

Does *Best Practice* Makes Perfect?

Fitting Off-the-Shelf Applications to Meet *Your* Needs

Introduction

by Gabriele Piccoli 3

User Experiences with the Implementation and Use of Application Package Software

by Erica Wagner and Sue Newell 5

Application Package Software: The Promise Vs. Reality

by William M. Ulrich 13

It's a New World ... So Grab Your Old Weapons

by Gabriele Piccoli 20

Application Package Software Survey Data

Collected by Cutter Consortium 21

About Cutter Benchmark Review

With *Cutter Benchmark Review*, you get in-depth statistics, analysis, and advice on the hottest IT trends and challenges that IT professionals like you face each day.

We Do the Research for You

CBR gives you high-level analyses of the current trends and hot-button issues you face. You'll discover the strategies and tactics of companies worldwide and, most importantly, where they're succeeding and failing.

Your Access to the Experts

Each issue of *CBR* focuses on a different topic, presenting fresh empirical data describing the current landscape. But beyond the statistics, you'll get insight into what the findings mean. Analysis from two different viewpoints — that of a distinguished academic and also an expert practitioner — plus future predictions and their potential impact from editor Gabriele Piccoli, provides you with a framework in which to apply the data to your situation. This is insider information you can't afford to miss.

With *CBR*, your organization benefits from the unique opportunity to glean insight from the cutting-edge ideas being studied in some of the world's leading educational institutions, as well

as the perspective that only seasoned experts with hands-on experience solving real problems in real enterprises can provide.

CBR gives you a unique opportunity to:

- Look critically, objectively, and through a variety of lenses at business technology issues impacting your enterprise
- Understand how your organization compares to others in your industry and/or around the world
- Understand the potential impact of today's trends on the future health and direction of your company
- Identify key IT initiatives that are worth pursuing and — just as importantly — those you should avoid
- Support your IT initiatives with unbiased research and judicious statistical analysis

Nowhere else will you find a marriage of the academic and practitioner perspectives like you get in *CBR*. Add to this the interpretation and key take-away lessons from Gabriele Piccoli, and you'll find *CBR* is a resource unlike any other available today.

Cutter Benchmark Review® is published 12 times a year by Cutter Information LLC, 37 Broadway, Suite 1, Arlington, MA 02474-5552 (Tel: +1 781 648 8700; Fax: +1 781 648 1950; E-mail: service@cutter.com; Web site: www.cutter.com).

Publisher: Karen Fine Coburn;
Group Publisher: Kim Leonard
(E-mail: kleonard@cutter.com);
Managing Editor: Cindy Swain
(E-mail: cswain@cutter.com);
Production Editor: Pamela Shalit
(E-mail: pshalit@cutter.com).

©2006 by Cutter Information LLC.
All rights reserved. *Cutter Benchmark Review*® is a trademark of Cutter Information LLC. No material in this publication may be reproduced or distributed without written permission from the publisher. Unauthorized reproduction in any form, including photocopying, faxing, image scanning, and downloading electronic copies, is against the law. Reprints make an excellent training tool. For information about reprints and/or back issues of Cutter Consortium publications, call +1 781 648 8700 or e-mail service@cutter.com.

Subscription rates are US \$477 a year in North America, US \$577 elsewhere, payable to Cutter Information LLC. Reprints, bulk purchases, past issues, and multiple subscription and site license rates are available on request.



From the Editor, Gabriele Piccoli

Does *Best Practice* Make Perfect? Fitting Off-the-Shelf Applications to Meet *Your* Needs

I am just now wrapping up writing a book (which turned out to be a much more involved undertaking than I had planned, but let's not rehash the wisdom of questionable decisions made in the past). The target of the book is the required information systems course for MBAs — you know, the course that a few decades into the Information Age happens to be one of the most important ones in the modern MBA curriculum? That seemingly simple fact, however, is lost on many masters students, who come into the semester thinking they don't need to learn all that techno-mumbo-jumbo. Thus, the critical success factor of any textbook of the "information systems for non-IT managers" kind is its ability to "enroll" the student, as a very good friend of mine would put it. In other words, the book must be good at "making the case" for its own existence. I have a feeling of *déjà vu* here; where have I heard of this problem before?!

