



CAMPUSSM
MANAGEMENT

Mapping Information Technology Strategies to Answer Today's Agenda

The .NET Blueprint for Growth-Oriented Higher Education Institutions

Mapping the mission to Information Technology

Overview

The realities of higher education in the 21st century continuously drive college and university leaders to pursue new avenues to growth. The expanding menu of initiatives can include:

- Online course delivery
- Satellite campuses
- Executive education and corporate seminars
- Portals to serve prospective students, enrolled students, administrators, faculty, employees, and corporate recruiters
- Innovative content delivery via smart phones or PDAs
- Partnerships with organizations to attract, retain, educate, and cultivate diverse student populations

Ultimately, IT is challenged to make these initiatives operational, and to provide comprehensive information snapshots for better decision-making—as close to real time as possible. Also, technology professionals must integrate otherwise disparate legacy information, creating dynamic data-sharing and workflows among multiple campuses, departments, employees, faculty, students, and corporate recruiters.

This all must occur against the backdrop of regulatory and institutional requirements. At the same time, budgets are not geared to grow at the same rate as enrollment and institutional advancement income. IT must achieve more, faster, and with only incrementally greater resources:

- Students and faculty expect a single view and coordinated interactions using Windows or Web-based portals.
- Administrators need to track students and their relationships to the school from recruitment and admissions to graduation and beyond.
- Recruiting, admissions, and academic records directors must streamline processes to meet student expectations and hold steady or decrease departmental overhead.

- Presidents, Finance Directors, and other executives require better information more frequently, to make decision and understand trends.

Unconventional wisdom

Forward-looking post-secondary educational institutions have long aimed to perfect their critical business functions. Recruiting, enrollment, financial aid packaging, and student retention have long been the lynchpins of for-profit educational enterprises. By examining their best practices in terms of their IT decisions, specific guidelines emerge as it relates to technology and infrastructure.

Select a configurable rather than a custom software application suite. Configurable applications are robust systems requiring less custom programming, integration, and maintenance. Typically, configurable applications implement more swiftly. A recent Core Data Services study has shown not-for-profit IT leaders increasingly prefer configurable solutions [2003]. Employing systems based on the configuration model helps to mitigate the constant resource strain most IT managers encounter.

Ensure that a contact and workflow engine is embedded across all applications that manage the student lifecycle. Constituent Relationship Management (CRM) is a leading tool in recruiting, retaining, and cultivating students and alumni. Engines that are native to all aspects of a school's ERP system provide consistent data and decision-support tools that are both immediate and extremely thorough.

Prioritize the executive dashboard when specifying an ERP system. Know what kinds of business intelligence and decision tools a system offers. Executives should come to expect up-to-the-minute visibility and auto-reporting of critical data—regardless of school, department, or campus locale.

Favor solutions supportable by the most available data base and programming talent. Microsoft .NET architecture uses simplified business objects rather than lines of code. It is a framework, not a proprietary language; .NET programmers and administrators are more plentiful and productive.

Emphasize security and scalability for a long-term fit. Microsoft SQL enhances scalability through 64-bit computing, relies on open standards and interfaces, and provides the management framework to ensure constant availability.

Leave options open for future vertical application integrations. By requiring a language-independent platform, such as .NET, future best-of-breed applications and development philosophies remain more viable.

Why .NET software is better software

Software architecture for ERP has evolved since the late 1990s and focuses on efficiently implementing business rules. That approach allows the administrator to focus on how rules are packaged and invoked within the system and how easily they are exposed to partner systems. There is less need for emphasis on the 'how' and more flexibility to implement the 'what'. That is why .NET and the standards-based approach warrant a re-consideration of ERP systems from the ground up. A checklist of the characteristics of a modern ERP system must include:

