adding College resources to the implementation project, including these full-time dedicated positions:*
 - a. *Project manager*
 - b. *Two business analysts*
 - c. *Quality assurance administrator*
 - d. *Database administrator*
 - e. *Applications administrator*
 - f. *Technologies specialist and/or helpdesk analyst**In addition, a Web designer and a systems administrator should be assigned part-time to the job.*

Implementations of ERPs are complex and varied. The response to the request for proposal will form the basis of the contract. This consultant can only speculate what the terms will be, but I

will list some services that the College might expect to see in the contract. This list is not complete.

The vendor must define the required data structures and should build data conversion routines. The vendor should describe “best fit” (for their software) business practices and assist in the design of the business practice analysis. The College should expect a close match between “best fit” and industry “best practice.” Other Colleges, including a community college running on the CM student system, say that one of the major benefits of an ERP conversion is changing business practices toward best practice.

The vendor will install the software in partnership with the College. They should assist in the set up of development, test and production instances.

They will configure the software, but they should not be expected to recode it. Configuration means making the software work for the College (unless recoding would be required). It means getting the labels and choices right. It means making the attributes of students, employees, accounts and courses, for example, Yavapai-specific. Many of these attributes will need further delineation with constraints like filters, including minimums, maximums and legal sets. To do the configuration, the vendor will need to understand the College very well.

They will also set specifications for third-party software interfaces, but the College should expect that the vendor will suggest third-party software for which they have already built interfaces. Finally, they will design and perform training.

The Yavapai implementation staff and users will be busy. The College must deconstruct its business practices. Purchasing provides a good example for business rule deconstruction: Who approves? Who gets notified? How is budget over expenditure handled? The College must then weigh the merits and disadvantages of the reconfiguration options that work within the software.

The College will need to optimize database performance during its construction. It will be required to manage web deployment, including client configuration. Cleaning existing data so that the data fits within the vendor’s structure will also be required. (Having already moved legacy system data to SQL/Server will be a major help.) Conversion is also the time when the College may find that it needs to expand categories, requiring a clever selection among or even hand recoding of existing identical entries.

Much of the work of the system will require the College to define cohorts, for example, defining those who should be billed at a point in time. These record-set definitions are used to generate reports and to initiate actions (like bills). The College’s business practices will determine these definitions. My guess is that defining and setting up these cohorts will be one of the most challenging and tedious tasks College staff will encounter.

While many reports come with the software, many will require refining. Reporting flexibility is both a blessing and a curse. It will be good that the College has the flexibility, but the opportunity can also increase workload. CM software integrates Crystal Reports. My experience

tells me that some users will find that they will master building their own reports, but many will not.

ITS staff will also need to be trained in maintaining and reconfiguring the software. They will also be required to train along with users to reinforce their practical knowledge of the software.

This workload will require seven to eight people to manage the implementation with full focus and two with partial focus. In addition, the time of users will be greatly imposed upon. We believe that the College has sufficient resources to assign people to this project without increasing the number of ITS staff, but only if the Legacy system and PeopleSoft HR could be left unattended. The additional costs (approximately two people plus licensing fees) during the transition to keep the current systems running are factored into the cost estimates for the CM implementation in the "RFP Required Elements" section of this report below. The legacy savings in the cost estimates are decreased during transitions by the support needed to continue production on the legacy and HR systems.

Two programmers are currently working on the .Net effort and two programmers are focused on the PeopleSoft HR system. In addition, we believe that a project manager, database administrator, a technologies specialist and/or helpdesk analyst, and the part-timers could be reassigned to this project. We believe that many of these people will be required for the operation of the system after completion of implementation. The College is continually searching for better ways to provide service. This search results in new projects directly requiring management and business analysis. (Nevertheless, ongoing support levels for a CM installation at another community college were lower than this.) The tables show that 6.25 people currently support legacy and HR (not including help desk and other user support people). CM implementation will require six people (plus the conversion of a Technologies Specialist or Help Desk Analyst). During transition, with two individuals needed to maintain legacy and HR, the 6.25 will have to grow to eight. After transition, four people will be required (plus the converted user-support person).

5. Replace the PeopleSoft HR system as part of the implementation

Although we dislike imposing a new implementation when a system appears complaint free, the resources now used by the PeopleSoft system must be conserved. The system is too powerful for the College. Its approximately seventy tables are interdependent, requiring broad changes for single problems. Unless the College replaces the HR system, it will have to hire more people for the Campus Management implementation. This is a cost of the HR system, not of the Campus Management implementation.

The College needs to adopt systems where many people can share knowledge. Having several very different platforms running, especially one as esoteric as PeopleSoft, requires specialists. The lack of flexibility is a drain on people and increases the risk of downtime.

6. The College should not purchase the Campus Management portals. The Blackboard portal can be more easily configured to serve students. (Nevertheless, vendor portals should be evaluated.)

The interface between Blackboard and the legacy system has been established. The College would need to reestablish that interface in any event with the new system. With the interface Blackboard has access to information in all the tables. The closer link between the Blackboard portal and Blackboard classroom information motivates this recommendation.

7. *Institute a Users' Advisory Group, headed by a business analyst and staffed by the project manager, quality assurance administrator, the second business analyst, and applications administrator, to make procedural decisions during the implementation (and thereafter). The UA Group should include representatives from records and registration, admissions, accounting, accounts payable, academic administrative assistants, development, student billing, the Provost's office, the Foundation, and housing.*

A major challenge of system implementation is getting design and business rule decisions quickly. This group must meet at least once a week to advise the implementation team on set-up issues that cross office lines. All aspects of registration, including billing, need to be debated by a broad group. The role of the business analyst is to frame business rule set-up questions in ways that bridge the gap between the capabilities of the software and the needs of the College. There should be no changes to the software. The software should have the flexibility to accommodate the best practice (but not necessarily the current practice). Billing rules, non-payment sanctions and responses, class waitlist procedures, and the text ordering process all need careful assessment.

The excessive length of most implementations often results from the lack of a smooth process for involving users in business rule set up and getting offices involved in testing and training. Implementations would be fast if any business rule would do and if extant data did not require recoding.

8. *Institute an Administrative Systems Policy (ASP) Group consisting at least of the chief academic officer, the chief business officer, the chief student services officer, and the chief information officer. This group will make policy level decisions brought to it by the Users' Advisory Group.*

This group should respond quickly to policy issues raised by the UA Group. Its makeup should reflect "the president's team." Ideally, it might meet with the Users' Advisory Group during the last 15 minutes or 30 minutes of a UA Group meeting. At the level of the ASP Group's purview, for example, is the 24 hour grace period after enrollment without payment policy. The exact sequence of follow-up steps on students with open bills probably is not. Many implementations stall while higher-level policy decisions wait in limbo. Users generally do not feel that they should make such decisions, but the next step up in the decision process is too often neither smooth nor obvious.

9. *As part of the implementation examine and challenge every existing business rule defined within the administrative system. Follow these rules:*
 - a. *Do not modify any code in Campus Management to suit what appears to be a special practice.*

Implementation teams must take the attitude that conflict between software capabilities and current business rules is an indication that the rules are not best practice, not that the software needs modification. Modifications are expensive. Modifications make upgrades very difficult,

sometimes impossible. Even if upgrades are possible, modifications make the new software, even patches, unpredictable. Software upgrades never arrive already tested with client modifications.

b. Design each business rule to tear down barriers to enrollment growth.

The financial stability of the College rests on enrollment growth. All rules should give students the benefit of the doubt. Rules should be designed to err in favor of the convenience of the student. Occasionally, we have heard on other campuses that stringent penalties for missed deadlines, for example, can be justified as teaching students about the real world. Those of us who have filed for income tax extensions know that the real world usually balances the good efficiency of a business rule, like deadlines, with the reality of revenue maximization.

c. Assign maximum responsibility for decision making to faculty, advisors and students. Do not build control systems. Build systems that allow easy correction.

The implementation of new administrative software is a good time to evaluate the College's attitude toward students, staff and faculty. While fatal mistakes should be guarded against, assuming (even without that conscious thought) that these people cannot make mature definitions builds resentment into the broad attitude toward the new software.

Faculty should be able to override class limits to admit students with special needs (like imminent graduation). Department administrative assistants should be able to override prerequisites for students with good cases. Staff should be able to change demographic and benefit data online. All this flexibility should be under the control of strong permission and security structures.

d. Take the time to test the intuitive feel and communication of business rule design decisions. Do students understand the message? Can an ordinary user figure out what to do next? Maximize communication by using all available channels.

Don't wait for the "grand rollout" to introduce people to the software. Let them see screens and comment on the ease of use. Take their suggestions back to the UA group to discuss and present to the software implementation specialists.

Put a discussion of new features on the Web for people to understand before they use the software. Tell people of progress and set-backs. Full involvement and communication are the keys to a successful implementation.

10. Make third-parties take more responsibility for providing service and record keeping (with electronic information exchange), or change the relationship, especially vendors like the bookstore, the bank, benefit providers, and Blackboard.

Administrative assistants are now doing much of the work of the bookstore. Faculty should communicate their needs with departmental sign-off. The bookstore should give feedback on new editions and availability.

The food service should upload its weekly menus so that campus community members could pull that information into a portal window if they so choose.

Benefit providers should provide links for annuity calculations.

