EDUCAUSE CENTER FOR ANALYSIS AND RESEARCH

Enterprise Application Projects in Higher Education

EDUCAUSE

Contents

Foreword 2
Executive Summary 3
Introduction 7
Findings 9
- Why Were These Projects Undertaken? 9
- How Well Were Institutions Positioned for Their Projects? 12
- What Approaches Were Employed? 13
- What Challenges Were Encountered? 16
- How Successful Were These Projects? 20
- What Objectives Were Achieved? 22
- How Much Did These Projects Cost, in Dollars and Time? 24
- What Are the Ongoing Costs of System Operation? 29
- What Internal Staff FTE Are Needed for Implementation and Ongoing Support? 32

Advice and Recommendations 33
Methodology 36
Participating Institutions 37

Author

David Trevvett, Consultant and Former Senior Director for Administrative Systems, University of Chicago

Citation

David Trevvett. *Enterprise Application Projects in Higher Education* (Research Report). Louisville, CO: EDUCAUSE Center for Analysis and Research, August 2013, available from http://www.educause.edu/ecar.

EDUCAUSE

EDUCAUSE is a nonprofit association and the foremost community of IT leaders and professionals committed to advancing higher education. EDUCAUSE programs and services are focused on analysis, advocacy, community building, professional development, and knowledge creation because IT plays a transformative role in higher education. EDUCAUSE supports those who lead, manage, and use information technology through a comprehensive range of resources and activities. For more information, visit http://www.educause.edu.

Foreword

What will it cost us? How long will it take? What can we do to help ensure success?

Any institution needing to implement or upgrade a major, enterprise-wide application such as a core student or financial system quickly confronts such questions. But up until now extremely little data have been available to help provide answers. This ECAR study represents an initiative by EDUCAUSE to begin addressing some of those gaps in our knowledge. What emerges is a fascinating blend of incremental change in some areas yet rapid shifts in others, with some common lessons across all, and perhaps the first systematic collection of cost data across system areas and institutional classifications.

The survey underlying this study collected data from institutions that had completed a primary enterprise application implementation within the previous five years in one of five core application areas: financial, grants management (pre-award), human resources, learning management, and student information.[1] The survey sought answers to the following questions and looked for relationships among them:

- Why was the project undertaken?
- How well was the institution positioned for the project?
- What approaches were taken?
- What challenges were encountered?
- How successful was the project?
- What objectives were achieved?
- How much did the project cost? How long did it take?
- What are the ongoing costs of system operation?
- What internal staffing levels were needed for implementation and ongoing support?
- What key lessons were learned? What advice would respondents offer to others?

The 128 survey responses came from a wide range of institutions, with 107 U.S. and 21 international institutions, including representation from the major Carnegie Classification groupings. See the Methodology section of this report for a summary of institutional characteristics of participants. The largest numbers of responses were for student information and learning management systems, which provided an intriguing study in contrasts between these two characteristically different areas. Open source, outsourcing, external staffing costs, and key challenges were all areas with interesting findings.

There is never an end to enterprise application projects—they always lie in each institution's future. The findings of this report can help EDUCAUSE members better position themselves for those inevitable projects to come.

Executive Summary

> Respondent comment: "The least expensive and most successful way to do a large project is to invest properly in the planning, implementation, and ongoing costs. The cost of shortchanging a large project (as differentiated from a small project) is huge in terms of dollars and perception."

Enterprise application implementation projects represent substantial investments by higher education institutions, with some projects costing millions or even tens of millions of dollars. Their ongoing operating expenses play a large role in institutions' annual budget planning. Specifically, the 123 institutions that reported implementation cost data in this survey had spent in the aggregate over $320 million implementing the reported systems and currently spend more than $39 million every year operating them. That is *without* counting much of the internal staff support for these projects and systems.

Institutions often want to know the average or expected cost of implementing an enterprise application, or more specifically, a particular vendor's software. Focusing on averages or medians, however, can be misleading—just as extraordinary diversity exists in the size, culture, complexity, and orientation of higher education institutions, so too is wide variation found in the nature, duration, and cost of enterprise application projects. Two institutions implementing exactly the same vendor's software can experience substantially different costs, as an example in this survey illustrates. Nevertheless, some common themes and findings emerged.

Key Findings

- The success rate is high.

  Resoundingly, most projects reported on in this survey were deemed successful, even very successful. Most institutions reported having been well positioned for their projects, with strong executive and institutional support and with effective project processes in place. Given the difficulties experienced by many institutions in the late 1990s and early 2000s with such enterprise application projects, this is good news.[2]

  Across all system areas, upgrades were regarded as more successful than new system implementations.

  Overall, these projects were overwhelmingly seen as producing many positive benefits for their respective institutions, from improving service delivery and reducing business risk to enabling primary users to better perform their roles.

- It's really about the business, not the technology.

  Enterprise application projects of the type reported here are primarily centered on and challenged by business processes and business objectives,

more than by technology. That is not empty rhetoric—the survey respondents made it clear that project success can depend heavily on business readiness and on the ability of both organizations and people to change. It was in these areas that some of the greatest difficulties arose—for example, user adaptability to change and user training were the most-cited project challenges.

- Open-source applications are in spectacular ascendency in some areas but almost completely missing from others.

  Open-source software has made impressive gains in the learning management arena (61% of LMS responses to this survey, mostly Moodle but with some Sakai) and even in financial systems (Kuali). Yet the student systems reported were almost exclusively vendor software.

  Open-source applications were less expensive to implement than vendor systems, a result that could not be explained simply by the absence of software license and support fees.

- Outsourced strategies show a significant presence overall, turning up in 40% of responses, but as with open source, the extent varies greatly by system area.

  The effect was strongest for learning management systems, with 63% of respondents reporting some form of outsourcing strategy, compared with 18% for all other areas combined. This LMS percentage is substantially larger than that reported in the larger annual Core Data Service, where only 23% of LMS implementations in the past five years were outsourced, perhaps indicating the use of external hosting during the project but not in subsequent operation. Both surveys agree on the trend toward outsourcing LMSs.

- Both implementation and operating costs vary substantially by system area and institutional classification, with LMS implementations uniformly the shortest and least expensive, and with research-intensive institutions spending the most across all system areas, even compared with other doctoral institutions.
- In almost two-thirds of reported projects, the application implementation required other separate, generally independent capabilities or technologies as well (e.g., identity management, document management, other infrastructure).

The tables and charts are informative, but also valuable were the numerous comments provided by survey respondents—some are quoted throughout the report. Recurring themes appeared among these comments, highlighting what it can take to be successful. Factors cited include the following:

- Patience, persistence, perseverance
- Effective planning and preparation
- Strong, capable staffing for the project
- Time and financial resources
- Strong project governance

- Attention to critical deliverables
- Flexibility when absolutely necessary
- Getting the project on people's priority lists
- Excellent partners (vendors, contractors, peer community)
- 100% dedication, plain old hard work, long hours, and sometimes brute force

It is likely that the cost data in this report will be of particular interest to readers. Table 1 provides an overview of implementation costs reported in this survey. Table 2 lists the particular software vendors and other sources that were reported for different system areas and solutions.[3]

Table 1. Implementation Cost Ranges (dollars in thousands)

Category	Number of Institutions	Minimum	Average	Maximum
All projects	119	$0	$2,205	$50,038
Upgrades				
All system areas	18	$23	$444	$1,700
New Implementations, by System Area				
All system areas	101	$0	$2,519	$50,038
Learning management	56	$0	$278	$5,000
Student information	31	$105	$5,333	$50,038
New Implementations, by Solution Type				
Open source	41	$0	$1,121	$14,400
Vendor	57	$0	$3,637	$50,038

Table 2. Vendors/Solution Sources Represented in Survey Data

Type of System	Vendor/Solution Source
Financial Information Systems	Kuali Financial System PeopleSoft SAP
Human Resources Systems	Ceridian Meta4 Oracle (non-PeopleSoft) PeopleAdmin PeopleSoft SAP Ellucian (SunGard HE) Talent2
Grants Management (pre-award)	Click Commerce eVisions/Cayuse Homegrown SAP
Learning Management Systems	Blackboard Desire2Learn Edvance360 Instructure Canvas Moodle Trust Moodlerooms Sakai Foundation
Student Information Systems	Campus Management Ellucian (Datatel) Homegrown ITS Jenzabar PeopleSoft SAP Ellucian (SunGard HE) Three Rivers Tribal

Introduction

Enterprise applications change slowly. The projects to implement them can be hugely expensive, require enormous willpower and focus, and represent a major disruption to the normal daily functioning of many areas of the institution. In relatively few areas do new systems offer serious or long-lasting competitive advantage. More often they are driven by the institution's need to stay afloat and to make progress, by legacy systems reaching end of life, by budget pressures, or by a realization that the institution has failed to keep pace with the rapidly changing world of technology-enabled activity that fuels student and other constituent expectations.

As a result, enterprise projects are undertaken relatively infrequently, which in turn can mean that when they do occur, they can trigger a seismic shift in how the business processes of the institution work and in how staff and faculty must function. Processes that have been in place for years and even decades may be swept aside. Terminology changes. Prior expertise in specific processes or technologies loses value. In short, such projects can be traumatic for the institutions undergoing them. It's therefore understandable that substantial resistance to such change can be encountered, or that in some past cases, institutions have even invested large sums to rewrite substantial portions of vendor software to minimize the extent of their institution's process changes.

It is good news, therefore, that in spite of the many challenges and difficulties cited by respondents to this survey, the sense of accomplishment and success is dominant. But such success does not happen by itself. The same factors that influence and drive success in other types of projects are equally important here—e.g., careful planning, clearly stated and appropriately vetted requirements and objectives, realistic expectations, executive commitment, effective working relationships across key organizations, adequate funding and staffing, excellent communication, and highly capable project management.