- **100% managed code.** Truly modern software packages should be completely constructed of managed code. This frees the development organization from spending cycles constructing baseline functionality surrounding security, naming conventions, component communication, and language integration. Not only does this methodology allow for more energy toward solving business problems, it provides for a consistent platform on which to build future functionality.
- **Single set of business rules.** Systems should package business rules in such a way that duplication is avoided. Ideally one copy of each rule should exist and all transactions from all sources use that one instance. The clear advantage of a single set of rules is much lower cost of development and maintenance and greater speed and agility in forging connections to other systems.
- **Bundle validation rules with processing rules.** A key to building a powerful single set of business rules is the ability to keep the logic that tests incoming data for validity readily accessible to the rules that perform the processing and database updates for each transaction. Allow the very same validation logic that serves the user interface to be invoked to protect the system in non-user-active situations such as EDI and BizTalk scenarios.
- **Dynamic choreography of business rules.** A desirable trait in architecture is that it allows business rules to act like a network of specialists, each handing off at the limits of its expertise to another specialist for further processing. Each specialist knows nothing of the actual work performed by its associates. The architecture carefully orchestrates the transaction threading to bring each specialist into play. A powerful architecture lets all of these silos operate according to their rules, but if problems arise it will handle a graceful rollback to previous states.
- **Support rich and lite-client interfaces.** Neither rich client nor lite-client interfaces are superior - each has its place. The architecture should provide strong support for both native code and browser-based user interfaces without any development time penalty, thus allowing the software developer the freedom to select the appropriate presentation layer for the job.
- **Distributed processing.** The architecture should allow processing to be completely accomplished on a single CPU or distributed across many systems to spread the workload. In other words, provide both "scale-up" and "scale-out" capability.

- **Readily configurable.** A modern architecture should provide a framework in which business rules and user interfaces can be easily tailored to the individual organization and where those configurations are kept isolated from the standard business rules and UI code. This allows for pain-free software upgrades and guaranteed retention of institution-specific configuration. When a single, universal set of rules exists, changes can be implemented by revising the one single instance of the school's custom rules.
- **Native support for standards-based connectivity.** XML is now the indisputable standard for data exchange. A modern architecture should natively speak XML when conversing with the outside world by publishing its inbound data formats with XSD and receiving inbound data as XML. Internally, data flowing through the system that is visible as XML opens the door to innovation. However, to ensure compatibility with legacy systems there must also exist the capability to exchange data in traditional flat formats.
- **Loosely coupled.** Interactions with outside systems should be at the level of granularity of the business document: invoices, sales orders, forecasts, shipping notices, etc. Some ERP systems today expose only a fine-grained interface to the outside world that requires intimate knowledge of the ERP system on the part of the outside party. Such knowledge, while allowing a close connection, causes the interface as a whole to be fragile and more costly to create and maintain.
- **Expose a rich standards-based set of data to the outside world.** While separate business entities should exchange data at the more formal level of a business document, the architecture should also allow outside parties to answer their own questions by obtaining detailed data through an open, standards-based mechanism. In short, under security control, any ERP data or process should be readily exposable through Web Services. The architecture should expose familiar business objects for major entities and transactions in the system: Customer, Customer Statement, Vendor,

Vendor Performance, Item, Inventory, Work Order, Purchase Order, Project, etc., allowing trading partners to readily find and retrieve the information they need.

- **Single focal point for inbound data.** For inbound documents, the architecture should expose a single "catchers mitt" supporting EDI-style batch transactions and "batch of one" transactions. Such a single entry point should accept all inbound XML documents that are in the proper (published) format arriving from a valid and permitted source. Integrators simply pass a valid XML document or equivalent text file and let the business logic be invoked to either successfully process it into the system or suspend it, awaiting error review and correction. Documents can come from external Web Services, BizTalk or any other Enterprise Application Integration (EAI) mechanism.
- **Provide event-based communication.** Besides supporting deliberate bi-directional data exchange with trading partners, the architecture should fire events which can be listened for by trading partners. When situations arise or certain conditions are met, collaborating systems can respond to each other's needs. In other words, the architecture should let trading partners ask for or pull what they need, and also push transactions and event-based messages when situations warrant.

The new reality

Some analysts are quick to point out that .NET concepts have existed for a decade or longer. The difference today is that the technologies used are no longer overly complex and proprietary. Companies have used integration technologies, such as CORBA, DCE, and DCOM, to build IT architectures, but they were only feasible for businesses with tight control over the implementations—not typically a characteristic of higher education. Further, DCE was too complex for most organizations and CORBA was too loosely defined with vendors offering different, incompatible tools.