Organization

11. *Change the way ITS (Information Technology Services) is organized to reduce the depth of the organization and to move from a specialty-based organization to a responsibility-based organization.*
 - a. *Form a unified ITS division of User Support and Training.*
 - i. *Include in this division TELS (Technology Enhanced Learning Services), ITV (Instructional Television) support, PTSS (Presentation Technology Support Services), Help Desk and FRC (Faculty Resource Center—planned).*
 - b. *Form a second ITS division of Systems Management and Implementation.*
 - i. *Take advantage of the informal teams that now form for projects and larger challenges across technical specialty lines.*

Many of the people interviewed from outside of ITS stated that they did not understand the structure. “Responsible parties” are not easily identified.

We recommend that the IT training structure remain within ITS, closely allied with other elements of user support. All aspects of technology, whether used in classrooms, on the Web, in administrative offices, or in the library, are interrelated. Users, especially academics, need to move between various systems, both academic and administrative, easily and seamlessly. Training programs must be seen as integrated and seamless as well.

We also saw the potential of greater synergy between TELS, ITV, PTSS, and the Help Desk. These areas work well together now, but we felt their coordination could be reinforced with organizational changes. Increased synergy can help extend limited resources.

12. *The President should appoint a Web Advisory Board. The board should be a permanent decision-making body, consisting of faculty members and administrators with representation from ITS.*

The Web Advisory Committee has been engaged in creating a structure and design for the Yavapai College Website that will be more dynamic, up-to-date, informative, and communicative. Currently, a complete Website redesign is being prepared. It will be launched later in the year. However, Website administration issues remain to be addressed. Resolution of these issues is crucial to the success of the development and ongoing maintenance of a Yavapai Web presence among all College audiences. The major issues are that responsibility for setting Web policy is currently diffuse and that when policies are set they are communicated with insufficient energy.

- a. *A Web Advisory Board should be charged by the president to take leadership in determining Web structure, control and ownership, policies and procedures, directions, and communications. (Our recommendations for each of these issues are given in the Web portion of these recommendations.)*

The current committee appears not to have a formal, permanent role in Web decision making. It appears to be working through and/or competing with several other agencies in order to gain

implementation of recommendations. The president needs to clarify the role and structure of a decision-making group that will make Web policy.

- b. Decisions, directions, changes, procedures and policies must be more strongly communicated to users. The advisory board must determine the information channel and the type of information to be communicated to users.*

There were troubling indications that users felt uniformed about Web policy and angry over unexplained decisions. Few interviewees knew who was responsible for what on the website. The rationale for using a content management system was not explained to users, nor was the choice of the Content XML – Cascade content management software. Faculty Web pages will be available soon, but this has not been communicated. Developers have been cautious about advertising that there is a project to redesign the website. Advisory committees have information responsibilities in two directions: advising in and communicating out.

- 13. All IT advisory groups, including the Implementation Users' Group and the Web Advisory Group should be strengthened and given a strong charge by the president.*

Advisory groups have a history of being formed and working reasonably well for a year or two. Then effectiveness wanes (or is undermined after a change in administration). This administration should renew its commitment to collegial governance.

Advisory groups should receive annual training in group function. A half-day workshop would be effective. Each advisory group should report to the president's team on its evolving strategy for success in the area. Each advisory group should analyze major policy level issues, giving options and strengths and weaknesses of options, for resolution by the president's team.

Advisory groups should refrain from day-to-day management of areas or even short-term priority setting. The advisory groups should work with area leaders to develop a definition of success within available resources for the end of each year. Within that timeframe it is the obligation of the area manager to set short-term priorities, methods of implementation, and work assignments. Requests for new functionality of any kind should be sent only to the advisory groups. The advisory group may wish to alter the end-of-year vision by replacing one new function development project with the newly suggested one. Advisory groups do not handle emergencies, and managers should not reorder priorities with every request from the field.

Advisory groups should not be concerned with emergency triage. We define an emergency as a loss of existing functionality. Emergency management should stay within the realm of managers.

Managers, however, need to do a better job of convincing advisory bodies that emergencies are being adequately handled. There are several good rules for dealing with emergencies. The degree of emergency should be judged by the closeness of the lost function to the core of the College's operations, the number of people affected, and the harm of the loss of functionality to those people. High level emergencies override work on user group-determined projects. Lower level emergencies should be assigned to standard blocks of time set aside for such work. Lower-level emergency work should not be allowed to leak into assigned project work time.

Training

14. *Create a single Training unit within the ITS division of User Support and Training to support faculty, staff and students in gaining technological proficiency.*
- Members of this unit should be skilled trainers first, technology experts second.*
 - Members of this unit should be able to present training suited to specific user needs across a broad spectrum of technologies and software. Understanding the needs of users is more useful for trainers than deep product expertise.*
 - Training should be tailored to user needs.*
 - Trainers should follow up with users to provide additional help and to provide guidance to the trainers on designs for new training sessions.*
 - There should be one training calendar.*

Yavapai can greatly improve the process of training, support and communication by creating one integrated and coordinated process for user training on all issues of IT. Trainers should integrate training on Blackboard, classroom technology, and the web. The work of the training area should be closely coordinated with TELS and PTSS and informed by the demands place on the help desk and ITV. Training for faculty should be supported by the developing Faculty Resource Center as part of the Center for Teaching Excellence.

Web

15. *Web structure: The College website should follow an agreed upon unified look and design throughout, while maintaining the unique identities of each department.*
- Pages directly linked to the College homepage should follow the homepage design more closely than more distantly linked pages. First clicks should not startle.*
 - All linked pages within the site should have at least one uniform navigation bar with an identical logo-link back to the homepage.*

The Yavapai Website, not unlike that of many other institutions, has grown organically with no particular plan, but based on the separate initiatives of interested individuals. The current website is made of hundreds of scattered pages, some with outdated information, some with broken links, some that are properly maintained, and some that are not.

The deployment of the content management system, Content XML, renamed Cascade, has done little to rectify the problems. Difficulty of use, lack of training, and lack of support are among the issues mentioned in most interviews.

Design continuity should be subtle, not iron-fisted. Consistency of navigation tools and masthead design (at least for initial layers) are important. Consistency of body page layout and color is much less important. Nevertheless, users should feel consistency between College divisions, usually directly linked to the homepage. Web users should not feel like they have entered another world on the first click. Departments, however, are entitled to designs representing their special characters. Even with departments, masthead and basic navigational cue design should be maintained.

16. *Ownership and control: Each page or group of pages should have an identified owner with a given level of control.*

The Web Advisory Board should manage ownership and control. Owners are responsible for maintaining content. Level of control issues include whether new designs need to be approved by

the board and whether content can be changed without third-party approval. For example, the person who owns strategic planning pages should not make changes unless requested to do so by a strategic planning executive committee.

17. *Production and maintenance: The initial production of Web pages should be outsourced. The producer should use agreed-upon Web authoring tools, compatible with the Yavapai Web environment and support structure within the unified design structure approved by the Web Advisory Board.*
- a. *After initial construction of the site, individual departments will be responsible for the upkeep of their pages.*
 - b. *Upkeep should require minimal technical skills.*
 - c. *Departmental staff responsible for maintaining pages should receive training.*

Initial production of Web pages might be done within ITS, an option that was attractive to many who were interviewed. The attitude of these individuals seemed to be that webpage creation was technically difficult and an unnecessary burden for them. ITS should do it for them. Nevertheless, ITS simply does not have enough resources to carry this out effectively. Currently, ITS has a backlog of projects that began in 2005.

An efficient compromise is to outsource initial page building and train page owners in how to maintain their pages using a content management system. Users have been turned off by a regretful training experience on software that had not yet matured. Trainers from ITS, whose specialty is to present material most relevant to Yavapai people, can produce a much better experience. The software is now more user-friendly. Nevertheless, initial construction is beyond the capability of most users. Specialists in webpage construction can do a much better job, while consuming fewer College resources.

18. *Content Management. Content XML—Cascade is a good solution for assisting departments in maintaining Web pages and should remain the standard. The College must engage in a more effective training program presented by the Yavapai training staff as recommended above.*

The introduction of the content management system has not been successful. The following are some of factors contributing to the lack of success:

- The rationale for and advantages of using a content management system were not communicated to users. Content management systems impose design limitations on users, and Content XML is no exception in that regard. User expectations were not managed.
- The budget determined the choice of Content XML. Again, users were not informed and their expectations were not met.
- Content XML – renamed Cascade – has been greatly improved over the past few years and has become more user friendly, but this fact has not been communicated to users and their perceptions have not changed.
- The initial training was not successful. It was taught by technicians rather than professional trainers. TELS professional trainers were not – and are not – involved.
- No further attempt has been made to restore training because of the shortage of staff.

This issue can be rectified by creating one seamless and incorporated training program staffed by the current professional trainers in TELS and PTSS. The Web Advisory Board must work to change current perceptions.

- 19. The College should use the e-Portfolio capabilities already purchased with Blackboard. Faculty will find that this technology can assist them in creating personal Web pages. The technology should also be used as a tool for student development assessment.*

A number of institutions have developed capability with e-Portfolio with the support of grants. Many of the positive changes that students at Yavapai undergo are not easily reproduced with pencil and paper. An e-Portfolio can be a way of tracking these changes. e-Portfolios from students who permit it may be viewed on the Web to demonstrate the developmental power of the College.