Institutional complexity and culture can have a dramatic impact on enterprise application projects, yet they are extraordinarily difficult to measure and compare. This survey attempted to capture some of this through questions on the supportiveness of the institutional environment, the effectiveness of current project-related processes, the willingness of the institution to change, and the effectiveness of relationships among and degree of autonomy across units.

Success is similarly hard to measure because it can vary hugely depending on one's perspective—a new system that seems a roaring success to a central business office can still be panned as onerous and unworkable by distributed units and hated by the help desk. Indeed, success is more often measured by collective perceptions than by hard metrics. Success is not just one thing but rather has multiple dimensions. This survey recognized that reality and posed questions both on the success assessments

by a variety of stakeholders and on specific outcomes such as functionality improvements, cost experience, and workload changes.

This report is essentially a guided tour of the survey results, searching for patterns and correlations, highlighting particular findings, and pointing out any surprises along the way. Of the 128 survey responses, 62 were for learning management systems and 42 for student information systems; since these two areas had the most responses by far, some tables in this report focus on these two areas, plus the aggregate of all five system areas. We attempt to place it all in the context of the general practice of implementing enterprise applications in higher education—a practice that seems to be increasingly mature and successful.

Findings

Why Were These Projects Undertaken?

Several survey questions probed the motivation for and objectives of these projects. Figure 1 summarizes respondents' motivations for changing software. Across all projects reported, the most frequent responses were a change to acquire better or additional functionality (31%) or to move from legacy or end-of-life software (28%); the latter was more prominent among student and financial system projects. Similarly, in the survey question on important objectives, where respondents could select all that applied, the most frequently mentioned objective was to acquire transformative new functionality. Combining that result with the other choice of "useful but not transformative functionality," functionality improvement accounted for almost 40% of all designations of important objectives.

Projects to upgrade existing software accounted for 16% of responses, and even among these a desire for improved or even transformative new functionality was cited as an important objective by 70% of the upgrade respondents.

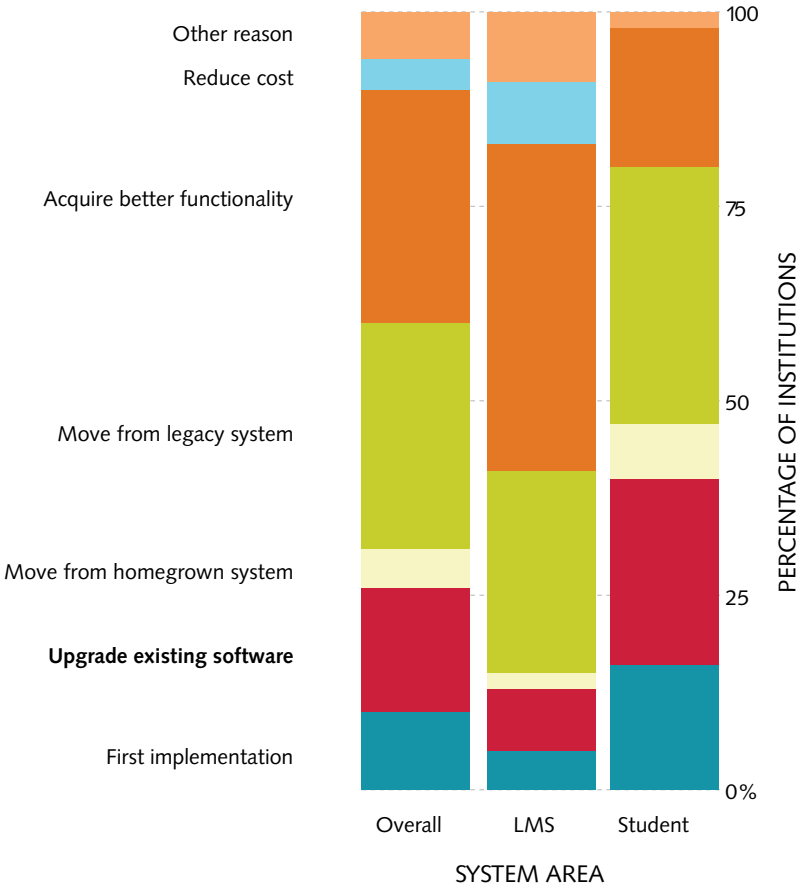

Figure 1. Motivation for Software Change

The high proportion of LMS projects in the survey overall is easy to understand: Not only is this a rapidly changing and increasingly important area from an institutional business perspective, but extraordinary vendor consolidation and open-source development have occurred in recent years, not to mention substantial increases in functionality—as shown in Figure 1, close to half of the LMS respondents (42%) selected improving functionality as best describing their project, as compared with the 31% overall.

Among student system projects, 11 were upgrades of existing software—a higher proportion of upgrade projects than in any other area. One might conclude that student system customers are electing to stay with their current providers and simply upgrade, whereas LMS customers are changing providers.

To determine which factors were important in approving and funding these projects, respondents were asked to rate 14 separate decision factors, ranging from very important to very unimportant (see Figure 2; responses of neutral, unimportant, and very unimportant are displayed to the left of zero). These included implementation and ongoing costs, functional scope, risks in implementing a new system, security/privacy concerns, prior experience with solution or provider, and the proposed system's ease of use. It is striking that of the 1,780 separate ratings received in this section, 72% identified the factor in question as important or very important. Only 8% characterized a factor as unimportant or very unimportant. This demonstrates that multiple factors loom large in planning for and obtaining support for an enterprise application implementation project.

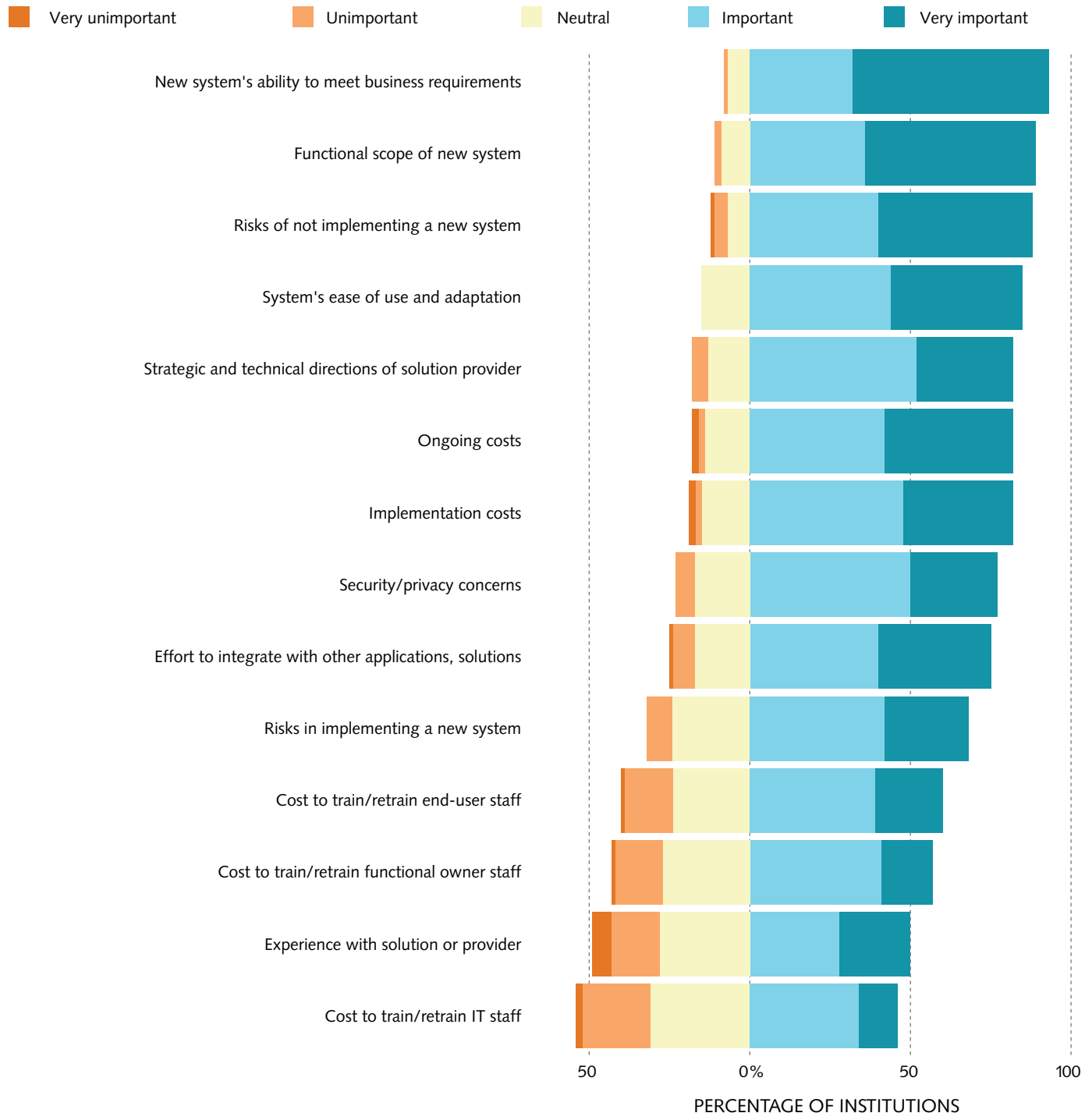

Figure 2. Factors in Project Approval

It is reassuring that ongoing costs were deemed important, since over the lifetime of an enterprise system, the aggregate support costs can easily be larger than the implementation costs yet were often ignored in past decisions.

Respondents were asked to describe in their own words the most important business area targets for their respective projects. Most responses were very much in keeping with what one would expect from an implementation in the particular system area. Some provided examples of how one might articulate the objectives on campus, illustrated by the following quote:

> Goal 1: Improve intervention response to academically at-risk students. Goal 2: Improve the accessibility of advising information. Goal 3: Reengineer more efficient institution-wide operational processes. Goal 4: Increase the availability of institutional research data for planning and assessment. Goal 5: Increase the effectiveness of the business, admissions, and advancement departments.

How Well Were Institutions Positioned for Their Projects?