Last week, I was working on the chapter titled "Getting IT Done" — a chapter that focuses on the process by which information systems come to be in the organization (i.e., systems design and development, systems selection and acquisition). That chapter benefited greatly from this month's *CBR* survey, as I was able to use fresh data to make the case that the way information systems come to be in modern organizations is changing — and that managers need to be more sophisticated and more involved than ever. We see much less of the dichotomy traditionally captured in the "make or buy" decisions. Rather, we are purchasing more and more off-the-shelf applications that subsequently go through a configuration and customization process. In the book, I call this the "buy and make" process.

OK, enough about my book, let's get to the important stuff: this installment of *CBR*! The idea for this issue was seeded by the many discussions I had with one of the authors (and my colleague here at Cornell) Erica Wagner. Erica has some very creative ideas and some strong opinions about software, particularly large-scale

applications (e.g., ERP), having done much of her work studying their implementation. I have seen her doing the "best for whom?" routine a few times, in which she raises some very interesting challenges to the notion of best practice software. The best-for-whom routine, in addition to being interesting and stimulating, is a perfect fit with *CBR*, where we feel quite comfortable having and voicing opinions.

Moreover, because here at *CBR* we like to back up our opinions with evidence and fresh data, Cutter issued a survey designed to take stock of user experiences with the implementation and use of application package software. We then invited a team of experts to comment on the results. Our academic contribution for this installment is provided by Erica Wagner, Assistant Professor of Information Systems at the School of Hotel Administration at Cornell University (USA), and Sue Newell, Cammarata Professor of Management in the Department of Management at Bentley College (USA). Both Erica and Sue do substantial field research on the design, development, and implementation of large-scale software applications. Providing our view from the field is Bill Ulrich, Senior Consultant with the Cutter Consortium Business-IT Strategies and Enterprise Architecture practices. Bill is also the President of Tactical Strategy Group, Inc., and has more than 25 years' experience advising a wide range of organizations on migration and transformation projects, having worked with *Fortune* 1000 companies, government agencies, and high-tech firms.

In this issue of *CBR*, Erica and Sue bring to bear their years of studying enterprise systems (ES) and best practice software implementations to frame the survey. They use the survey as a springboard to draw conclusions about the opportunities and pitfalls presented by the buy-and-make paradigm. But, in good old Cutter style, they don't shy away from challenging vendor (and executive!) rhetoric. I particularly draw to your attention their discussion of the limitations of the best practice

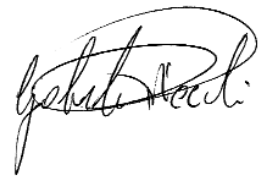
software concept. Much negotiation and many tradeoffs are required during the deployment and implementation of large, highly integrated applications like enterprise systems. For this reason, it is very dangerous to fall prey to the overly simplifying “transform the organization” rhetoric. Erica and Sue conclude by presenting six actionable guidelines that I am sure you will find useful.

Bill then does an outstanding job of systematically analyzing the survey and painting a comprehensive picture of the experience our respondents have had with application package software. I specifically draw to your attention his discussion of the implementation and integration challenges that these packages bring about. Bill concludes by summarizing the findings of the survey — a nice touch for busy readers and a feature that perhaps we should have in every issue — and contributing a set of broad recommendations.

The buy-and-make paradigm to systems development is here to stay. Today’s organizations have access to an unprecedented array of off-the-shelf applications seemingly addressing almost every single one of their

software needs. But the truth is that organizations today are as unique as they have ever been (if not more), and purchasing prepackaged applications requires (often significant) adjustments both to the software and within the organization. The result? Get ready for more of the same: hard-to-establish systems requirements, extensive negotiations, and significant need for change management during and after systems implementation.

I hope that you will find the ideas and suggestions in this issue of *Cutter Benchmark Review* to be helpful as you continue to fearlessly navigate this brave, new, buy-and-make landscape.