- 20. The College should use the Blackboard portal system to help the navigational structure of the Website.*

Portals allow Web entry to be tuned to the needs of various groups initially and then individually developed further by users. The difficult question of how to make one homepage friendly to potential students, current students, faculty, staff, community members and alumni is answered by the use of a portal system.

- 21. The College should use the Blackboard Content System in conjunction with the library's ERes Docutec. Each serves a distinct function and can be used to complement the other.*

Content in Blackboard can be made available to specific classes. The ERes Docutec system presents material to the general College community and the public. The College should encourage coordination, not competition.

- 22. The College should plan to keep the Interactive Television (ITV) structure in service for another decade. New investment, however, should be limited to improving multicasting capability. The College should monitor the progress made in the development of online video streaming. We predict that this technology will become powerful enough to supplant ITV within the decade.*

The shortage of faculty in several areas dictates that the College must use ITV to serve students in two locations with a single faculty member. Because the hardware has to be in place for this need, the College can effectively make use of it simply to make teaching more efficient. The College could not justify the cost of the system on efficiency alone, however.

The current video-streaming technology is not sufficiently advanced to achieve ITV's quality of presentation. Nevertheless, its quality and reliability is advancing steadily. This suggests that what is now achieved in a separate system may be integrated with online instruction systems, making maintenance and control more efficient.

- 23. At the present time the College should not adopt strong standards for online courses.*

The College's strategic intent is to encourage the use of technology in teaching. Setting a rigorous standard at this time may inhibit experimentation in the use of technology. The time for standards will come when the College moves into distance learning to a greater degree than it is now.

RFP Required Elements

1. Legacy system review

a. Cost of operation and maintenance

Because the legacy system is fully embedded in the infrastructure of the College, we cannot easily draw a line where legacy stops and all else begins. For the purposes of this analysis, we will define the legacy system as stopping at the VAX. We are not including shares of net server upkeep, ITS administration or help desk time. These costs elements will be common to all options in any case. The following are the budgeted costs of the core of the legacy system.

Table Two: Core Legacy Costs

i.	1 1/4 programmer analysts	\$ 69,793	
		15,947	(63,791/4)
ii.	7/8 data base administrator	54,839	(73,119*3/4)
		8,800	(70,400/8)
iii.	Infrastructure contracts	55,750	(750+7,000+48,000)
iv.	Operating system upgrades	10,000	
v.	Training	<u>4,000</u>	
		\$219,129	

Total annual VAX/VMS portion of legacy: \$220,000 (of which about \$80,000 would be needed during a moratorium transition).

Currently, the COBOL system is being upgraded within a Microsoft .Net framework. If we include these costs as part of the legacy system we would add:

Table Three: Legacy Upgrade Costs

i.	1 programmer analyst	\$66,560	
ii.	1 1/8 database administrators	18,280	(73,119/4)
		61,600	(70,400*7/8)
iii.	1/2 Director	46,249	(92,498/2)
iv.	Infrastructure contracts	25,000	(10,000 + 15,000)
v.	Training	<u>30,000</u>	
		\$247,689	

Total annual cost for the .Net effort to upgrade the legacy: \$250,000 (of which \$10,000 would be needed during a transition).

The PeopleSoft HR system should also be considered as part of the legacy system. The support costs for this system include:

Table Four: PeopleSoft HR Costs

i.	1 ½ programmer analysts	\$ 66,981	
		31,896	(63,791/2)
ii.	Maintenance	62,000	
iii.	Training	1,200	
		\$162,077	

Total annual PeopleSoft cost: \$160,000 (of which \$130,000 would be needed during a moratorium transition). (There is also an anticipated \$90,000 version upgrade cost for 2009.)

Total annual full legacy system cost: \$630,000 bumping up to \$720,000 in 2009 (of which \$220,000 would be required during a transition).

b. Functionality

Currently, users rate the functionality of the system very low (see Appendix three). The complexity of using a non-GUI interface to set up faculty, classes and order textbooks is a significant burden on academic administrative assistants. The ponderous nature of online class registration and the difficulty of presenting legible bills keep students from fully utilizing Web services. The lack of encumbrances forces all departments to maintain shadow budget systems. The lack of month-end closing flexibility makes presentation of financial reports fraught with uncertainty.

Nevertheless, the .Net framework upgrade path with the SQL/Server database has the potential of correcting all these difficulties. Unfortunately, at the current level of support and with the structure of an experimental upgrade process instead of a true replacement project, the realization of full functionality is not likely to happen before the June 30, 2009 deadline set by the Project Steering Committee for analysis purposes. Only by adding the resources and structure necessary to convert the experiment into a sanctioned project can the deadline be met.

c. End user satisfaction

The online survey asked respondents to rate each of 14 major functions of the legacy system. The 304 respondents gave out 1,137 function ratings, excluding those responses indicating a lack of familiarity with a function. Of these 1,137 function ratings, 42% were below acceptable. This includes the much higher ratings for the payroll and student financial aid functions (only 17% rated each of these functions as below acceptable). This level of dissatisfaction was reflected in the interviews. Appendix two summarizes the results of the interviews. Appendix three contains the results of the online survey.

d. Programming staff satisfaction

Programming staff agreed that legacy functionality had not kept up with demands because of the three-year period when Quodata and then PeopleSoft were expected to be replacements. They agree that the upgrade path using the .Net framework will eventually result in full functionality. This path will lead to a complete replacement of all COBOL programs and VMS database elements in the current system.

ITS staff members have made the VAX/VMS system more reliable than we have found at other installations with similar and also with more advanced operating systems. They have also added functionality to the VAX/VMS COBOL systems such that input/output functions can be plugged in like modules. Online capabilities are higher with this system than at many other VAX/VMS installations.

The programming staff members involved with the HR PeopleSoft system have praised its power and ultimate flexibility. These staff members recommend strongly that the College's involvement with PeopleSoft be expanded at least to include the PeopleSoft finance system.

Staff members also agreed that without sanctioning, direction and resources, the full transformation of the VAX/VMS system to the .Net framework was unlikely to conclude in a reasonable amount of time. They felt that their energies were being dissipated by having to work on emergencies with the current production system. They felt under-trained for .Net development work. They felt that the effort lacked consistent priorities and full project structure.

2. Review legacy alternatives

- a. .Net project
 - i. Cost and timeline

A full .Net project would require these elements:

Table Five: .Net Project Costs

		2007	2008	2009	2010	2011
1.	Project manager (3 years only)	\$92,498	\$ 92,498	\$ 92,498	\$ -	\$ -
2.	Two business analysts (1 for 3 yrs, 1 ongoing)	89,600	89,600	89,600	44,800	44,800
3.	One database administrator (DBA) (ongoing)	70,400	70,400	70,400	70,400	70,400
4.	One quality assurance administrator (ongoing)	63,791	63,791	63,791	63,791	63,791
5.	Four programmer analysts (1 for 3 yrs, 3 ongoing)	266,240	266,240	266,240	199,680	199,680
6.	One Web designer (ongoing)	57,600	57,600	57,600	57,600	57,600
7.	.Net consulting services (1 year only: foundation)	100,000				
8.	Transition infrastructure costs (Attunity, 3 years)	10,000	10,000	10,000		
9.	Project management software (one-time only)	20,000				
10.	Training (mostly one time only)	60,000			-	-
		\$830,129	\$650,129	\$650,129	\$436,271	\$436,271

First year (rounded):	\$830,000
Second and third years of project (annual):	\$650,000
Ongoing annual costs after project:	\$435,000

Note: \$225,000 of this is currently being expended annually for the legacy upgrade. These costs do not include software costs for any third-party modules, like finance or HR. The finance and HR modules could add as much as \$400,000 in one-time costs and \$40,000 annual maintenance costs. If HR is not replaced, \$470,000 would be available at the end of the project from the shut down of the VAX VMS and .Net upgrade effort to cover the needed \$435,000 ongoing costs. If

HR is replaced in this project, \$630,000 would be available to cover the ongoing, after-project costs.

ii. Support

The major elements of support for the .Net project are training and project management software. The Attunity software is only necessary as long as the SQL/Server database must mirror the VMS database. The personnel required include a business analyst, a quality assurance administrator, three programmer analysts, and a Web designer. These people will be necessary for a normal level of functionality upgrade requests and integration with new third-party software (for example, online payment check clearing).

Compared with industry standard use of purchased ERP systems, these numbers are low. For example, for the single PeopleSoft module the College now employs, 1 ½ people are required for support. Moreover, this module is not fully utilized and the two people contributing have become proficient at maintaining the software.

Compared with other .Net development projects, the personnel requirement is above average. The College's location away from major urban areas with few software development firms requires that programmers be developed. Few people with a background that would be perfect to work in this project will be found.

iii. Functionality

My examination of existing .Net framework/SQL Server/C# products leads me to believe that the .Net project will result in fully functional administrative software for the College. This framework is ideally suited for the development of online systems. Once foundational work is completed (including security, documentation methods, templates, and basic input/output modules) the construction of function modules can proceed quickly.