The survey posed eight statements on "institutional environment" factors whose presence could promote or hinder project success, and respondents were asked to indicate their agreement or disagreement with each statement as it applied to their institution at the time of the project:

- Executive leadership provided the support needed.
- Project was a high priority for IT.
- Affected functional areas provided the support needed.
- Central IT and affected functional areas had experience working together successfully.
- There existed a cultural willingness to change at the institution.
- IT was highly centralized.
- Disparate units had a high degree of autonomy.
- Disparate units worked well together.

Three-quarters of all respondents "agreed" or "strongly agreed" that the positive elements were in place. The lowest percentages of strong agreement with success-promoting factors were for "disparate units worked well together" (13%) and "cultural willingness to change" (26%). In short, most of the projects reported in this survey occurred within a supportive institutional environment.

The statement "disparate units had a high degree of autonomy" received the expected stronger agreement from doctoral institutions than from all others (67% versus 46%). Subsequent analysis showed little direct correlation between most of the environmental factors and project costs or success rates, but across all classes of institutions, those with units having high degrees of autonomy took, on average, 43% longer for their implementation projects. This is as expected—more autonomous units can result in more complex and therefore more time-consuming decision-making processes, as well as more complex system and process requirements.

"Change is difficult no matter how well planned, communicated, executed, and supported. Prepare accordingly."

The survey also explored the extent to which respondent institutions had effective project and governance processes in place, e.g., change management, project management, and central IT's quality assurance and testing. These responses were not as strongly positive as those for institutional environment yet were still positive: While only 14% of the responses indicated that their project-related processes were very effective, another 48% reported effective processes. Overall, then, well over half of the institutions reported having effective or very effective project and IT governance processes in place. The lowest assessment was for business process reengineering/redesign, consistent with the general theme of challenges in business preparedness and business process change. Even here, 53 of the 128 responding institutions indicated that their process was at least effective, and for some very effective.

Counterintuitively, the survey did not find a meaningful connection between these various indicators of project readiness and the later indicators of project success. This may well be related to there being so many success stories in the survey and so few problematic projects. For most institutions in this survey, the picture that emerges is one of overall healthy positioning for enterprise projects, yet with a slightly wider range of success outcomes than one might expect if positioning were all that mattered. In short, other important factors are at work in determining success—a supportive institutional environment and effective project-related processes are necessary, but not sufficient.

"Involve your stakeholders early, help them with understanding needs/business case for change, get the decision to change to come from them… be ready to back up your initiative with resources to ensure success."

What Approaches Were Employed?

Customization and Locally Developed Code

Customization means different things to different people. The survey was careful to distinguish customization from locally developed code, providing this explanation: "Even when institutions are careful not to modify vendor-delivered code, they may develop (and therefore support) substantial local code (e.g., bolt-on functions, web front ends, system-to-system interfaces, 'user exits,' or other executable objects)." Respondents were then asked to select from five options to indicate the extent of customization and also the extent of locally developed code in their system implementations. The choices ranged from "none" to "extreme amount."

Overall, some customization and some locally developed code are quite common, but in most cases at a low level. Most institutions seem to have learned the lesson of keeping customizations and locally developed code to a minimum; of those that didn't customize, some wish they had (as indicated in open-ended responses in the survey). As expected, locally developed code is somewhat more prevalent than direct customization of the software vendor's code.

Figure 3 illustrates the contrast between the patterns for customization and for locally developed code. As noted earlier, only the LMS and student system areas had enough responses to be broken out separately here. We see on the left that although student system projects in general experienced more customization than LMS projects, the overall patterns of customization responses were similar across system areas. Figure 3 also shows on the right that in contrast, a sharper difference between student systems and other areas occurred in the percentage of respondents reporting significant amounts of locally developed code. Overall, the least amount of substantial customization occurs with LMSs (3%).

Research-intensive institutions reported the highest levels of customization. Non-U.S. institutions were next, followed by other doctoral/research institutions and special (arts, faith) institutions. These same relative rankings apply to locally developed code as well.

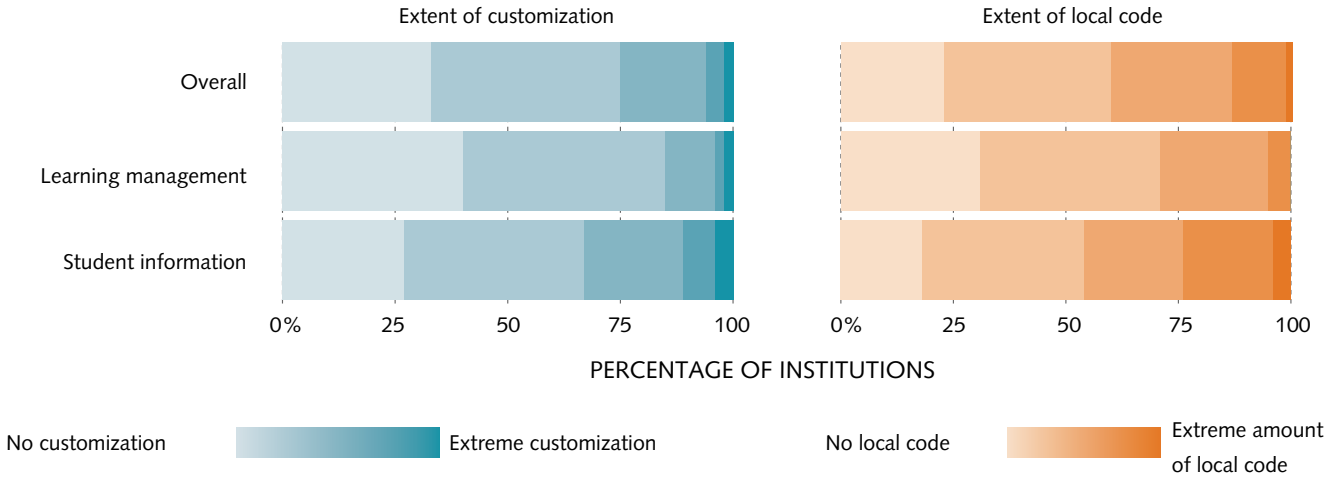

Figure 3. Extent of Customization and Local Code

Strategies Employed

Respondents were presented with a list of strategies and asked to check all used in their projects; the results are shown in Figure 4.

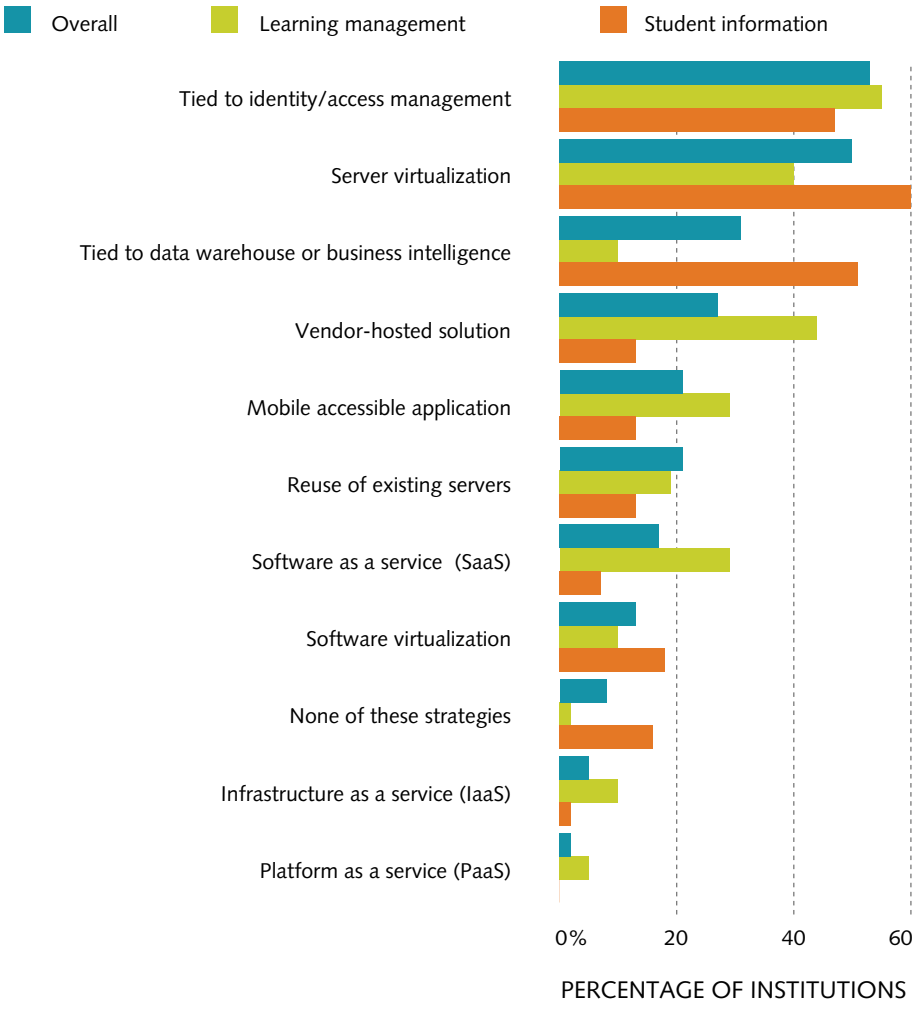

Figure 4. Strategies Used

Several observations are noteworthy:

- Half of all projects used server virtualization. If we exclude systems that seem to be outsourced, the proportion increases to 67%. This is a widely used approach, and on this survey question it ranked second only to "tied to identity/access management."
- LMS projects are vastly less likely than those in other system areas to be tied to data warehousing or business intelligence (10% versus 51% for all other areas combined), independent of whether the LMS was supported in-house or outsourced. The contrast with student systems is striking, especially given recent interest in using LMS data for student retention and assistance.