The .NET of today embraced by Campus Management focuses on simple connections to Web Services. Business analysts and clients work together with developers to design services and interfaces that other applications can use. Developers are still integral to the process; they have to code behind the scenes to set up the implementations, but programming languages now make that easier.

On the other hand, as long as implementation delivers operational services effectively, Campus Management's clients are less concerned about the details of implementation. This distinction is paramount to the success of .NET.

Although almost all vendors accept the separation of web interfaces from programming mechanics, other issues are not so clear-cut. For example, some vendors choose to expose Web Services through proprietary messaging middleware and associated adapters (sometimes called connectors). This can achieve higher performance or quality of service than available with SOAP and HTTP. Not surprisingly, fully standardized and proprietary solutions represent tradeoffs among efficiency, quality of service, cost, and complexity.

Campus Management's .NET implementation supports loosely coupled services and simplicity at the integration layer. Campus Management then augments security, reliability, and quality of service as necessary.

Agility for educators means success

The world of higher education is characterized by rapid change, unexpected shifts, and relentless competition. In this environment, colleges and universities need the ability to adapt quickly to change, and to use change as an advantage. Agility means success.

Campus Management's .NET solutions offer educators a flexible, responsive IT architecture that empowers their system to reach their potential faster.

There is growing consensus in the industry that the way to create this kind of adaptive, agile IT architecture is

through Web Services—discrete units of software that interoperate, based on industry-standard protocols, across platforms and programming languages.

How does Campus Management use .NET?

Campus Management builds web components which compose the administrative processes and business systems. A key best practice has been to model our service interfaces to align closely with an administrative capability model. For example, most educators have a business requirement to accept tuition payments. Defining an information set that describes a tuition payment and an end point that accepts instances of this information set is an example of a well-aligned service.

Administrative processes are much more volatile than the information they manipulate. Processes are subject to the judgment and whims of human, as well as to myriad administrative exceptions. Every administrative process instance may chart a unique path through our service portfolio.

Campus Management believes systems must be agile. Process orchestration should be easy to modify or even be completely dynamic. Hard-coding business process sequencing into compiled executables is a negative approach for successful administrative systems.

The concept of event-driven architecture can be instructive as to what constitutes effective process design. Processes should be aware of the administrative events that move them forward or that derail them from the normal path. When an exceptional event derails a process, the focus should be on normalizing the process and getting it back on the main track; manually seeing the process through to completion will balloon the process cost far beyond its value to the organization.

CampusVue administrative software

The release of CampusVue™ marks the first release of an ERP product with any significant portion of functionality in pure Microsoft .NET technology; our CampusPortal™ runs solely in .NET.

Campus Management is committed to making CampusVue the first ERP software product in the Higher Education market to be pure .NET. To reach this goal, Campus Management started from the ground up and spent 18 months designing a new architecture that takes full advantage of .NET enabled programming, XML, Web Services, and the new features of Microsoft SQL Server.

In 2003, Microsoft recognized Campus Management's efforts by naming it the Global ERP Solution Developer of the Year.

CampusVue architecture

Exclusive use of Microsoft technology establishes a seamless interaction among all elements of the CampusVue architecture. The CampusVue platform is implemented using Microsoft's industry leading technologies. We write CampusVue using Microsoft Visual Basic, various .NET languages, and Microsoft Transact SQL (T-SQL). CampusVue uses Microsoft SQL Server® as the database platform.

Windows Server2003 is the foundation upon which all of our solutions operate. This provides for an easy deployment, a secure infrastructure, rapid development and collaboration, and a stable, standards-based internet environment.

Analytics and business intelligence

Campus Management's business intelligence platform strategy is rooted in Microsoft's database offering: SQL Server. SQL Server is the anchoring storage and query technology behind .NET environments. SQL Server features provide OLAP tools, data mining, and "build and manage" capabilities for relational and multi-dimensional data warehouses.

Microsoft's platform strategy enables organizations of any size and of any IT budget to deploy business intelligence throughout and to deliver and achieve improved effectiveness, greater efficiency, and higher quality throughout all their administrative processes.