The framework may, in fact, be too able to create functionality. Less than optimal business rules will not be challenged by software limitations.

iv. Programming staff satisfaction with direction

The majority of the programming staff believes that the .Net project is the best direction to go. Because of its expense and relative inflexibility, third-party software is rated by ITS staff as considerably lower than expectations for the .Net project. All other development platforms are rated lower as well. Those involved with PeopleSoft would increase the use of PeopleSoft modules. All others add caveats that the project must be fully supported by the administration and the College. They want to see careful project design and competent direction. They wish that the functional requirements of users be managed by a process that leads to a final signoff on design. Those most directly involved with the .Net framework would like more training.

.Net Option Caveats

1. Continuity of project personnel is critical and will be hard to achieve.
2. Business rules will not be automatically challenged, because the software can be custom designed.

3. While a three-year horizon has been accepted as feasible, software projects are never really done.
4. Actions tend to be hard-coded in custom software, requiring technicians to alter them. Designing and producing highly-flexible software that users can change is more challenging and time consuming.
5. A high degree of technical expertise must be maintained to keep the software up-to-date, especially if HR, payroll and financial aid modules are included. If they are not included, then integrating new third-party modules will require expertise.
6. Version control is difficult to maintain with in-house projects. Emergency bug fixes become tempting times to add other pending updates. Users then discover unexpected changes.
7. As-you-go improvements to software design can make training challenging. Users find that they have to unlearn much every time a new version comes out.

.Net Option Strengths

1. Custom-built software can often be made to effect processes in a single step, especially with .Net technology. Vendor software often requires several steps to achieve the same result.
2. The expertise to link the software to any third-party software module remains in-house. With commercial software, experts from two vendors must be brought in (unless a software consultant can be found who has a practice of linking the two systems).
3. The College can more freely design its business systems, knowing that the development people are committed to make the business systems work as optimally designed for the special situation of the College.
4. Screen design can be more easily customized to the needs of the user. Vendors who build software for a range of users tend to put too much on a screen, forcing users to work with multiple screens. Too much that is irrelevant gets displayed.
5. A more gradual rollout makes training easier. Smaller groups can be trained at a time.

b. Other technical options

i. Other development platforms

The .Net framework is the major Microsoft development platform. As such it is being widely used and tested. Help and problem solving among developers is widely shared online. This makes it an easy fit to many new applications. The platform represents the highest development of object-oriented programming and the use of the XML data exchange standard.

ii. Other purchased platforms

PeopleSoft, SCT Banner, SAP, Datatel, Jenzabar, and Campus Management were considered as options. Campus Management rated highest on technology, functionality, corporate stability, and price.

iii. Campus Management evaluation

1. Price

The recommendations include accepting the Campus Management proposal if the software price is \$350,000 or less for the student, finance/HR/payroll, alumni/ development and housing

modules and negotiating the CM implementation price to \$530,000 or below by shifting or adding College resources to the implementation project, including these full-time dedicated positions (including salary and benefit cost estimates):

Table Six: Campus Management Project Costs

			2007	2008	2009	ongoing
a.	Project manager	(2 years)	\$92,498	\$92,498	\$	\$
b.	Two business analysts	(One for 2 years, one ongoing)	89,600	89,600	44,800	44,800
c.	Quality assurance administrator	(ongoing)	63,791	63,791	63,791	63,791
d.	Database administrator	(ongoing)	70,400	70,400	70,400	70,400
e.	Applications administrator	(ongoing)	66,560	66,560	66,560	66,560
f.	Technologies specialist or helpdesk analyst	(ongoing)	55,000	55,000	55,000	55,000
g.	Maintenance and licensing fees	(ongoing)	120,000	120,000	120,000	120,000
			\$557,849	\$557,849	\$420,551	\$420,551

The \$560,000 costs would apply to the two years of implementation, going down to \$425,000 thereafter. In addition, the one-time cost for software and implementation is \$880,000. (With a 6% lease purchase the debt service would be \$210,000 annually over five years, or \$160,000 annually if the lease is for seven years.)

Functionality (Campus Management)

Unlike the systems from the 1980s and 1990s like PeopleSoft and Banner, CM business rules are not contained in tables. Many rules in CM are available from dropdown menus. Others are put into effect with selections of students (all those that owe more than \$100, for example). Operations from the dropdown menus are then put into action against the selection. The keys to a system like this working are 1) the rules available contain a full set of the actions needed, 2) the set of rule triggers is sufficient to cover the needs of the College and 3) the data available for selections contains all the attributes needed.

Actions that change the state of a student (enrolled to withdrawn, for example) or the student's registration in a course (enrolled to completed, for example) are clearly available in CM. Actions like "suggest another similar course if first course is full" are not available. Actions are limited by underlying programming. CM does not appear to be missing any basic actions. All-College course schedule building may be a weakness. They advise using third-party software for scheduling.

Triggers are changes in the database (or time) that cause an event. For example, receiving a passing grade in a final course should trigger a change of state from degree seeking to eligible for degree. Triggers that work in real time require underlying programming. CM software requires more intervention than "fully trigger-able" software. Lists are built (and the building logic can be permanently stored) first, then run to cause the action.

Selections are viewed as a function of the internal Client Resource Management (CRM) of CM. The only limitation of selections is the lack of availability of a necessary data element. Fortunately, CM has a flexible data structure, allowing the addition of missing elements.

CM caveats

1. The software is being used by only one other community college.
2. No users have Blackboard. The tie will be a first time endeavor.
3. No users have PowerFAids. The tie will be challenging.
4. Integration with National Student Clearinghouse, DARS/Miami degree audit, and SPEEDE/EDI transcript transmission will require special set up.
5. Some one-step processes embedded in COBOL programming will require several steps in CM, first to set the selection criteria and second to run the action against the list.
6. No textbook ordering functions are visible. Third-party software will be required.
7. No course scheduling functions are visible. Third-party software will be required.

CM strengths

1. User friendly Web-based interaction screens.
2. Flexible reporting (with Crystal Reports).
3. Easier database access for analysis.
4. Full fund accounting with encumbrances and online requisitions.

A discussion with CM was based on the demonstration specifications listed in Appendix Four. The appendix should be used to guide a series of demonstrations to the software selection committee by Campus Management. The appendix should also be used to set specifications for ERP software in an RFP, in terms of how the software will be evaluated.

Overview Cost Analysis

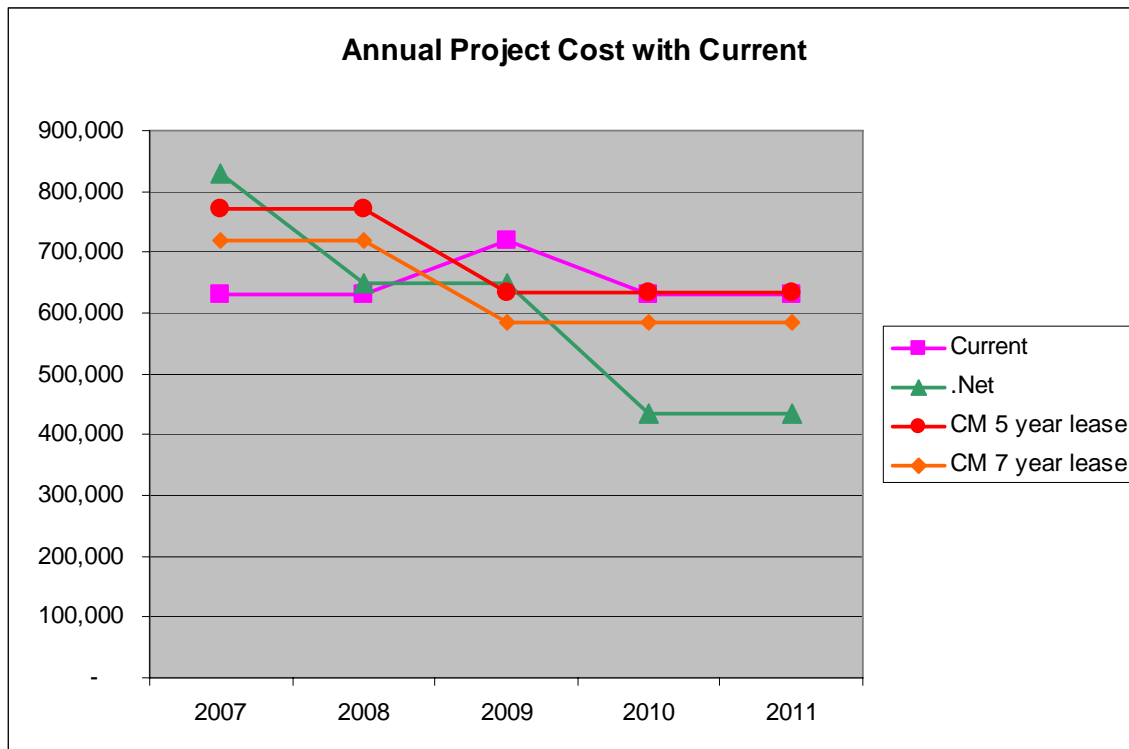
The annual cost of each alternative, assuming software and implementation leases of either five or seven years, is shown in the following table.

Table Seven: Option Direct Costs

Year	2007	2008	2009	2010	2011
Current	630,000	630,000	720,000	630,000	630,000
.Net	830,000	650,000	650,000	435,000	435,000
CM 5-year lease	770,000	770,000	635,000	635,000	635,000
CM 7-year lease	720,000	720,000	585,000	585,000	585,000

A graph of this table follows.