- LMS projects, both new system implementations and upgrades, are distinctly more likely to employ outsourcing strategies than other system areas.[4] Taking all of this question's indicators of outsourcing strategies together, the responses showed 63% of LMS projects using outsourcing strategies versus 18% for all other system areas, again in marked contrast to student systems. The overwhelming majority of LMS outsourcing responses also cited external hosting costs, consistent with outsourcing. The LMS outsourcing effect was most pronounced for vendor systems (notably Desire2Learn) but still present for open-source systems. It is not clear why the occurrence of LMS outsourcing strategies seemed so much higher in this survey than outsourcing levels reported in the annual Core Data Service, where only 23% of LMS implementations in the past five years were reported as outsourced. This could be an artifact of the real differences in question structure and focus between the two surveys.
- Consistent with a June 2011 ECAR report,[5] the extent of ties to identity/access management was heavily dependent on type of institution. In this survey 72% of U.S. doctoral institutions reported a link, but only 40% of AA and BA institutions said the same.

"Be willing to adopt changes to your original project plan; if the original assumption does not work, change it and try something else quickly."

Inclusion of Separate Capabilities as Part of the Project

The survey asked whether separate and generally independent capabilities were necessary to complete the application implementation being reported. Examples provided in the survey question included identity management, document management, desktop upgrades, and other infrastructure.

Among the 63% of respondents who indicated that yes, such additional capabilities were needed, 79% said that such work was included in the application implementation project's timeline and costs. This finding underscores the need to consider, budget for, and plan for all technical environment requirements for successful application implementation.

Of the 64 projects reported as including such independent capabilities, two-thirds experienced only minor resulting impact on project timeline and over half experienced only minor resulting impact on project cost. Only two experienced significant impact (on timeline).

What Challenges Were Encountered?

The challenges encountered were addressed by two survey questions. The first provided a set of predefined areas, with respondents asked to indicate any that constituted a significant challenge and/or a significant success (many were both). The second asked respondents to describe in their own words the greatest challenge they

faced and how they overcame it. These responses were quite illuminating, and several are provided later in this section.

As can be seen in Figure 5, user adaptability to change and user training were cited by the most respondents as challenges. None of the other challenges was cited by more than 50% of respondents, although data migration and project schedule came close.

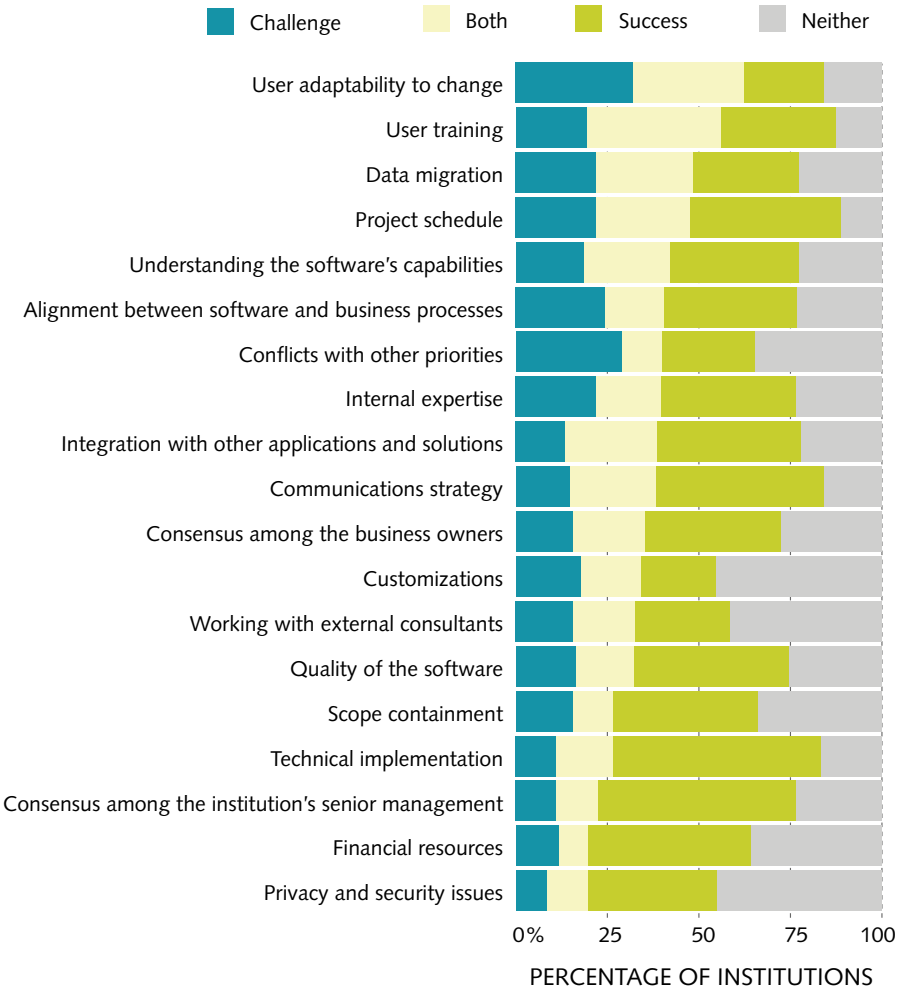

Figure 5. Significant Project Challenges and Successes

This figure reinforces points that arose throughout the survey—for example, the need for effective project processes and project management, and the importance of business and user readiness. Notable is the large number of significant successes that were *not* also significant challenges (37% of all responses), which is evidence of sound project preparedness by many institutions. Also note that the one area most often cited as a significant success but not a challenge was technical implementation. Enterprise projects evidently do not struggle as often in technical areas as they do with process, organization, management, and human inertia.

Variations by System Area, Solution Type, and Project Type

When we examine the challenge responses by system area, we find noticeably more reports of significant challenges with student system projects than with LMS projects—51% of possible responses versus 32% for LMSs. This was an across-the-board effect: Of the 19 potential challenge areas, not a single one was reported as a challenge for a greater percentage of LMS projects than of student system projects. The difference was most pronounced for software quality, with 56% of student system respondents citing this as a challenge versus only 16% for LMS projects—a surprising result, given that the student system marketplace is substantially more mature than that for LMSs.

Within new system implementations, open-source solutions saw substantially fewer significant challenges than vendor solutions (28% of possible challenge areas cited for open source versus 45% for vendor solutions). Given that LMS implementations in this study are mostly for open-source software, it's natural to ask whether LMS applications are inherently less difficult or whether the open-source factor dominates. Slicing the data more finely, we find that within LMS implementations, the open-source solutions present fewer challenges in almost every area. The only three areas in which open-source LMS solutions had a higher frequency of challenge reports than vendor LMSs were customizations, internal expertise, and privacy/security issues.

Upgrade projects reported slightly fewer challenges overall than implementation projects, yet several challenges were cited in a higher percentage of upgrade projects, namely, conflicts with other priorities (50% of projects), customizations (45%), software quality (35%), and scope containment (30%). That last number is unexpected—one would have anticipated scope containment being much more of an issue with initial system implementations.

Table 3. Selected Quotes from Respondents on Greatest Challenges and How They Met Them

Greatest Challenge	Solution
Our own modification of the software	Large repair project, new strategy—go back to central baseline as much as possible.
Keeping within scope	We ended up bending on a few things. Often times you don't have a full understanding when you embark on a project. It is important to have some flexibility.
Unwillingness of some staff in the central [business] office to delegate work functions during the project to other staff so they themselves could focus time on the project	Had to work through [business] office leadership to force delegation of duties within [that] office to get project work done in a timely manner.
Our college was very silo-based. This was the first project that brought the entire college together, it was really the first time we needed governance.	We leveraged lots of data (including EDUCAUSE's) on establishing a governance model, keeping users/managers engaged, etc.
Faculty's time to move content from [legacy LMS to new one]	We employed students to work with moving the content for faculty.
Getting people in the training room and keeping them there	Made training and crosswalks/configuration as much fun and as comfortable as possible.
Even with backfill, the effort required by all concerned to have the project be successful	Quite frankly, we powered through it.

How Successful Were These Projects?

Overall, the survey respondents reported very high levels of success. In the ratings of how each of 10 stakeholder groups viewed the project's outcome, 80% of all responses were "very successful" (50%) or "somewhat successful" (30%). In contrast, only 5% of all responses were "very unsuccessful" or "somewhat unsuccessful." Of course, we must remain aware that selection bias may be present, if those institutions with troubled or even failed projects declined to respond to the survey. A follow-up study focusing specifically on lessons from *unsuccessful* projects could be an excellent counterpart to the present study, since sharing failures is often more illuminating than sharing successes.

While we do not know how each respondent determined the assessments of their various stakeholder groups, it is informative to look at the differences reported across these groups, as shown in Figure 6. IT application area leaders and CIOs were the most likely to be reported as viewing the project as somewhat or very successful. Even end users seem fairly satisfied, with over 75% of the projects viewed as somewhat or very successful.