— Gabriele Piccoli, Editor,
Cutter Benchmark Review

THIS MONTH'S CONTRIBUTORS



Erica Wagner

Assistant Professor, Information Systems, School of Hotel Administration, Cornell University



Sue Newell,

Cammarata Professor of Management, Bentley College; Visiting Professor of Management at Royal Holloway, University of London



William M. Ulrich

President and Founder, Tactical Strategy Group, Inc.; Senior Consultant, Cutter Consortium

COMING NEXT MONTH

Service-Oriented Architecture

AN ACADEMIC PERSPECTIVE

Andy Schwarz, Assistant Professor, Information Systems and Decision Sciences Department, E.J. Ourso College of Business, Louisiana State University (LSU); Rudy Hirschheim, Ourso Family Distinguished Professor, Information Systems and Decision Sciences Department, E.J. Ourso College of Business, LSU

A VIEW FROM THE FIELD

Michael Rosen, Director, Enterprise Architecture Practice, Cutter Consortium



Erica Wagner, Assistant Professor, Information Systems, School of Hotel Administration, Cornell University; and Sue Newell, Cammarata Professor of Management, Bentley College

User Experiences with the Implementation and Use of Application Package Software

INTRODUCTION

In the past, companies have designed systems to support work activities unique to their organization. Such systems were implemented at the level of department or business function in order to effectively meet specific needs. However, this functional approach to development often led to the creation of “islands of automation,” where a depth of information about a single domain was available through a standalone data repository, but there was little or no potential for sharing information across the firm. Furthermore, isolated development led to disparate data formats, multiple operating systems, and various interpretations of data. These variations contributed to a systems environment that was difficult to maintain, made integration across business processes inefficient, and failed to meet user expectations.

This situation has changed dramatically, with companies moving from custom-made systems to configurable packaged enterprise software. These products supposedly embed best practices so that, through their adoption, the firm receives the gold standard for accomplishing particular business processes. This is attractive to businesses not only because of the ability to benchmark current processes against those designed into the software templates, but also because packaged software promises to cut down on inhouse IT development resources by offering a prebuilt framework around which to make configuration choices. Increasingly, vendors have begun to offer preconfigured packages for different industries and organizations of different sizes (e.g., packages specially designed for SMEs). This has been done to speed implementations by reducing the complexity of the configuration process, and it is beneficial for the vendor. For example, Oracle took its government/public-sector product and retooled it as a higher-education solution. As a result, Oracle successfully entered an emerging market.

When choosing packaged software products, adopting organizations are encouraged to first install the “vanilla”

version of the software — that is, with minimal or no customizations to the actual software program. Doing so is expected to ensure benchmarking standards and save unnecessary local development. Organizations are then encouraged to “configure” the software from a catalog of options and then to simply transfer existing data and processes to mirror those supported by the software. Consequently, buying instead of building software means that the normative practice is to change organizational processes and structures to match the vanilla design supported by the software. In addition, organizations must be willing to upgrade the software at regular intervals if support by the vendor is to continue (vendors argue that such upgrades are necessary to ensure that the market is supported with a state-of-the-art product).

Buying instead of building software means that the normative practice is to change organizational processes and structures to match the vanilla design supported by the software.

Enterprise resource planning (ERP) is the most prominent packaged software of this kind and arguably the most popular of the 20th century. The market reach of ERP is mirrored in the Cutter survey; 70% of responding organizations have adopted such a product (see Graph 1 in the Survey Data section starting on page 21). Historically, ERP products were confined to support system modules, but today they are being extended to include options for customer, supply chain, and human resource management — a trend that is evident from the survey. In this article, we use the generic “enterprise system” (ES) as an umbrella term to reference packaged software products that are designed to support different business processes.

Benefits typically associated with adopting an ES are that it can support integration of business information

across an organization, thus potentially enabling smoother and more efficient workflows around business processes. Yet evidence mounts as to the difficulties of achieving these benefits. The basic premise that an ES product can effectively meet the diverse needs of various organizations is questioned, as is the extent to which their sale in the marketplace enhances competitive advantage. This latter point relates to the question of whether the best practices that are embedded in the software are indeed the best choice, a point we will discuss later. Here, we draw upon the data collected from the Cutter benchmarking survey of ES package adoption and implementation to consider these issues.