Graph One



The differences between legacy-option costs and legacy costs are as follows. (Negative figures indicate that legacy-option costs are less than legacy costs.)

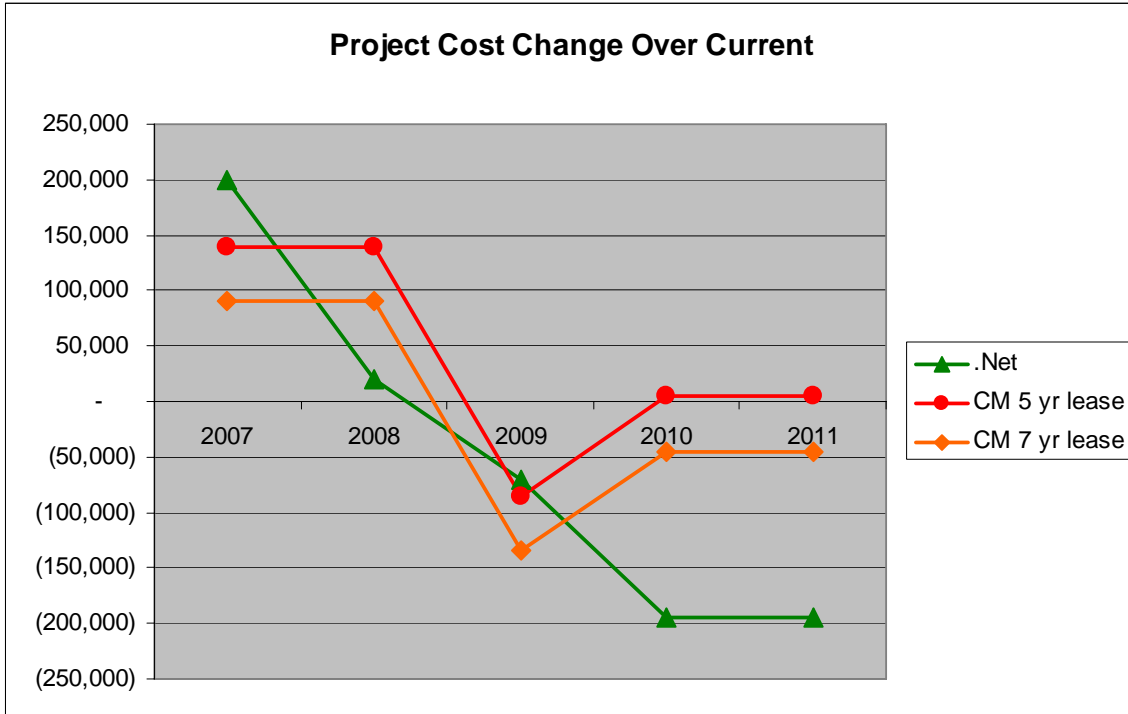
Table Eight: Project Cost Over/(Under) Current Costs

Year	2007	2008	2009	2010	2011	5 yr Sum
.Net	200,000	20,000	(70,000)	(195,000)	(195,000)	(240,000)
CM 5-yr lease	140,000	140,000	(85,000)	5,000	5,000	205,000
CM 7-yr lease	90,000	90,000	(135,000)	(45,000)	(45,000)	(45,000)

(Note: the seven-year lease would extend past 2011.)

Although the costs of the early years of each of the projects is higher than expected for maintaining the current system, later years should provide a savings.) A graph of this table follows:

Graph Two



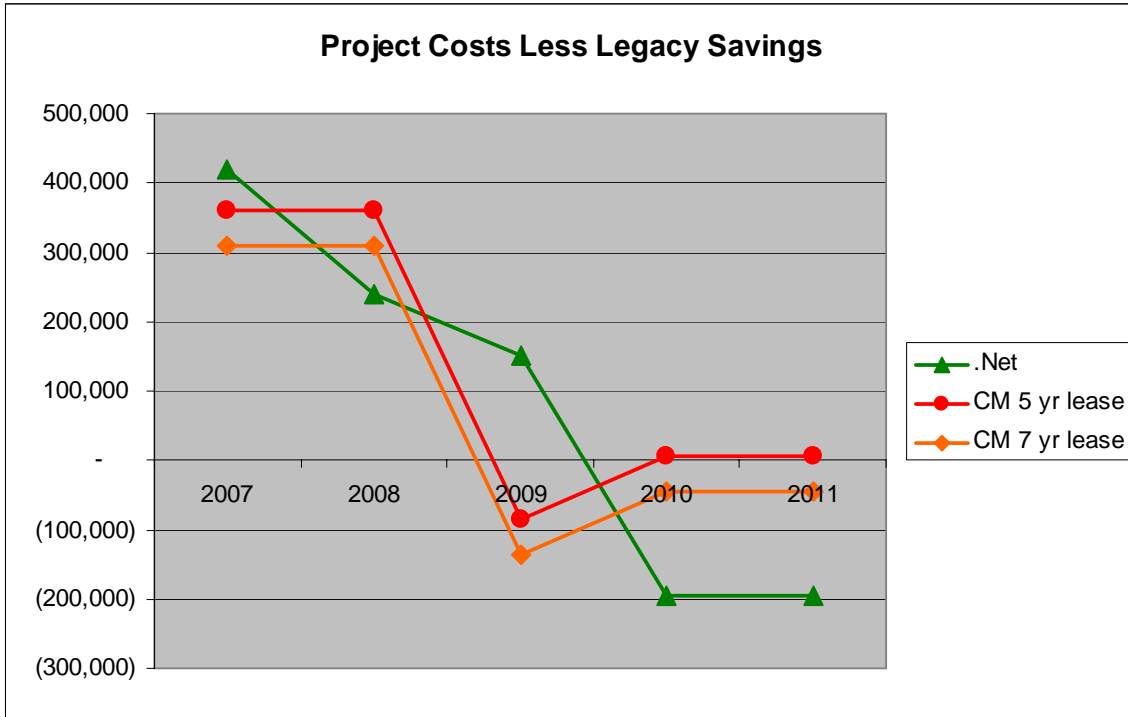
The funds available for the projects would not equal the differences in costs during the transition period. The net cost of each project less funds released from current efforts during the transition is given in the table below. Negative figures indicate a savings over projected legacy costs. For example, in 2010 the direct costs of the .Net project are more than offset, by \$195,000, the savings from ending support for the legacy system and PeopleSoft HR. During transition years, savings do not, of course, include the costs of keeping the legacy system and PeopleSoft HR running. The .Net project transition goes from 2007 through 2009. The Campus Management transition goes from 2007 through 2008.

Table Nine: Project Costs Less Savings From Legacy Decommission

Year	2007	2008	2009	2010	2011	5 yr Sum
.Net	420,000	240,000	150,000	(195,000)	(195,000)	420,000
CM 5-yr lease	360,000	360,000	(85,000)	5,000	5,000	645,000
CM 7-yr lease	310,000	310,000	(135,000)	(45,000)	(45,000)	395,000

Because of the three-year transition for the .Net project, and the two-year transition for Campus Management, the five-year costs for Campus Management with a longer lease are lowest. If the analysis were to be extended to seven years, the .Net project would be lowest. A graph of these results follows.

Graph Three



Summary

The cost differences over five years are too small to allow us to form a decision on this basis alone. Adding in the risks inherent in an in-house development project tips the decision toward the purchase of a full administrative software system within cost constraints. The Campus Management system demonstrates well. It appears to have the functions required by Yavapai College. It should withstand the scrutiny of the Yavapai review and purchasing process. It is priced competitively with an in-house project.

Yavapai should be able to negotiate a very favorable contract with Campus Management. Sale of the system to the College will open a new market area to the company. Nevertheless, the College should not let low pricing open it up to inferior functioning. The College must thoroughly review the software before purchase. On the other hand, users should be warned that in-house systems always have several very special features that are not and cannot be duplicated in a purchase system. To be acceptable, however, the new software should accomplish all critical functions and the special functions should be available with a “work-around” or abandoned as unnecessary.

If the CM software fails the College’s review, the College should set a deadline of six months to bring a full development project together, including hiring. We suspect, however, that programmer turnover will make this impossible. Project design must also be completed in that time.

The best backup to that option is to join the Open Source Consortium. The Open Source project will be slow to unfold, lessening the impact of its probable high implementation cost. This will force the College to live with much of its current system longer than desirable. Nevertheless, reviews of the developing software are good and the Consortium appears to be willing to work with another community college. The first step for this option would be to bring consultants up from the Phoenix-based RSmart group to introduce the College to this option (and, of course, to their services).

The College has taken several steps to address “Web direction” questions since issuing the RFP. We applaud these steps and encourage the College to open up communication channels on Web issues and to gain synergy by consolidating support and training functions.

Yavapai College is a remarkable place. It is quickly moving through an important transition in its development. The maturity being gained during this leadership and strategic change will be necessary to successfully implement “someone-else’s” software.

We found everyone to be open, friendly and extremely helpful. We found the entire IT staff to be absolutely top notch. We found an administration with a strong sense of what will be necessary to make the College successful. We found faculty dedicated to teaching. Even faculty members who evinced a suspicion of technology were taking advantage of it in ways that were new to us.

Yavapai College’s future is bright.