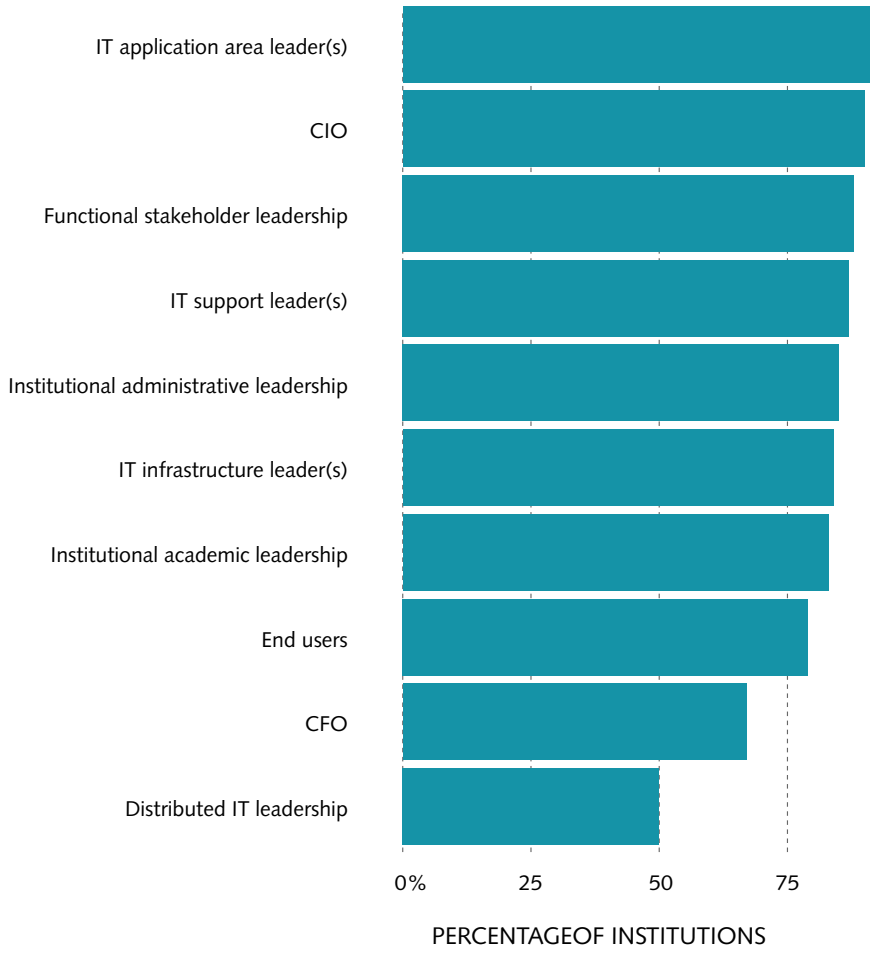

Figure 6. Stakeholders Who Viewed Projects as Successful

The survey also asked how successful each of eight project phases was; again, the reports were of high levels of success, with 81% of all responses being "very successful" (47%) or "somewhat successful" (34%). Quality assurance/testing and change management were the two areas *least* likely to be judged very successful.

To explore possible correlations of various factors with these success indicators, we constructed a consolidated "success score" for each institution across all the success-related responses—the 10 stakeholder area assessments and the 8 project phase ratings.[6] We then examined these overall success scores and their variation by system area, solution type (homegrown/vendor/open source), institutional classification, and new implementation versus upgrade:

- Overall, upgrade projects had higher average success scores and less variation than new system implementations.
- New student system implementations were more likely than other areas to be considered unsuccessful, accounting for half of the lowest 10% of consolidated success scores on this survey, yet the overall average success score for student projects was comparable to those for other areas.
- Institutional systems, associate, and bachelor's institutions report the highest success scores, both overall and separately within the LMS and student system categories.
- Insufficient data was available to detect statistically significant correlations between implementation cost and success score, although some may be present. It is likely that in some cases extra dollars spent can procure higher levels of resources or expertise (technical, business, project management), which can promote greater success, whereas in others poor management can lead both to less successful outcomes and to cost overruns.
  - The 8 vendor and 11 open-source LMSs with the highest range of success scores (4.8–5.0) were also noticeably more expensive than the average in their respective solution type groups. Nevertheless, across all other ranges of success scores for the remaining 43 LMS implementations, no discernible relationship to dollars spent was found.
  - Within student system projects, the only other area with enough data points to look for correlations, not even this weak connection between cost and success was found.

Respondents were also presented with the following nine areas and asked to indicate to what extent their expectations for each area were met:

- Hardware costs (implementation)
- Software costs (implementation)
- Number of FTE staff (implementation)
- Hardware costs (ongoing)

- Software costs (ongoing)
- Number of FTE staff (ongoing)
- Time to roll out final product
- Features and functionality of final product
- Primary business objectives

Two-thirds of all responses were "met expectations," which indicates that respondent institutions were very realistic about these projects. This is consistent with the earlier result of strong project readiness across most institutions. Only "time to roll out final product" had a noticeable number of "much higher than expected" responses. The other areas of disappointment were staff effort required for both implementation and ongoing support, and software costs. On the other hand, it was encouraging to see a number of instances (20%) in which the functionality of the new system and its ability to meet the primary business objectives exceeded expectations.

What Objectives Were Achieved?

Survey participants were presented with a list of 19 potential positive outcomes for their respective institutions as a result of these projects and were asked to what extent they agreed or disagreed with each statement. The results are shown in Figure 7. Overall, there were almost four times as many "agree" and "strongly agree" responses as there were "disagree" or "strongly disagree," indicating a large number of positive outcomes for most institutions. Improved service delivery to staff and students, enhanced support of the institution's mission, and reduced business risk had the most "strongly agree" responses.

Nevertheless, a couple of responses served to underscore a certain reality: Substantially more disagreement than agreement was found with the notions that IT or business workloads were reduced as a result of these projects. This, in spite of the fact that, according to our respondents, both IT and business productivity were higher as a result of the project. The inescapable conclusion is that there is more work to be done with the new system, which reminds us that new systems are rarely, if ever, simply one-for-one replacements for the functions of older systems. In almost all cases, new systems represent expanded functional scope and often expanded user populations compared with predecessor systems.

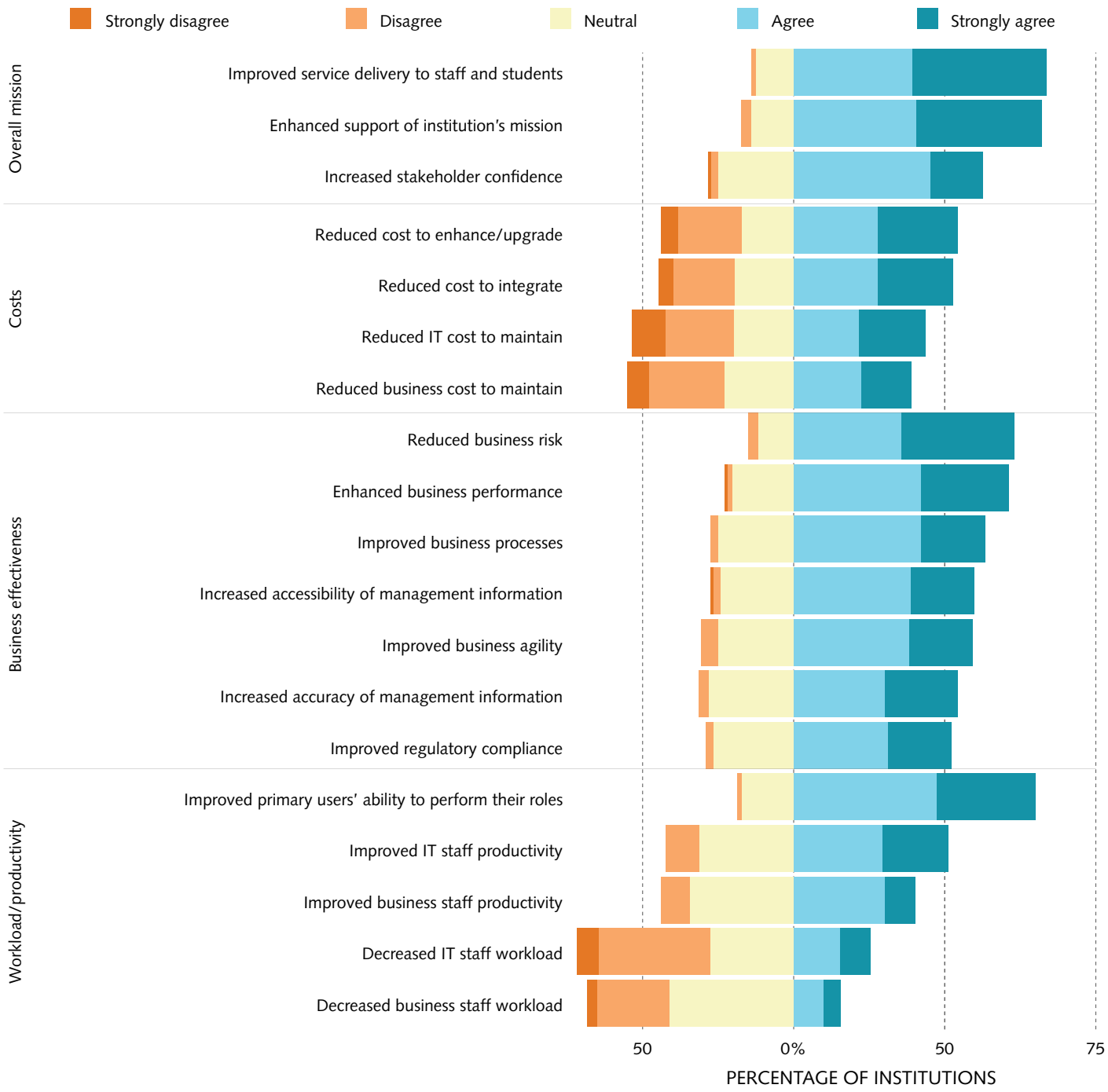

Figure 7. Project Outcomes

How Much Did These Projects Cost, in Dollars and Time?

As we saw in Table 1, extremely wide variation existed in the reported system implementation costs, even within system areas. Furthermore, this wide variation continues with ongoing costs, as can be seen in Table 6, which is the companion to Table 1 for annual operating costs.

Note also the reports of zero-cost new system implementations, four involving open-source LMS projects, and even one with outsourced vendor software. It would require additional study to confirm those results and, if confirmed, to learn how they were possible. Most likely certain costs were covered by other budgets that were not formally considered "project budgets"—something to remember when trying to compare costs across different institutions' approaches to project accounting.

Clear differences can be seen from the cost data in Table 1:

- The least expensive new implementation projects were overwhelmingly LMSs, yet one cost $5 million and another cost $1.6 million. Conversely, student system and financial system implementations were substantially higher in cost than average.
- As one might expect (and hope), upgrade projects were distinctly less costly than new system implementations, although there was still a wide range of costs—from $23,000 to $1.7 million.