ES DEPLOYMENT: PARTIAL VS. FULL DEPLOYMENT

The Cutter survey results support the notion that ES in general are difficult to exploit fully, but nevertheless, organizational actors are working with the products. Only 28% of the organizations have fully deployed an application package meant to support business processes (see Graph 2). Yet only 7% have failed to deploy or have deployed and then later pulled their software product out of production. This means that the majority of those who purchased an ES are in production with the software but are operating at less than full deployment. One reason for this are escalation costs related to complex software projects, where it becomes expensive quickly and the implementing organization is unlikely to completely pull the plug — so they try to make it work in spite of the difficulty associated with its implementation. Thus, while most of the responding organizations are operating an ES, over 90% of those that implemented one claim the organizational changes associated with implementation had a degree of difficulty (see Graph 11).

Bill Ulrich's article (beginning on page 13) highlights this finding as well, and he argues that management should adjust its expectations and realize that this is the way things are when it comes to ES deployment and use. We agree and argue that practitioners need support that extends beyond implementation if they are to manage these behemoth project initiatives. Practitioners should take solace that nearly 95% of the organizations represented in the survey have managed to create a system that is in production. Too often, when we visit organizations and talk with project teams trying to implement ES products, we hear disappointment and failure stories because they were unable to fully deploy the product, to realize every last benefit of the software. Evidence in this survey suggests that it is standard practice to deploy rather than give up — but to deploy

with less than what one might have initially hoped is possible. Perhaps not surprisingly, the findings also indicate that when less functionality is deployed, fewer goals are satisfied. For us, this indicates that expectations can be raised over time. It is important for management to persevere after deployment in order to exploit more of the functionality from the system.

IMPLEMENTING ES: IMPROVING EFFICIENCY VS. FOLLOWING THE COMPETITION

The survey results suggest that internal efficiency factors are more important than external competitor factors in influencing adoption decisions (see Graph 4). Many respondents do not feel that their organization chose to implement an ES as a way of benchmarking against the competition, that is, "keeping up with the Joneses" so as not to lose competitive position. Yet when we've asked practitioners why ERP, the repeated response is a blank stare followed by the statement, "No one builds their own systems anymore." This indicates two things. First, that humans like to attribute decision making to a rational assessment of needs, rather than to a desire to be like others. Second, it points to the "ERP bandwagon" effect, where we hop on board because there doesn't seem to be another wagon coming along anytime soon that can get us where we need to go.

ERP advocates have done an exceptional job at clearing the field of viable options and making it appear as though enterprise systems are a prerequisite for business success. This is not to discount our respondents' perceptions that they are motivated by internal efficiency factors more than external competition, but rather to offer an alternative interpretation based on our knowledge of the management "fads and fashions" literature. This literature indicates that the copycat trend is pervasive when it comes to management ideas and is illustrated by data on a diverse range of new technologies (including IT) that demonstrate an S-shaped adoption curve: innovative early adopters being copied by an increasing number of followers, leaving a few laggards or late adopters who refuse to "follow the fashion." The BlackBerry, for example, is indeed a nifty tool for accessing e-mail, but it certainly isn't the only tool, and yet its sales to businesspeople far exceed other brands. We see this in all kinds of consumer products, where trendsetters are followed by the masses, who essentially adopt to try and "be like" them.

Adopting technologies because others have done so is not necessarily a bad thing. Indeed, one could ask the

question, why adopt a best practice software if only to improve organizational performance and secure data, when inherent within an ES is the idea that one is connecting up with state-of-the-art practices as defined by others? Fashions offer advantages since they can provide powerful symbols for efficiency and legitimacy. In addition, ERP became a management fashion at a point in our history when issues of globalization dominated. The end of the 20th century brought with it the dot-com era, the rise of the Internet, outsourcing, and global competition. No longer did it feel safe to opt out of that which others were opting in to — in this case, ERP. What the management fashion literature reminds us, however, is that new technologies can diffuse very widely even though they are not necessarily more technically efficient than preceding technologies. For example, many of us remember the Beta versus VHS video recorder choice; the fact that VHS “won” the consumer vote was despite the fact that it was actually technically inferior to the Beta technology from Sony. Indeed, the very notion of “best practice” is inherently problematic, and we argue that what is actually best is determined by context and can never be captured by industry-wide best practice solutions because of the uniqueness of each organization and its history, environment, culture, and so forth. We next consider in more detail this issue of best practice.