Appendix One Definitions and Clarifications

Definitions

1. Enterprise Resource Planning (ERP). “ERP” seems oddly ungrammatical the way it is commonly used today. ERP is generally employed as shorthand for the full administrative computing system, extending occasionally into instruction-support software.
2. Legacy system. This is the system that the College began developing in the late 1980s. The system is a unified approach to applications that ran in the COBOL/OpenVMS environment on (at that time) Digital Equipment Corporation VAX Alpha hardware. The two basic applications software modules are (from the RFP):
 - “**AMIS** - Admissions, Registration, Student Financial, Financial Aid, Curriculum Development, Course Catalog Development, Textbook Management, Advising, Housing, Room Scheduling, Grade processing, Transcripts processing and Facilities Tracking.”
 - “**Financial Systems** - General Ledger, Accounts Payable, Student Accounts Receivable, Fixed Asset Management, Budget and Bank File Maintenance & Reconciliation.”
3. Hardware. Computers, servers, disk memory, and printers are examples of hardware. The VMS/COBOL system runs on VAX Alpha hardware. C# programs run on server hardware from several manufacturers, including Dell.
4. Operating system. The operating system is the computer software language that controls how applications software runs. VMS is the operating language of the legacy system. In today’s desktop computing environment, operating systems are also called “platforms.” Windows XP and Windows Server 2003 are operating systems/platforms.
5. Platform software development kit. Microsoft Visual Studio is a platform software development kit and is an advanced integrated development environment. It lets programmers create programs, websites, Web applications, and Web services that run on Microsoft Windows, and the Internet.
6. Development and execution environment. The .NET Framework is a development and execution environment that allows different programming languages and libraries to work together seamlessly to create Windows-based applications that are easier to build, manage, deploy, and integrate with other networked systems.
7. .NET framework. The .NET Framework consists of (this is a Googled Web definition):
 - a. “The Common Language Runtime (CLR) A language-neutral development & execution environment that provides services to help ‘manage’ application execution.
 - b. The Framework Class Libraries (FCL) A consistent, object-oriented library of prepackaged functionality.

- c. The .NET Framework provides the basic infrastructure that Windows-based applications need to make Microsoft's .NET vision of connecting information, people, systems, and devices a reality:
 - i. Support for standard networking protocols & specifications. The .NET Framework uses standard Internet protocols and specifications like TCP/IP, SOAP, XML, & HTTP to allow a broad range of information, people, systems, and devices to be connected.
 - ii. Support for different programming languages. The .NET Framework supports a variety of different programming languages so developers can pick the language of their choice.
 - iii. Support for programming libraries developed in different languages. The .NET Framework provides a consistent programming model for using prepackaged units of functionality (libraries) which makes application development faster, easier & cheaper.
 - iv. Support for different platforms. The .NET Framework is available for a variety of Windows platforms, which allows people, systems, and devices to be connected using different computing platforms. E.g. People using desktop platforms like Windows XP or device platforms like Windows CE can connect to server systems using Windows Server 2003.”
8. C#. C# is a new object-oriented programming language from Microsoft, which aims to combine the computing power of C++ with the programming ease of Visual Basic. Based on C++ and containing features similar to those of Java, C# is designed to work with Microsoft's .Net platform. C# simplifies programming through its use of Extensible Markup Language (XML) and Simple Object Access Protocol (SOAP) which allow access to a programming object or method without requiring the programmer to write additional code for each step.
9. XML. XML is a universal programming language for data exchange.
10. Relational database management system. A relational database is a database made of tables, designed such that information in one table (student information) can be linked to information in another (course information), using an element common to both tables (course taken number in the student's file, for example).

Conventions and Clarifications

1. “The .Net effort.” Adding features to the COBOL-based system using the .Net framework (as defined above), a SQL Server database mirroring the COBOL-based database, and employing C# coding will be called “the .Net effort” as a shorthand in this report. “The .Net effort” has not been a truly sanctioned College project to transform the technology of administrative computing. It has been a testing, learning and experimenting effort. Unlike a project, resources have not been fully dedicated to this effort, nor has it been managed like a true project. The .Net effort should be thought of as part of the natural development of the legacy COBOL system. The new mirrored database has increased the capability of the Office of Institutional Research, for example, to keep up with the demands in that area. Returning to a pre-.Net state would remove necessary functions been added to the legacy system. This would not be desirable.

2. “The .Net project.” This is the shorthand name for the option to completely upgrade the administrative computing technology before June 30, 2009. If this path is chosen by the College, the project will be sanctioned, monitored, accountable, and managed by a project manager with all the formality required of such a complex software development effort. Time lines, sequestered resources, milestones, “drop-dead dates” (critical milestones that, if missed, require that the project be abandoned as impossible to complete), training schedules, team designations, documentation standards, common conventions and definitions, object libraries, and project tracking software are basic characteristics of development projects.

Appendix Two Interview Findings

Summary

From the list of “Technology Functions Needed” and “Areas That Need Correction” we have summarized the important ones—the ones that we think would begin to build a basis for judging the adequacy of any system.

The first thing to note is that there are two types of functions that were not well covered by interviewees: those that are obvious and those no one had heard of.

It’s obvious that any finance system should be able to write checks. When buying a system, it is relatively easy to make sure of basic functionality. References are checked. A system missing a basic capability would leave a trail of complaints. When building a system in-house, however, we have to remember that if it is not specified, it may not be built. Using a functioning system as a model reduces the chance of missing a basic capability.

The new “hold technology” is an example of “no one has heard of this.” If users had been familiar with what commercial student systems are providing in the way of hold capability, they might have requested that enhanced capability in the interviews. You can have bursar, registrar, library, security or any number of other types of holds. Each hold sub-type can have its own authorization and removal rules. You can selectively hold registration, re-enrollment, drops and adds, financial aid, refunds, and even informal transcript access. You can have automatic holds and manual holds. Any student with a balance over \$100 gets an auto-hold/bursar (transcripts and add/drop allowed), for example. Holds can have auto-removal conditions. Holds can pop-up on the student’s sign-on page (portal technology). The bursar can remove a library hold for fine payment, but not for non-returned items. A registration hold can trigger an e-mail, warning the baseball coach.

HR systems come with “Training Management Solutions.” Finance systems have workflow capabilities like, delegated approval rights. They can also have the ability to configure business rules on transaction entry.

Not all the requirements that interviewees mentioned had to do with system capability. Some could best be initiated with a review of business processes. Others were ideas on project management and implementation. Many were related more to Web direction than enterprise system specification.

The enterprise system functional requirements were of two types: 1) interaction quality/ease of use, and 2) business process functions. The first group of requirements can be divided into screen-level interaction design, visible design, and technical design. Shopping cart capability is, to me, a screen level interaction requirement. Single sign-on is a visible design element. Trigger-based action capabilities are an element of technical design.

Below is an outline of interviewee requirements in the order described above, stripped of redundancies and less-relevant items.

Full Requirements List from Interviews

1. Functional:
 - a. Interaction quality/ease of use
 - i. Technical capabilities, screen level
 1. Mouse and pointer
 2. Cut and paste
 3. Drop-down
 4. Shopping cart
 5. Click to select
 6. Auto-fill
 - ii. Visible design
 1. “Tight” screens (everything you need for the activity on one screen)
 2. Easy navigation
 3. Standardization with common locations on a screen for common functions
 4. Report writer
 5. Single sign-on
 6. Customizable home pages (portal)
 7. Macintosh equivalency
 - iii. Technical design
 1. Easy to extract data structure
 2. Actions (e-mails, re-packaging, re-billing) triggered by change of state (credit load, scholarship eligibility)
 3. Integrated databases with enterprise
 - a. Skills assessment scores
 - b. Advising
 - c. Admissions (status)
 - b. Business process functions
 - i. A/R
 1. Credit card refunds to students’ cc
 2. Third-party payment system
 3. Transaction audit trail from student account to ledger
 4. Remote cashiering
 - ii. Financial aid
 1. Scholarship database for packaging
 2. Automatic FA eligibility verification
 - iii. Registration
 1. Able to suspend an online registration session (and not have to close out the registration process) for course, faculty or degree information look-up
 2. Automatic wait list status change notification
 3. Waiting for permission state auto-fixed by online authorization
 4. Conditional prerequisites

5. Full explanation of holds and course enrollment rejections
 6. Course scheduling (integrated with event scheduling)
 - iv. Student records
 1. Degree audit
 2. Reverse degree audit (“What courses are needed?”)
 3. Course change history
 4. Online course indicator
 - v. Admissions
 1. Assisted transcript evaluation
 - vi. Finance
 1. Web-based
 - a. Online requisition, travel voucher, and reimbursement request submission
 2. System capabilities
 - a. Accrual
 - b. Encumbrances from purchase orders
 - i. Partial order completion capabilities
 - c. Encumbrances from employee contracts
 - i. Salary disencumbrance upon payment
 - d. Cross-fund entries
 - e. Period reporting
 - f. More than one month open at a time
 - g. Job costing ready
 - h. General ledger
 3. Special features
 - a. State and local tax-owed calculations
 - b. Linked purchase card invoice and department billing
 4. Budget sub-system
 - a. Line-item pooling
 - b. Auto-return of unused personnel budgets
 - c. Projections allowing individual salaries, line-item inflations rates, and adjustments
 - vii. Human resources
 1. Cafeteria plan administration
 2. Applicant tracking
 3. Workflow (new employee)
 4. Payroll integrated with (for pay stub information)
 - a. Cafeteria plan
 - b. Courses (adjunct pay)
2. Business Process:
- a. Person’s address update rules
 - b. Payment deadlines
 - c. Student refunds
 - d. Student residency determination
 - e. Course waitlist rules

- f. Dual enrollment limitations
 - g. Required signatures
 - h. Textbook ordering
 - i. Transcript evaluation
 - j. HIPPA compliance
3. Project/Implementation:
- a. Project management
 - i. Firm report format user sign-off
 - ii. Templates
 - b. Production management
 - i. Version control
4. Web direction
- a. Portal windows
 - i. Instant updates on course changes and cancellations
 - ii. Alumni support and interaction portal
 - b. Online donation capability
 - c. Content management
 - d. New technology mastery
 - i. Blogging
 - ii. E-portfolio access
 - iii. Seamless streaming video
 - iv. Podcasting

Appendix Three Online Survey Analysis Administrative Software Study

Over 300 people responded; twice as many were staff as faculty.