The survey probed the different major components of project costs, and subcomponents of external costs. For those institutions that provided implementation costs in the three major categories, we show key results in Figure 8 and Table 4. With the exception of in-house, open-source projects, the dominant component of implementation costs is external costs (86%). For new implementations of in-house, open-source systems (these were mostly LMS or financial systems), internal compensation expense is more important, accounting for 54% of the costs. Our investigations showed that this difference between in-house, open-source projects and all others overshadowed any other cost distribution differences across system area.

Consider institutions A and B with roughly the same annual operating budget of $500 million. Both implement exactly the same vendor's student system over the same number of months. A's project is for multiple campuses—presumably a more challenging and expensive undertaking than B's single campus. A has twice as many students as B, which can drive up software license fees. It therefore would seem that A's implementation should cost more than B's, yet the reverse is true: A spends "only" $7.4 million on the project, while B spends 55% more—$11.5 million, or an additional $4.1 million. Why this surprising difference? It turns out that B is a research-intensive institution and A, while doctorate-granting, is not. (This case is based on two real institutional responses in the survey; the dollar figures have been altered by a common multiplier for confidentiality, but the ratios between the two institutions remain real and accurate.)

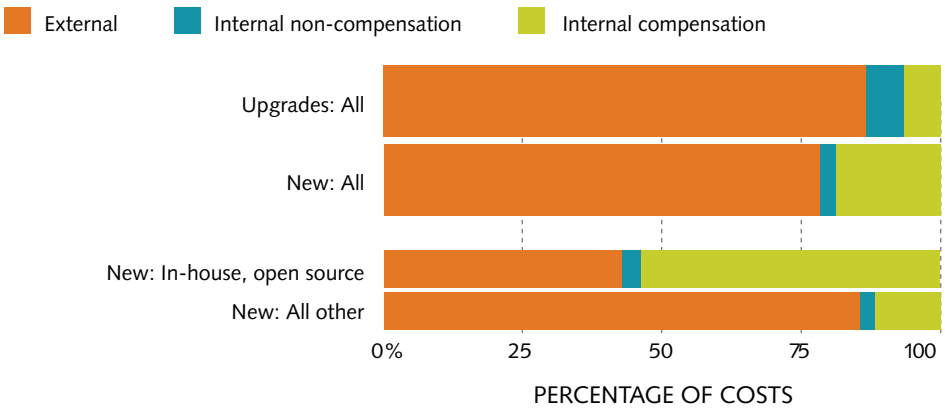

Figure 8. Average Project Cost Components

Table 4. Average Implementation Cost Components (dollars in thousands)

System Area	Number of Institutions	External	Internal Non-Comp	Internal Comp	Total
Upgrades: All	18	$386	$30	$29	$444
New: All	101	$1,976	$70	$473	$2,519
New: In-house, open source	19	$959	$77	$1,199	$2,235
New: All other	82	$2,209	$67	$301	$2,577

In anticipation of this dominance of external costs, the survey asked respondents to provide percentage distribution of external costs across five categories—hardware, software, hosting, staffing, and other. More than 100 institutions provided these breakdowns for single-primary-system projects. Initial inspection of the data showed that the distributions for LMS projects differed markedly from those in other system areas. In addition, the distributions for outsourced projects are understandably different from those for in-house projects. For these reasons, the results for LMS and non-LMS systems are presented separately in Figure 9.

Surprisingly, we found that the external staffing percentage, which was distinctly lower for LMS projects as compared with other systems, was lowest for in-house implementations of vendor packages.

For non-LMS projects:

- The largest single component of external costs is overwhelmingly staffing (consultants, services, contractors). This is true for both new system implementations and upgrade projects, for both in-house and most outsourced systems, and across vendor and open-source solution types—a fairly remarkable consistency.

- Software is the second largest component of many projects, though understandably much less for in-house, open-source projects and for upgrades. We observed that at least some software costs are still present even with in-house, open-source applications, most likely because other proprietary software (such as database) is used in conjunction with the open-source application.

Figure 9 shows percentage breakdowns *within* external costs. It's also worth considering their relationship to overall project costs, especially for the external staffing component. We find that of the $305 million in aggregate implementation costs reported in this survey for the 112 projects that also provided external cost breakdowns, $149 million (49%) was spent on external staffing and $44 million (14%) on software. Note that external staffing is also a common expense component of enterprise projects—of the projects that provided percentage breakdowns, only a fourth reported zero external staffing costs.

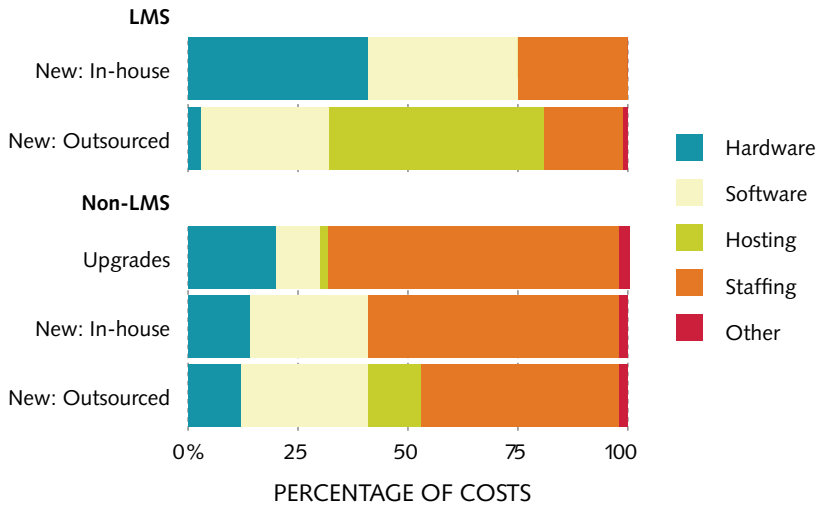

Figure 9. Average External Project Cost Distributions

Separately, we found that higher-cost projects experienced a substantially higher percentage of external staffing costs. This makes sense—for these large projects, institutions typically feel they must bring in external staff resources in order to get the job done. This is not simply a tautology, where the presence of external staffing is what makes the project costs large in the first place—the "higher-cost projects" would be large in cost compared to others in the survey even without the external staffing costs included.

We also examined total expenditures on system implementation projects within the context of overall institutional expenditures for each institution for which we had budget data (only U.S. institutions; no university systems). The findings seem

to indicate that some institutions have spent surprisingly large relative amounts on system implementations. We note that 1% of an institution's annual budget is a sizable investment in a single system, and as expected, most institutions surveyed (73%) did indeed report enterprise system implementation costs less than this—in some cases far less. Yet 19 of the 94 institutions for which we have annual institutional expense data reported project costs equivalent to 1%–5% of their annual institutional budget, and 7 reported even higher amounts, up to a surprising (and perhaps questionable) 14.6%.[7] One percent may sound small, but 1% of a $2 billion budget is $20 million.

Project Duration and Its Impact on Cost

The classic project management maxim is that one has three major interdependent constraints to manage—scope, time, and resources (such as dollars and staff). One cannot adjust one without affecting the others. One therefore expects project duration to be a factor in implementation costs. Suppose, for instance, that one has engaged outside consultants for the duration of a major implementation. The longer the project, the longer one must keep paying the consultants, and therefore the higher the project costs. And in fact we do observe that the most expensive projects reported do take substantial amounts of time to complete (1.5–3 years) and do involve substantial external staffing expense. Furthermore, at a summary level, the higher the project cost range, the longer the average project duration (see Table 5[8]). This relationship holds separately for the set of all LMS implementations and the set of all student system implementations, and overall for all projects combined. Yet when one examines the data within these groupings, the expected relationship between duration and cost is not found—some low-cost projects take much longer than vastly more expensive ones, and some higher-cost projects are below the median duration.

Table 5. Project Duration (months)

Category	Number of Institutions	Minimum	Median	Average	Maximum
All projects	112	1	14.0	15.5	65
New Implementations, by System Area					
Learning management	55	1	11.0	12.7	32
Open source	35	1	10.5	12.3	32
Vendor	19	3	11.0	13.3	28
Student information	23	7	20.0	21.3	48
All system areas	93	1	14.0	16.2	65
Upgrades					
Student information	10	3	13.5	13.7	21
All system areas	19	1	13.0	12.4	21
New Implementations, by Solution Type					
Open source	41	1	13.0	14.7	65
Vendor	51	1	14.0	17.2	48
New Implementations, by Cost Range					
$5 million or more	11	12	27.0	27.7	42
$0.5 million to less than $5 million	30	5	19.0	20.1	65
$0.1 million to less than $0.5 million	21	2	11.0	13.5	32
Less than $0.1 million	29	1	8.0	10.1	32

Several explanations are possible. The time factor could potentially work in both directions, and one respondent reported purposely taking a longer time for a project expressly so that their in-house staff could take on more of the project workload, thereby lessening the need to spend money on outside consultants.

When a system's "go live" actually occurs and when the project actually ends is sometimes a difficult question. Some new systems may go live in stages or be rolled out to users in phases over an extended period. Especially when LMSs were first introduced, years often passed before a majority of faculty and students were using the systems. Furthermore, as Jack McCredie and Dan Updegrove observed in their classic 1999 *CAUSE/EFFECT* article, "[an enterprise implementation] project will never be completely finished."[9] Therefore, this ECAR survey defined the project go-live date very carefully: "For purposes of this question, 'go live' refers to the time where the majority of intended users of this system were able to use all their authorized functions of the system being delivered as part of this implementation project." Yet some respondents reported new enterprise

application implementations—mostly LMSs—that took a surprisingly short 1–3 months. At the other end of the spectrum, although one very large project was reported as taking 18 months, an examination of that institution's website produced evidence that 2 years of extensive preparation took place prior to the formal project launch.

Here are several observations from the data on project durations:

- Upgrade projects do, as expected, tend to be shorter in duration than new system implementations, but the difference is surprisingly small, taking only 23% less time—an important factor to keep in mind when planning an upgrade.
- Just as LMS implementations are less expensive than those in other areas, they are also faster, needing 40% less time on average than student system implementations.
- On average, projects using outsourcing did not seem to be measurably shorter than the others.