SOFTWARE “BEST PRACTICE”: A PROBLEMATIC CONCEPT

The idea that adopting the package would enable the organization to adopt best practices is strong, with 70% of respondents citing this as a primary or important motivator (see Graph 3). Nevertheless, respondents have varied ideas of exactly what the concept of best practice refers to, suggesting ambiguity of this term. As illustrated in Graph 5, only 39% of our respondents believe that best practices have to be external practices, indicating that if an internal practice is perceived as superior to the proposed external practice, then one might choose to retain this internal practice as the best choice. Thus, we can define software-based best practices as:

Superior business processes from both within and outside an organization that are applied in order to create routine uses of knowledge. These practices are judged to be superior by those making the software and are then included in a catalog of software templates from which one can configure ES.

The selection of those practices to include in a catalog is at the crux of what we see as problematic about best practice software claims. In our fieldwork, this has been played out as a tension between the view that superior

processes are embedded in the software versus the view that any given business has learned over time how best to handle its unique situation. Companies often resent having to change their processes to suit the software, processes that they have strived to hone over the years. As one interviewee said to us, when talking about how their software vendor was pressing them to change their business processes rather than accept that they knew best how to operate in their particular market niche: “We had spent 15 years developing something that is successful to become the number one dealer in this industry — we must have done something right. Give us credit for doing something right.”

Best practices are superior business processes from both within and outside an organization that are applied in order to create routine uses of knowledge.

While this interviewee was critical of the idea that the software represented best practice, we have found that this is not always the case, especially among the senior decision makers who are often removed from day-to-day practice. Assuming that superior processes are embedded in software relates to what we have observed in our field research when decision makers assume that adopting some kind of information technology will enable them to change the business; thus, the oft-touted view that adopting an ES will “transform” the organization. There is an assumption that the presence of so-called best practice software templates will enable an organization to achieve superior business processes as a result of implementing software. In other words, the crucial step is in the acquisition of the product; after that, it is just a matter of doing the legwork.

The problem with this mindset is the lack of criticality that would normally precede a decision of this magnitude. Anyone who has ever trained for an athletic event or been a member of a sports team knows that there are a variety of coaching styles, multiple training plans considered optimal, and recommendations for rest, cross-training, hydration, and nutrition. Yet one becomes an athlete when one is able to critically discern for him or herself what works and what is sub-optimal. To just “do the legwork” suggested by an outside expert might in fact be detrimental. Case in point, a five-time marathoner was never able to run under 3:40:00 despite reading multiple books on training approaches. Against all the wisdom culminated from “real” runners, the marathoner began to take walk

breaks every three miles during training and abided by this on race day. Real runners don't walk; yet he finished in a shorter amount of time than ever before. The difference here was not in the execution — hard work was consistent across all his marathon trainings — rather, it was his point of departure. He redefined what was the best practice in terms of marathon training, and he learned that by letting his body recover every three miles with a short walk break, he had more energy in reserve when he got to the 18-mile mark. Where previously he was running on empty and his time slowed dramatically, through his run/break training, he was able to ramp it up for the last eight miles to finish faster.

Developing an ES that works for you will entail experimenting with and continuing to adapt both the technology and the organization in the post-implementation environment.

The point of this example is that organizations, too, need to consciously develop an ES that works for them and, while the configuration options provide for some flexibility, this may not be sufficient for their specific business needs. This is because the menu of configuration options is limited so that, for example, the run/break training option, or its equivalent in organizational process terms, may not be part of the menu. Developing an ES that works for you will entail experimenting with and continuing to adapt both the technology and the organization in the post-implementation environment. It is very unlikely that the “best practice configuration,” even one tailored for your specific industry, will be the ideal fit for your particular organization.

In this way, functional deployment can gradually be built up. The message to companies, then, is not to regard the implementation as completed at the point at which deployment is achieved and there is an information system in production. As we have seen, most companies at this point will be using only some of the potential functionality of the ES. You would do well to not simply stop at this point and bemoan the fact that all the functionality has not been exploited. Rather, you need to see the ES implementation as an ongoing process of continuous learning and adaptation. This means that there will need to be an allocation of

resources for these post-implementation developments. In his article, Bill emphasizes the importance of being realistic about the budgetary implications of an ES implementation. We agree with this point and would add the importance of recognizing the need to allocate resources for these post-implementation developments.