System satisfaction

The first set of questions requested ratings of administrative systems. More than 65% of the respondents had never used any system, except for registration related systems (registration, course scheduling and instructor scheduling), where between 44% and 57% had used the systems.

Among users, financial aid and payroll received the highest relative marks for working well and being easy to use. Travel expense, budgeting, event scheduling, and text ordering, in that order, received the lowest relative marks.

The largest *numbers* of people saying that a system was not easy to use or worse were for class scheduling, student registration, and travel expense. The counts of people who found event scheduling and textbook ordering difficult was not far behind.

Finding: The two purchased, third-party systems (financial aid and payroll) were rated as relatively easier to use (but one comment faulted the number of screens that an HR office person had to work through in the HR/payroll system). In face-to-face interviews with the consultant, the systems most faulted by academic division and departmental administrative assistants (text ordering, class scheduling, and budget management) were also perceived as being difficult by respondents.

Upgrades

The second set of questions asked about the importance of upgrades. More than 50% of respondents said they used registration, class scheduling and data extraction. Fewer than 31% said they used payroll or student billing.

Among those using each system, the highest proportions of users who said that an upgrade was less important were the payroll, financial aid, billing, and hiring users. Among those using each system, the highest proportions of users who said that an upgrade was important were the data extraction, class scheduling, budgeting, admissions, and registration users.

The largest *number* of people saying that a system should be upgraded noted the data extraction, class scheduling, registration, admissions, and instructor scheduling systems.

Finding: Data extraction got a surprising number of votes for an upgrade. More people are doing or wish to do analysis than expected. The next highest vote getters are important facets of student service: class scheduling, registration and admissions. Two were, once again, important to the efficiency of academic administrative assistants: class and instructor scheduling.

Online Services

The third set of questions asked about the ease of use of online services. 70% or more of respondents had not used the online student aid application, online billing, or online payment systems. 60% or more of respondents (but less than 70%) had not used the online registration, grades, admissions application, or course set up.

Among users of each online system, good ratings for use were given to grades, unofficial transcripts, course instruction, online course set up, and online grades. The hardest to use was the online room scheduling system. A nearly equal split between acceptable or better and not acceptable was found for online applications, registration, billing and payment.

Finding: It is heartening that online course instruction received a good rating. The good rating for online course set up appears anomalous. Straight-forward systems, like grades and transcripts, appear to be working well. Online room scheduling appears to be a burden.

Resource Allocation

The fourth set of questions asked whether more or fewer resources should be given to an area. More than 40% of the respondents had no opinion on the purchasing, accounting & budgeting, human resource & payroll, and online student financial aid systems. 94% of respondents had an opinion on resources for the website.

If we add the number requesting that more resources be allocated (the two highest categories), the top vote getters are: the College website, online instructional delivery, student information system, and high technology classrooms. These are all core technology systems.

If we add the number requesting that resources be kept the same or reduced (the lowest three categories) the highest vote getters are: interactive TV instructional delivery, high technology classrooms, online student registration, add and drop, and human resources and payroll system.

High technology classrooms got a lot of votes in both directions. If we subtract the number of less or the same resources votes from the more resources votes, we find that online instructional delivery, the College website, and the student information system are greatly favored. We also find that more people would like to see the same or fewer resources going to human resources/payroll and purchasing. In both of these cases the majority favored keeping resources the same, not reducing them.

Finding: The responses seem to favor “meat and potatoes” over “frills.” More support is desired for online instruction, the website and the student information system. These should be viewed as core technology systems. There was also good support for student recruiting, in line with strategic needs. Other forms of instructional and faculty support were favored, as well, although not as strongly: high tech classrooms and faculty resource centers. All more narrowly-focused areas seemed to have less support for investments.

Summary

1. Respondents would favor a replacement of the student information system.
2. Respondents do not want such a project to take resources from technology areas that support instruction.
3. Respondents believe the website is making some progress but needs considerable improvement.
4. Purchased, third-party systems (financial aid and human resources/payroll) are working adequately.
5. Online scheduling is a “hot button” area.
6. There is a base of support to improve the technology of student recruiting.

Appendix Four

Design for a Demonstration of Functionality for Administrative Software Packages

General

1. How would you set up an ad hoc report showing all unpaid students taking, for example, nursing courses? Show us the features available for customizing reports.
2. Show us an example of your standard reports i.e., trial balance, general ledger, balance sheet, monthly budget to actual reports, etc.
3. Show us how a person could be an alum, student, faculty and staff member simultaneously?
4. How would you set up one address for student bills during the summer, a different one for bills September to May, and another one for paper grades? How would you override the billing address with an e-mail address?
5. How would you reset Campus Management to reflect changes in these business rules?
 - a. Timing and amount of automatic late registration fee
 - b. Timing and amount of late payment fee
 - c. From follow-up unpaid bill letter five days after start of term to portal pushed follow-up unpaid bill banner/notice six days after the start of the class (not the term).
 - d. Increasing and decreasing the delay between student change in status (dropped a course) and generation of refund, or setting a dependent delay based on a set of factors like the time from first day of classes and a student characteristic, like number of credits earned.
 - e. Changing unpaid student registration from course wait list (keeping place in line) to “will be registered upon payment or wait listed at that time” (losing place in line).
 - f. Changing “firm course load limit” to “delayed limit expecting payment to delayed until permission granted or load decreased.”
6. How would a user schedule two 1 ½ hour events that bridge two class periods into classrooms at two different sites over the web?
7. Demonstrate how you manage record locking.
8. Show us how Macintosh users access your system.
9. Demonstrate "On-Line Analytical Processing" by showing how we could find out whether late registrants had poorer course completion rates or not.
10. Demonstrate concurrent record update locking so that we can better understand all the security requirements for confidentiality (cell level?) and, if multiple write access is needed.
11. Demonstrate the software’s authentication and authorization method. How are accounts/roles built? What, if any, interaction is there with Active Directory. If the software’s authentication and authorization is a custom implementation, please demonstrate the API to auto build.
12. Please demonstrate maximum load conditions. The College may have as many as 200 simultaneous staff users and hundreds of students attempting to register simultaneously. What are user limits for users limited to budget look-up and purchasing (requisition, receiving confirmation, and authorization)? Can you demonstrate that load?

13. Show us how you generate identification numbers and how you check for duplicate person entry.

Portal

14. How would your portal handle library security as part of single sign on?
15. Show us how you would automatically push a message to students of a cancelled course during registration or a course location change during the term.
16. Show us how the alumni portal offers the sale of logo merchandise.

Student Records

17. Show us how you would add a new course to an existing term.
18. Show us how you would roll courses forward from a previous term.
19. Show us how you would set up a course whose start dates would vary with each enrollment and whose end dates are uncertain for each enrollment.
20. Show us how you would set up a course whose credit value would be set at the time of grade submission.
21. Show us how you would change a course from requiring another course as a prerequisite to suggesting another course as a prerequisite.
22. Show us how to set up two courses as co-requisites.
23. Show us how you would set up a course that is team taught by three adjunct professors.
24. Show us how you would set up a course requiring a conditional pre-requisite, for example, a music ensemble course that required an audition only once (a prerequisite if audition not flagged in the student's record).
25. Show us how you would change graduation requirements for a major.
26. Show us how a student sees which of their already taken courses qualify for a major as they try different majors and how the courses needed to graduate in that major are displayed.
27. Degree audit: Show us how you would automatically flag a student as having completed requirements for a program. Show us graduation eligibility reporting. How do you show that a person could complete requirements this term, if successful in current courses? How do you display the match between courses taken and experiences against degree requirements for registrar auditing?
28. Curriculum Management: Show us how you track the course history: numeration, pre-/co-requisites, syllabus, outline, and etc. Show us how we would know that a course had had a different number previously and how we would know when the change took place and who authorized it.
29. Show us how we could add a (non-standard) attribute to the course file (like adding a flag to indicate an online course, or a course eligible for transfer to a particular outside program).
30. Demonstrate electronic transmission of data, including integration with National Student Clearinghouse, DARS/Miami (degree audit), and SPEEDE/EDI transcript transmission.
31. Show how your software automates transfer transcript evaluation.