Administrators must access and analyze information to make decisions about budgets, course offerings, recruitment, marketing strategies, hiring needs and investments in IT projects. Without the efficient use of information for supporting the institution and making administrative decisions, the academic goals simply can not be met.

Configurability versus customization

CampusVue software's Configuration Manager provides access to all control settings and keeps administration chores to an absolute minimum. These control settings give the administrator simple, yet powerful control over all areas of application presentation and behavior. This tool allows you to personalize the system to match the way your school does business without making customized code modifications. Settings are stored at the database level and are maintained throughout all upgrade cycles.

In addition to ultimate console control, the CampusVue flexibility extends to the actions of deployment, interface, and integration.

- CampusVue uses an automated deployment process that installs all server and client systems from a single package. It also automatically deploys any custom modifications.
- CampusVue provides a rich, highly visual client user interface experience. The familiar Windows interface reduces the learning curve and makes CampusVue extremely easy to use.
- CampusVue also takes advantage of the power and performance of common Microsoft desktop applications, seamlessly integrating with Microsoft Office, Internet Explorer, Exchange, and Outlook for enterprise communication and access to the Internet.

- The CampusVue Contact Manager tool allows you to automate the process of sending e-mails and on-screen messages to employees, customers, and partners in response to specific events.
- Import Wizards quickly and efficiently load your legacy data into the CampusVue database.

The bottom line

Campus Management's direction—reinforced by the experiences of our customers and partners—is to continue to develop our solutions using Microsoft's .NET architecture. Our goal is to eradicate the need for complexity in both the foundation and the presentation of user and programming models.

Our approach means we provide our customers with a robust and completely integrated infrastructure, ease of administration and modification, seamless connectivity to other products, and reduced IT costs. Consistent technology also means that as a software provider, Campus Management is able to put its development energy into building features and functions that directly benefit our customers, rather than spending time chasing solutions to tough problems rooted in technology clashes.

Driven by the need to achieve greater insight into education activities, IT departments are seeking effective and expedient means to integrate their application portfolios. The goals are transparency and coherency. To maintain an industry-leading approach we constantly ask ourselves:

- Do we have a consistent view of our students, faculty, and administrative relationships that allow us to serve student and administrative needs?
- Are administrative processes efficiently managed to comply with organizational mandates and government regulations?

- Are our systems effectively delivering value against our education objectives?
- Are our technology investments making our customers as productive as they can be?

To achieve transparency and coherency, educational organizations must create connections. To create consistent management of information, they must connect systems. To create consistent administrative processes, they must connect human and technical capabilities. To create collaborative teams, they must connect workers. To create effective value chains, they must connect organizations.

Campus Management believes its IT orientation is the core of an effective, constituent relationship management strategy. CampusVue is the direct result of belief in, and support for Microsoft's .NET Architecture and simplicity.

About Campus Management

Every day, more than 1,000 campuses and hundreds of thousands of students, faculty, administrators, and staff worldwide rely on Campus Management for complete and cost-effective administrative solutions. Rapid implementation, robust applications, and responsive support have defined Campus Management's leadership in higher education since 1988. For scalability, ease of use, and the lowest possible cost of ownership, CampusVue™ enterprise administrative software has become the cornerstone of excellence for a diversity of best-in-class institutions.

©2006 Campus Management Corporation.
 Campus Management Corp, CampusCare, CampusLink, CampusNet, CampusPortal, CampusVue and CampusVue Collegiate are trademarks of Campus Management Corporation, registered in the U.S., other countries or both.
 Microsoft and Microsoft.NET are registered trademarks of Microsoft Corporation.
 Macintosh is a registered trademark of Apple Computer, Inc.

Campus Management Corp.
 777 Yamato Road
 Boca Raton, FL 33431

Tel: 561-923-2500
 info@campusmanagement.com
 www.campusmanagement.com

CAMPUSSM
 MANAGEMENT

The information contained in this document represents the current view of Campus Management Corporation on the issues discussed as of the date of publication. Because Campus Management must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Campus Management, and Campus Management cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. **CAMPUS MANAGEMENT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.**

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Campus Management. Campus Management may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Campus Management, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.