REALIZING THE BENEFITS OF ES: THE TENSION BETWEEN EFFICIENCY AND INNOVATION

Realizing the benefits from their package implementations has been quite difficult or extremely difficult for 40% of the companies surveyed (see Graph 18). As shown in Graph 6, the benefit most realized relates to operational efficiencies. This is not surprising in the context of ERP products, which provide an integrated infrastructure for the organizational support systems. However, with products sold to help improve value-added activities, such as customer relationship management (CRM), we would expect to see evidence of improved competitive positioning within the marketplace as a cited benefit.

One reason that efficiency is more strongly enhanced than competitive positioning can be linked to the fact that there is tension between efficiency and flexibility/innovation. This is because efficiency is obtained by standardizing processes, with individuals then undertaking specialist tasks within those processes that are prescribed by rules and procedures. Some organizations, such as McDonald's, have grown to be very successful based on a rule-bound orientation to promote efficiency. Such bureaucratic work design, however, is not flexible nor is it conducive to innovation, as we can all attest to when we have tried to get a call center operator to respond to some unique circumstance that does not comply with the norm. Thus, innovation requires more organizational slack and a less rule-bound environment, where people can experiment with new ways of working or with new product designs. Today, there is some discussion that an organization can be ambidextrous; for example, with some parts of the organization being designed to support efficiency (e.g., the R&D function), while other parts are designed to support innovation (e.g., the production function). However, with a standardized ES penetrating all parts of an organization, a potential tension between efficiency and flexibility is one that organizations need to consider; more specifically, organizations need to consider how they are going to be able to respond quickly to the dynamic environment when their complex ES is very difficult and expensive to modify.

Importantly, a contradiction arises when analyzing the two survey questions related to ES benefits and the usefulness of the best practice software model (see Graphs 6 and 7, respectively). One might expect to find high correlation between the usefulness of the best practice software and organizational benefits. However, this was not the case; with the exception of operational efficiencies, the majority of respondents were able to realize limited organizational benefits (ranging from not at all to neutral). When juxtaposed against the perceived usefulness of the best practice model, most respondents found the model to be useful (ranging from very to neutral). Along with Bill, we interpret this to mean that while the best practice models embedded in software products are considered useful, the software still isn't benefiting the organization as much as it should. As Bill notes in his conclusions, there is a serious gap between the promise of best practice software and what it ultimately delivers in practice. This worrying finding is further substantiated with the 40% of respondents who claim that it was quite or extremely difficult to realize the benefits that were anticipated when the choice was made to adopt the packaged software, and another 42% were neutral about the level of difficulty (see Graph 18). This begs an answer to the question — what is the point of so-called best practice enterprise software? If the models are helpful, then shouldn't they also help the organizations to achieve expected benefits? This reflects some of the problems previously discussed with the very concept of best practice.

THE CHALLENGES OF ORGANIZATIONAL AND TECHNICAL CHANGE: THREAT OR OPPORTUNITY

The majority of organizations had modified their organizational and business processes to fit the application package software (see Graph 8), and there is a correlation between those modifications and the meeting of organizational goals (see Graph 9). This stresses the importance of being willing to modify process flows and existing roles and responsibilities. Nearly 50% of respondents rate the challenge of making these changes as either quite or extremely difficult and time-consuming (see Graph 11). Our experience suggests that achieving the organizational change necessary to support the processes embedded in the software is especially difficult when there is limited representation on the ES implementation project team from different office locations. For example, in one organization we observed, the project team was staffed with members from the central office, with very little departmental representation. This meant that the ES was configured to suit the needs of central administrators. When rollout began, the departmental administrators protested

that the new system made their jobs more, not less, difficult and thus they resisted the suggested organizational changes.

Technical challenges were also prevalent (see Graph 12). The most effort was expended on the migration and integration of existing data, demonstrating the challenges of cleaning up existing databases. Integration with other applications also required significant effort. Our previous research has also identified these technical difficulties. In one large global organization we were observing, for example, migration of data from legacy systems was difficult because, in Spain — where people often have double-barreled surnames — a field in the legacy system had been used for putting in the second part of the name, while in other countries, this field was used for something else. Integrating the data from different countries was therefore very challenging, taking much longer than had been anticipated.