Financial aid

32. Have you built interfaces to PowerFAids?

- a. Which of your customers are using PowerFAids?
33. Show us how to package a student with Pell, subsidized loans, unsubsidized loans, a named scholarship, and an institutional grant.
34. How would you show that 50% of an institutional grant budget had been awarded?
35. Show us how your system changes an aid package when a student drops a class.
 - a. Show us how to set a delay before repackaging, waiting a set time for a course addition.
 - b. Show us how to set an automatic student notification of an automatic change in aid a package by e-mail and letter.
 - c. Show us how to set the system to ask for human intervention in aid re-package after a change of aid status because of a course drop.
36. Show us how your system flags a student as being ineligible for certain types of financial aid because of:
 - a. Grade point (including mid-semester when an incomplete is changed to a low grade)
 - b. Academic progress
 - c. Credit load
 - d. Financial status

Registration

37. Show us how we would go (while counseling a student) from a student's transcript record, to courses required for a degree, to financial aid, to payment history.
38. How would you schedule a course that meets simultaneously (ITV) at two sites using a parameter driven schedule optimizing system?
39. Demonstrate online registration
 - a. Course shopping cart
 - b. Link to course information without losing registration progress
40. Show us how you forecast demand for courses. That is, do a reverse audit to see what students need, where to offer what they need and what type of delivery (for those who are online students, for example).
41. Show us how you would switch who can give online permission for a student to register for an ineligible course (missing a prerequisite or low test scores) from a department chair to a faculty member.
42. Show us how an instructor or advisor could be given override privileges on class size limits to enroll a limited number of students.
43. Show us how you would synch with external data (test scores) to block registration for courses at too high a level for the student's tested level.
44. Show us how to block certain courses by student attribute (admission's status, for example).
45. Show us how you would set up an e-mail message to go to a student automatically when he or she is lag-registered into a course for which he or she was waitlisted.
46. Show how to set warnings or blocks to students attempting to waitlist a class in a time conflict with existing class and to notify student when a wait listed class is opened that another remains on his/her wait list.

47. Show us how to allow students to set an automatic drop for a course when he or she becomes registered into a course for which he or she was waitlisted (to avoid an overload block).
48. Show us how to set a student assistance feature that suggests similar courses to the one that is waitlisted.
49. Show us how to link a class roster and course description to Blackboard. Which clients link your system to Blackboard?
50. Show us how textbooks are ordered with departmental sign-off and how required and suggested textbooks can be viewed by registering students.
51. Show us how class size information is transmitted to the bookstore and demonstrate how notification is sent when a class size limit is increased.
52. Show us how to edit text responses tied to reasons for blocking a student's online registration into a course ("course filled" or "missing prerequisite ENG 103," for example).
53. Show us how to set up a course taught by an adjunct that includes the set up of payroll information: earliest pay date, latest pay date, and amount to pay for course.
54. Show us how to recalculate the salary and send that information to payroll of an adjunct professor who stops teaching part of the way through a term.
55. Show us how to set up an exception to allow a student to register in the same course twice at the same time.
56. Show us how to register a cohort (block registration) by either having a single course sign-up cause multiple course sign-ups or by connecting registration in any cohort course to trigger registration in all.
57. Show us how advisors can log notes into the system, set flags for follow-up meetings, and set event-driven email to student advisees.

Student billing

58. How would you handle a cycle of accepting an online credit card payment, changing registration (student initiated), and refunding the balance to the payee's credit card account?
59. How do you handle third-party payments?
 - a. Entering a student?
 - b. Choosing among payment responsibilities for the third-party?
 - i. Dependent on completion or grade
 - ii. Full, fixed \$, percentage, and excluding certain fees and/or room and board?
 - c. Changing the format of the third-party bill?
 - i. Notification of student action
 - ii. Amount due
 - d. Having different billing dates for different third-party payers?
 - i. End of term
 - ii. Half-way
 - iii. Set date
 - e. Reporting amount owed by agency
60. Show us how to track a student accounts receivable general ledger entry back to the transactions in the student system that led to the entry.

61. Show us how you switch your system from posting student accounts receivable transactions live to batch posting after inspection.
62. Show us how to close a cashiering session at a remote campus.
63. Show us how to set up a payment allocation rule that goes from “first owed, first paid” to “pay only against current term tuition.”
64. Show us how to generate a list of students eligible for refunds and how to select students for refunds.
65. Show us how a student can make an online change to cause a credit balance to be refunded by check to “carry credit balance to next term.”

Finance

66. Show us how we could add attributes to transactions (for example, classifying outside of the accounting system a transaction as being part of “Strategic Initiative 3”).
67. Show us how to enter a new employee and encumber a salary budget with the appropriate budget (not payroll) amount.
68. Show us how to enter a purchase requisition, approve a purchase requisition and generate an electronic (sent electronically) purchase order. Show us an example of how an encumbrance is created for a PO. Show us an example of how an invoice is paid (check is cut) and related encumbrance is liquidated.
69. Show us an example of how a vendor file is added, deleted or changed.
70. Show us how we can have multiple ship to addresses (for multiple campuses) and one bill to address.
71. Show us how purchase order numbers will be assigned.
72. Show us how we can put notes or special instructions on our purchase orders.
73. Show us how we can complete partial receipts on purchase orders and share that information with Accounts Payable.
74. Show us how we can close purchase orders after all items are received and share that information with A/P.
75. Show us how we can write a report showing all orders that are pending/overdue/by department/by vendor/by commodity/with fixed assets.
76. Show us how we can electronically route and track our budget approval process, and a way to track where the requisition is in the process.
77. Show us how your product would interact with e-commerce solutions.
78. Show us how a vendor could register with the College on-line and be put in an awaiting approval file until a member of Purchasing has reviewed and approved the application.
79. Show us how the program would recognize whether an order includes a fixed asset.
80. Show us how to change from one approval for all purchases on a line to one approval except for purchases over \$10,000 which require a second.
81. Show us how to automatically add state and local taxes to a payment for certain types of goods or services.
82. Show us how to file a travel voucher on line.
83. Cross-fund entries: show us how to purchase a piece of capital equipment (capital fund) with funds from an unrestricted fund cash account.
84. Show us an example of an automatic self balancing entry which crosses two separate funds.

85. Show us how to keep two periods open (show us both two open months and two open years) and make entries into both periods.
86. Show us how items in the same budget pool can be tracked separately but draw from a single budget pool (office supplies, instructional supplies, and maintenance contracts expense accounts drawing from a single budget pool).
87. Show us how to project annual budgets ahead one year where we want to add to the base (positions and non-personnel budgets) and add an inflation factor that differs by line (benefits going up faster than faculty salaries and salaries going up faster than office supplies, for example).
88. Show us how the budget module populates the new budget amounts in the appropriate general ledger accounts after the adoption of the annual budget.
89. Show us an example of the year end closing process and the process of rolling balances forward into the next year.
90. Show us how to forecast an end-of-fiscal year budget position, using spending and revenue rates to-date.
91. Show us how to generate an ad hoc financial report like one that lists all academic advertising expenditure transactions larger than \$200 or that lists all expenditure transactions with a non-standard attribute (like a particular facilities renovation job number).
92. Show us how you have automated the distribution of charges from an electronic purchase card invoice to the relevant departments.
93. Show us how to find all software vendors from whom we have made purchases in the last two years (assuming two years of history, not a start year).
94. Show us how we could set up the accounting records of the Yavapai Foundation as a separate entity within the same accounting system.
95. Show us an example of how transactions are processed through a cash register.

Admissions and Recruitment

96. Show us how you record contacts with a potential student.
97. Show us how to design a workflow to alert staff on when follow-up should occur for contact, missing admissions information, and missing financial aid information.
98. Show us how you roll over an accepted student's admissions records into a formal student record.
99. How do you correct a newly entered address automatically that does not match an existing USPS-standard address?

HR/Payroll

100. Show us how applicants are tracked for particular jobs, for example, how to enter that an applicant missed the interview.
101. Show us how a new employee workflow works with notices to ITS to obtain a computer for the employee, notices add the new person to payroll, and automatic directory entry.
102. Show us how to set up a cafeteria benefit plan that allows a wide range of pension options, pension matching options, health plans etc., and shows all deductions on the check stub.

103. Show us how to generate a payroll entry based on adjunct file information for an adjunct teaching two classes with different start dates within the same semester.
104. Show us how to process and confirm a normal payroll.
105. Show us the set-ups for different employee pay groups.
106. Show us how your system handles multiple jobs for one employee.
107. Show us how your system handles account distributions for Federal Work Study Employees.
108. Show us how to reverse a paycheck or direct deposit.
109. Show us how to do an off cycle check.
110. Show us how to enter hours worked for part-time, hourly employees and how to enter overtime hours for full-time nonexempt employees.
111. Show us how your time and attendance system works with exception reporting.
112. Show us how your system does faculty accrual at the end of the fiscal year.
113. Show us how to query the system for custom reports.
114. Show us how your system handles limit tracking for multiple 403(b) companies for one employee.
115. Show us how to set up different check distributions for each employee.
116. Show us how to enter multiple direct deposit accounts for one employee.
117. Show us how to pay stipends or additional pay over a set period of time.
118. Show us how to set up and deduct for different garnishments (child support, tax levies, creditor garnishments).
119. Show us how vendor checks are processed for deductions.
120. Show us how to end a quarter and the reports that are generated to process a 941.
121. Show us how your system processes imputed income for GTL.