"Maintain a consistent path through completion and don't stop. You can listen to complaints, but remind everyone of the timeline and support from the top of the organization."

What Are the Ongoing Costs of System Operation?

Ongoing costs were obtained in the survey using the same general structure as implementation costs: first the annual external, internal non-compensation, and internal compensation costs charged directly to the ongoing application budget, and then the percentage distribution of the external costs. An overview is shown in Table 6.

Table 6. Annual Ongoing Costs for Operation (dollars in thousands)

Category	Number of Institutions	Minimum	Median	Average	Maximum
All systems	116	$0	$100	$294	$3,816
New Implementations, by System Area					
Learning management	55	$0	$70	$129	$900
Open source	37	$0	$45	$101	$550
Vendor	18	$31	$108	$185	$900
Student information	31	$20	$233	$492	$3,816
All system areas	99	$0	$100	$296	$3,816
Upgrades					
Student information	10	$6	$293	$390	$1,800
All system areas	17	$6	$98	$280	$1,800
New Implementations, by Solution Type					
Open source	41	$0	$50	$191	$1,985
Vendor	55	$0	$170	$388	$3,816

The patterns follow those seen earlier with implementation costs. Open-source operating costs average lower than vendor solutions, but with a wide range in both cases. LMSs are the least costly systems to operate.

It was interesting to observe instances in which annual operating costs exceeded the implementation costs—every year the institution pays more to support the system than it did to implement it. Yet in some circumstances this could make sense, especially with outsourced systems—for instance, relatively low start-up charges from the SaaS provider yet sizable ongoing hosting fees. One might also question whether the "unfinished business" of system implementation is really continuing under the label of ongoing support. This can be the case with systems where a large number of ultimate users are not yet on board when, from an IT perspective, the system moves into production operation. LMSs, portals, institutional calendars, research grant proposal systems, and several others can share this characteristic.

The breakdown of major ongoing cost components is shown in Table 7 and Figure 10. In particular, Figure 10 shows the higher proportion of internal staff cost for open-source projects.

Table 7. Average Annual Ongoing Cost Components (dollars in thousands)

System Area	Number of Institutions	External	Internal Non-Comp	Internal Comp	Total
New Implementations (all)	99	$176	$14	$106	$296
Learning management	55	$96	$5	$27	$129
Student information	31	$324	$25	$144	$492
Upgrades (all)	17	$198	$64	$18	$280
Total (all systems)	116	$180	$21	$93	$294

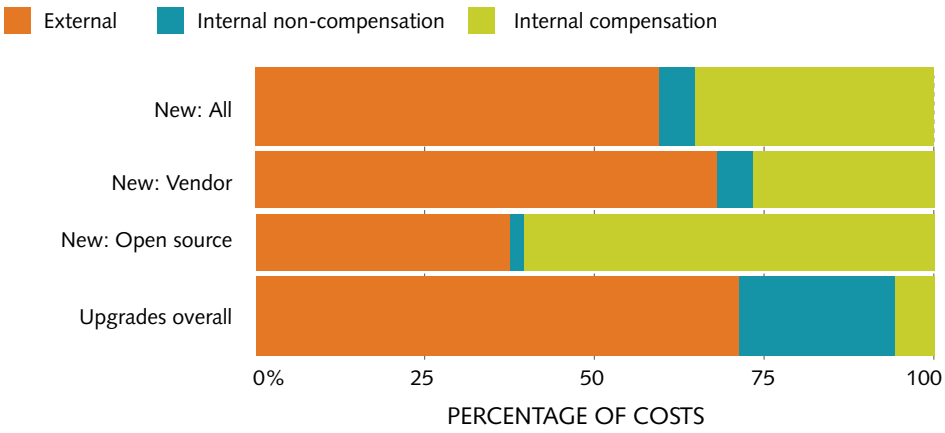

Figure 10. Ongoing Cost Components

As with implementation costs, the distribution of external costs is heavily dependent on the nature of the solution—outsourced systems will usually have substantial annual hosting fees, as opposed to the annual hardware costs seen with in-house systems. And once again, the patterns were different between LMSs and the other systems, as shown in Figure 11.[10]

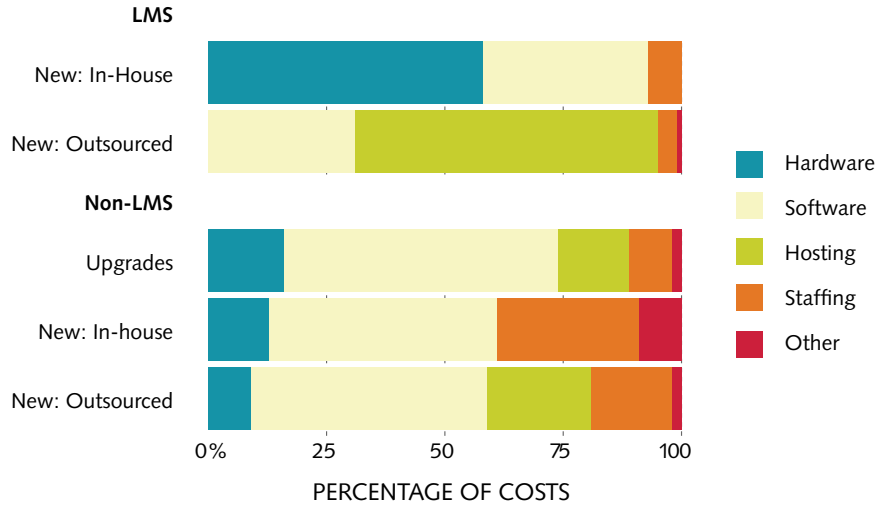

Figure 11. External Ongoing Cost Distributions

What Internal Staff FTE Are Needed for Implementation and Ongoing Support?

Survey respondents were asked to provide "the average number of internal staff FTE working on the core project team during the implementation period (from start date to end date)" and to do this separately for IT staff and for "other staff (e.g., business partner, functional area, or other non-IT internal staff)." As with implementation and ongoing costs, we found clear differences by system area, for both IT and non-IT staffing—financial and student system implementations are substantially more resource-intensive than other system areas, with LMSs the least so. The staffing results are displayed in Figure 12.[11]

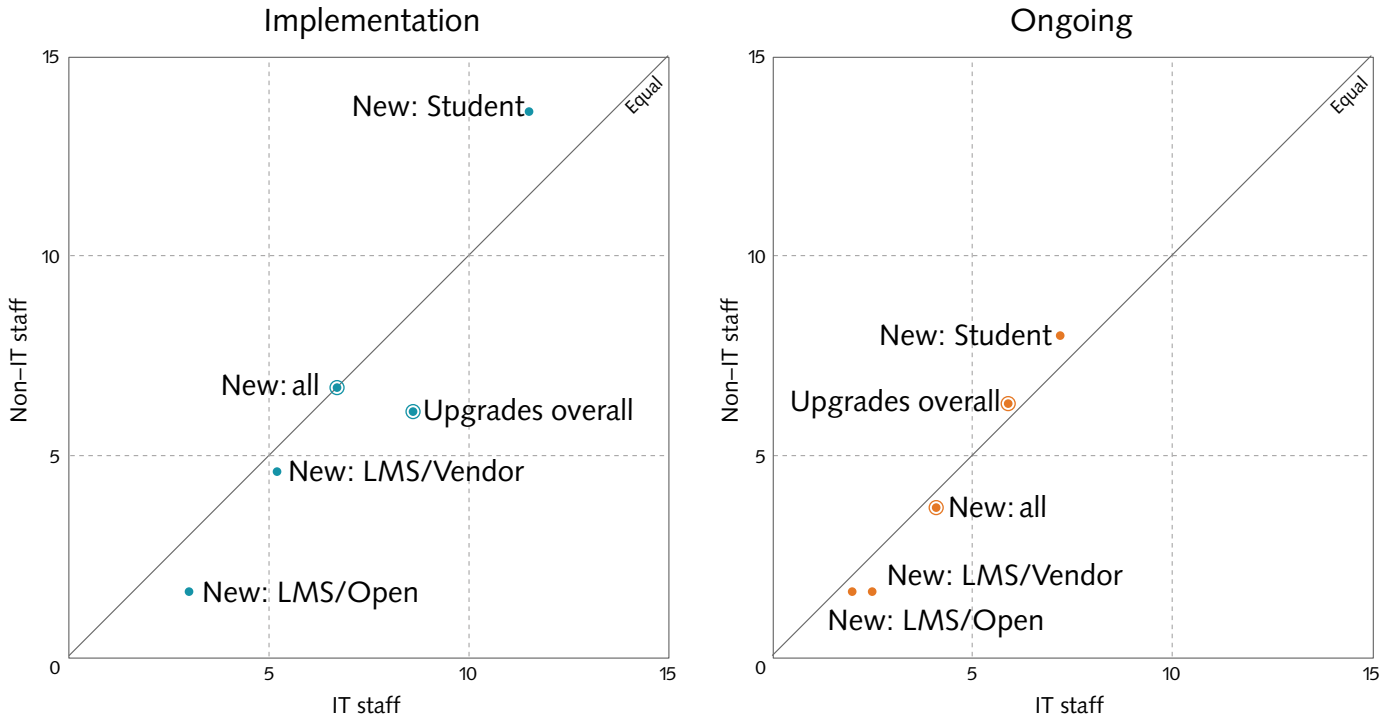

Figure 12. Average Internal Staffing FTE

A separate examination of average FTE by cost range produced the expected result—the more costly the project, the more staff are likely to be involved. Moreover, segmented by institutional classification, it is the research-intensive institutions and the systems of institutions that deploy the larger project staffs, by a wide margin.

An interesting pattern is that with few exceptions, the average numbers of IT and non-IT staff involved in projects are roughly comparable, as can be seen from Figure 12. Something similar can be seen with ongoing support, though we found one noticeable exception—the largest projects (implementation costs over $5 million) show almost twice as many IT staff as non-IT staff FTE involved in ongoing support.