Analysis of the survey results indicates that perceived benefits and goal achievement are tied to the degree of organizational and technical change experienced during a project. The more challenging these changes, the less stakeholders were able to extract benefit from their ES, and they had also been less successful in achieving their goals. In addition, the timing of such changes impacted the perception of benefits achieved. Unanticipated post-implementation changes that had not been budgeted for made realizing benefits from their ES deployment a challenge. In addition, 57% of the organizations made changes both before and after implementation (see Graph 10). This, we argue, indicates the need to change the lifecycle/rhetoric of successful software implementation. We think it is important for practitioners to hear from researchers and consultants that post-implementation modifications are a normal and expected part of the process rather than to conceptualize such activities as indicative of failure. Moreover, these results stress the importance of understanding the reasons why these organizational and technical challenges are so prevalent and difficult to manage in ES projects, an issue we address next.

Technical and organizational challenges occur because of the ways in which users improvise with any technology — for example, fields get used for things that were not originally intended (as with the double-barreled surnames), and users adopt workarounds in order to maintain legacy practices even when the new ES no longer supports these. This means that it is extremely difficult to maintain an integrated database when it is used across so many different user communities, each with its own local practices, which continue to evolve long after the ES has been first implemented.

Such user improvisation can be perceived as either a problem or an opportunity — a problem because it leads to resistance but an opportunity when it helps bring about unanticipated benefits. Respondents indicate that users resisted removal of their legacy systems and thus continued to use shadow systems to accomplish their work (see Graph 19); they also attempted to add to the package software. Yet seeing these user improvisations as only a problem undermines the potential to transform a system from a copy of what other organizations are doing to a unique system that can actually provide some competitive advantage. The number one reason for project failure is user rejection of the system. The notion that a for-profit company can force acquiescence to a particular system is dangerously false. People are at times recalcitrant and can always find a way to subvert formal policies/processes.

Seeing these user improvisations as only a problem undermines the potential to transform a system from a copy of what other organizations are doing to a unique system that can actually provide some competitive advantage.

CUSTOMIZATIONS: THE MIDDLE WAY

Interestingly, a high proportion of respondents had made either a great deal (24%) or some (54%) customizations to the vanilla ES (see Graph 14). The reason for such customizations links to the previously discussed issue of valuing local practices. If we accept that much organizational knowledge resides in the day-to-day practices of those undertaking the work, then we should be attempting to develop systems that blend the automating and integrating power of an enterprise platform with grounded knowledge. This depends, however, on users understanding enough about the ES to consider how to exploit its functionality. Unfortunately, given the complexity of an ES and the typical “press button” approaches to training, this interaction between user knowledge and ES functionality is often not easily obtained, especially in the short term. Alternative and longer-term approaches to training therefore may be usefully considered. We consider specific alternatives in the final section of this article.

Organizations are both customizing the software and changing organizational business processes and structures. This would indicate that it is false to assume customizations occur only when companies are being

“stubborn” and not wanting to change their organization to fit the software. Rather, both are happening, reflecting the tension discussed above between best practices embedded in the software and best practices that companies have developed based on their unique experience in the market. Moreover, there is no relationship in the survey data between customization and benefits realized and goals achieved, belying the idea that customization is inherently bad. In fact, new consultancy services are being offered that help clients decide when to tailor ES products. The market sought such expertise, and the consultancies responded with project methodologies, which take the customization of software seriously.

IMPLEMENTATION METHODOLOGIES: PLANNING VS. FLEXIBILITY

Organizational and technical changes were made both before and after implementation, and these changes exceeded project plan estimates (see Graph 17). This illustrates that it is not possible to preplan all activities. This is often not understood in project methodologies that assume it is possible to provide static guides that will direct implementation efforts. Rather, the survey results demonstrate that project teams need to be flexible and continuously negotiate through the project, changing plans and deadlines as certain project tasks take longer than anticipated, while other tasks may actually take less time. This must be expected since implementing an ES is inevitably going to involve a political process of negotiation. While it may be uncomfortable to think in terms of flexibility rather than control, it is a worthwhile shift. Hiring a person to analyze the project plans and work with a flexible time-phased approach to budgeting will be beneficial. Furthermore, we argue that it is human nature for ES users to become engaged in a system when it directly influences their sphere of work — in other words, at go-live. While we might involve users in the requirements definition and configuration phases of ES, we should expect that they are only partially tuned into what is being suggested. After all, don't you pay more attention to car advertisements and office parking lots when you are on the market for a new automobile? It should not be surprising to find users who seemed to be on board throughout the project later complaining about the system they helped configure when it comes time for them to use it. We cannot force genuine involvement at certain intervals in order to coordinate our project plans. Rather, if we believe that engagement naturally occurs when an issue becomes salient, then isn't it time to change the

project implementation story to include and *encourage* post-implementation modifications?