Advice and Recommendations

> "The whole idea of an administrative information systems implementation makes my heart beat fast and my stomach churn. But these are not the feelings that come with the anticipation of a new love. They are the feelings that come with the anticipation of something I dread. And each new implementation—or, even, each substantial upgrade to a working product—brings this sense of dread."
>
> *Annie Stunden, then CIO, University of Wisconsin–Madison*[12]

While Annie Stunden's excellent 2006 *EDUCAUSE Review* article is positive overall, pointing out many factors that can enhance the chances for success in enterprise application implementations, the quote above from that article highlights the trepidation that can accompany the launching of such endeavors and the trauma they can inflict along the way. These are not easy projects.

Respondents to this ECAR survey had several opportunities to provide free-form comments and additional information, including lessons learned and advice for others. Recurring themes included the need for excellent planning, strong executive support, and experienced project management in both IT and business areas. Some strongly opposite points of view also were expressed, reflecting those institutions' varying experiences—e.g., whether it was effective to engage project management from a vendor.

The specific recommendations that emerged from the study can be seen as prepare—manage—deliver.

Prepare

- Take advantage of advice and experience available from peer institutions, from EDUCAUSE, from open-source communities, or from vendor user groups. But don't assume you can know everything before launching the project.
- Acknowledge the trade-offs between different approaches, e.g., the different levels of institutional control with in-house versus outsourced applications. Different approaches may be appropriate for different system areas, institutions, and situations.[13]
- Work to ensure that the key business areas involved are "project ready"—that they understand the discipline needed for success; that they are ready to provide decisions, work out compromises, and develop new processes as needed; and that they dedicate time from experienced, knowledgeable, and creative business staff to the project, ensuring that those individuals are not constantly called away from the project by business as usual back at the home office.
- Know and plan for your institution's own decision-making and process culture. The more stakeholders who must be consulted on project decisions, the longer

they typically deliberate, or the more varied the business processes across the institution, the more time and resources one must allocate to the project plan.

- Find the balance between unrealistically underestimating the time and resources needed versus creating a self-fulfilling prophecy that it will take much longer and cost much more.
- Contract wisely, whether for software, consulting, or other services. Retain flexibility to cancel parts of a bundled arrangement if they don't work out.
- Check references and validate a service provider's capabilities. A poorly chosen implementation partner or outsourcer can sink a project; a well-chosen one can help ensure success.
- Be up-front with stakeholders and users about how disruptive it will be to move to a very different new system. Set and manage realistic expectations.
- Secure strong support for the project from executive and functional office leadership.

Manage

- Collect, record, check for feasibility, prioritize, and make available the detailed requirements for the system. Manage to that list. Be prepared for new requirements to surface during the project and establish a clear process for dealing with them.
- Ensure that the project has a strong project management discipline and an excellent project manager.
- Develop positive, productive working relationships with all parties, both internal and external, to facilitate dealing with issues quickly and effectively as they arise.
- Take the time to learn the software's capabilities in detail and then to align business processes and software, which can involve substantial business process redesign.
- Temper a strong push forward to completion with a measure of flexibility to accommodate the unexpected. But don't stop pushing.
- Accept that any programming by your staff or others working on the project results in code that you or contractors will need to support long term, whether you think of it as "customization" or not. To minimize ongoing support costs, minimize such custom/local programming, changing business processes as needed.
- Insist on strong change management, both in the IT sense of having effective processes for managing software, system, and process changes, and in the "organizational development" sense of assisting people and organizations with the move into new roles, processes, terminology, and system functions.

- Don't delay making changes in project structure or makeup if the project is not going well.
- Communicate, communicate, communicate—get as much end-user buy-in as possible and involve end users wherever you can.

Deliver

- Data quality and flexible data access matter a great deal. Pay attention to data cleanup, data migration, and query and reporting capabilities.
- Plan on more training and assistance than you thought necessary.
- Engage pilot testers to validate functionality and performance of the system being delivered.
- Allocate resources to support the end users as they begin to work with the new system and new processes.

As one respondent stated, "Involving, managing the expectation, and delivering to the expectation of the end user is the golden rule for the success of such projects."

Separately, the survey results also highlighted several interesting trends to watch, as open-source solutions continue to gain ground in certain areas, as vendor consolidation continues, as applications move increasingly to the cloud, as mobile technologies impact these applications, and as expectations for the functional richness of enterprise applications continue to rise.

If there is a dominant lesson to be gleaned from this study, it may be this: Many factors exist that can contribute to project success and ongoing cost-effectiveness, and it's worth paying attention to all of them. That may sound like bad news, but appropriate consideration for those many factors seems to be making its way into the organizational DNA—the overall impression that emerges from this study is an encouraging one, one of increased sophistication among institutions in preparing for, mounting, and executing such projects, with overall high levels of success.

Methodology

The ECAR ERP survey was conducted in September–October 2012. Some survey questions were developed in cooperation with the Council of Australian University Directors of Information Technology (CAUDIT), which sought EDUCAUSE member involvement in a study it was conducting internationally. EDUCAUSE and CAUDIT have shared the data from this survey, but each organization has analyzed it independently. The ECAR survey was sent to 435 EDUCAUSE member institutions that had indicated on the 2012 Core Data Survey Module 8 that they had implemented a system in one of the system focus areas within the prior five years. In addition, the survey was sent to a stratified sample of 538 other institutions that had not completed CDS Module 8 for 2012, so we did not know in advance whether they had implemented a system of interest. Total sample size was 973 institutions.

A total of 128 EDUCAUSE members responded to the survey. Of these, three were identified as implementations of ancillary, not primary, systems, and one was a project encompassing three separate primary systems of interest being implemented concurrently—these four responses are excluded from certain analyses. Twenty of the projects were upgrades to a newer version of existing software; since upgrades differ in some respects from new system implementations, they are broken out separately in certain analyses. Three of the reports were of new homegrown systems being implemented (two student systems and one grants management system)—too few to break out separately in any of the tables.

Participating Institutions

Table 8 summarizes participating institutions by classification and control. Classification groupings used in this report are based on the 2010 Carnegie Classifications.

Table 8. Survey Respondents by Carnegie Class and Control

Institutional Classification	Public	Private	For Profit	N/A	Total
Research intensive	8	5	—	—	13
Other doctoral	9	3	—	—	12
Master's	10	13	1	—	24
Bachelor's	2	25	—	—	27
Associate's	16	—	1	—	17
Special—medical/health	1	2	—	—	3
Special—arts/faith	1	3	—	—	4
Systems with research university	5	—	—	—	5
Community college systems	2	—	—	—	2
Canadian institutions	6	—	—	2	8
Other international	3	—	—	10	13
Total	**63**	**51**	**2**	**12**	**128**

Notes

1. EDUCAUSE recognizes that many institutions have multiple student-related systems, multiple financial systems, and multiple HR systems. The annual Core Data Service survey has traditionally focused on the one primary system in each area, as does this study.

2. One must, of course, be wary of overinterpretation—those with unsuccessful, problematic, or even failed projects may have disproportionately failed to respond to the survey.

3. Some respondent institutions did not answer all survey questions. As a result, slightly different subsets of the 128 institution responses are represented in different tables and charts in this report. Areas with small numbers of responses are often omitted from tables. And yes, there were five reports of zero-cost implementations, four of which were open-source LMS projects, and the other was an outsourced HR system.

4. The survey did not explicitly ask whether the system was externally hosted in ongoing operation, but it did include four questions related to outsourcing strategies, and several others (e.g., reuse of existing servers) that point to in-house operation. The cost questions also included a category for external hosting, both implementation and ongoing. For most cases, these several items collectively were sufficient to classify a particular response as in-house or outsourced. Even if the institution reported zero "external hosting" expense and did not indicate any outsourcing strategy, but the system being implemented is only offered by the vendor as SaaS, then that system is deemed to be outsourced. Nine cases—all student systems—had indications pointing equally to both in-house and outsourcing; these were counted as "mixed," since even in-house systems can involve externally hosted components, and vice versa.

5. Mark Sheehan, Cedric Bennett, Pam Arroway, Susan Grajek, Judith A. Pirani, and Ronald Yanosky, *Identity Management in Higher Education, 2011 Report*, Research Study (Boulder, CO: EDUCAUSE Center for Applied Research, June 1, 2011), available from http://www.educause.edu/ecar.

6. We assigned a numeric value to each response, from "very unsuccessful" = 1 to "very successful" = 5. "Not applicable or don't know" was excluded from all calculations. Each institution's overall success score was the simple average of their nonzero responses over all 18 indicators of success—the 10 stakeholder views and the 8 project phase assessments. These scores ranged from a low of 1.0 to a high of 5.0, with an average of 4.3 and standard deviation of 0.73.

7. For this analysis, institutional budget and expense data were taken from IPEDS (http://nces.ed.gov/ipeds/).

8. Table 5 does not include projects started prior to 2007, for which the start date was not obtained in the survey and for which the project duration is therefore unknown.

9. Jack McCredie and Dan Updegrove, "Enterprise System Implementations: Lessons from the Trenches," *CAUSE/EFFECT* 22, no. 4, 1999, http://net.educause.edu/ir/library/html/cem/cem99/cem9943.html.

10. Several instances occurred of outsourced systems with zero external hosting costs. These respondents likely thought of the SaaS charges as software costs and reported them in the latter category.

11. Some survey responses that seemed unlikely to be correct (e.g., 100 FTE) were excluded from these charts.

12. Annie Stunden, "The Toughest IT Challenge," *EDUCAUSE Review* 41, no. 3 (May/June 2006): 32–42, http://www.educause.edu/ero/article/toughest-it-challenge.

13. For a recent discussion and framework for different approaches to procuring IT services, with a special emphasis on different types of communities, see Bradley Wheeler and James L. Hilton, "The Marketecture of Community," *EDUCAUSE Review Online* (November 1, 2012), http://www.educause.edu/ero/article/marketecture-community.