The scope of ES change involves many people in an organization and in a way that necessitates much closer collaboration than will often have existed previously. Railroading the technical and organizational changes through the project oversight structure may provide an illusion that the initiative is on track, but this is likely to be at the expense of genuine stakeholder acceptance, thereby leading to a much higher probability of negatively motivated user resistance and workarounds. In recognizing the importance of stakeholder negotiation, it is critical to recognize that consensus is unlikely to always be possible, given the different points of view present in an organization. Instead, it is likely to be important to encourage different groups to “give a little” so that each compromises on some aspect of the ES design. In this approach, the ultimate project goal is not to be “on time” or “perfect,” but to deploy a system that improves what existed previously and recognize that future developments will exploit the potential of the powerful ES even more.

RECOMMENDATIONS

The message from the survey and from our previous research with companies implementing an ES is that these are complicated initiatives, and organizations need to expect delays and frustrations as they attempt to move a project forward. To do otherwise — to take unilateral control of decision making, for example — may result in an ES that gets deployed but then faces considerable user resistance. Instead, the results suggest that it is important to invest time and energy working with these problems, even if it means that the project timetables are revised. Such revisions should be seen as indications of long-term success, not a failure to meet a goal. While Bill’s article recommends better planning and budgeting up front, we would argue that instead of having strict milestones and deadlines, iteration could usefully be built into the project methodology, thus recognizing the need for flexibility. Companies want to understand the budgetary implications of an ES project, yet the reality is that budgets are typically overspent. Allowing for a more iterative project planning process may actually encourage a more realistic ongoing assessment of the financial costs of the project if negotiations include discussion of the costs involved as well as the project rationale for a suggested change. There is no simple formula for project success, but recognizing the legitimacy of diverse stakeholder views, even when they may differ substantially from one’s

own, is likely to be an important factor contributing to success.

With this proviso in mind, we offer the following recommendations, noting that you will need to customize these suggestions to your own local context:

1. **Persevere.** Move from an acquisition to an implementation mindset that recognizes implementation as a long-term organizational change project. Reward partial deployment and encourage your employees to persevere with the more challenging aspects of the ES. Your project team members might be feeling like they have failed to create the perfect solution. Remind them that this approach is not the goal because the dynamic nature of current business environments means that what is perfect today will not be tomorrow. Rather, teach them that an ES implementation is a long-term initiative that will involve continuous adaptation and learning. If your team perseveres, it will help you to gradually exploit the functionality of this powerful type of software package. Multiple times we have been told that an ES initiative that started off well ended up making only insignificant changes in terms of how work was done, and this was attributed to the project losing momentum after go-live when users need coaxing and the software needs tweaking. The project should not end at go-live; instead, when you backfill employees to work full-time on the project, you should include post-implementation time. Companies that try to make crucial post-implementation changes to the system with only a skeleton crew struggle and thus create political storms with end users who feel their concerns are not being addressed. When team members return to preproject positions and the project central command center is shut down, the project loses momentum even though it is a time when a lot more changes need to be made.
2. **Recognize.** ES projects need to be planned iteratively, recognizing that there will be divergent views about what is “best.” Don’t stifle — instead, look for these divergent views and focus on a “good enough” solution that meets the needs of multiple communities of practice across the firm. See users’ improvisations and workarounds as a source of innovation rather than resistance. Improvisation helps exploit ES functionality and can help users to learn to exploit the technology in the post-implementation environment.
3. **Customize.** Tailoring your ES should no longer be frowned upon. Vanilla is good, but adding hot fudge and a cherry is all the better for some. While the costs of customization need to be considered,

recognize that it can be helpful and possibly crucial to include software modifications since the so-called best practices will not always fit your unique organizational circumstances.

4. **Protect.** Whether market-based migrations to new releases of ES software are necessary is irrelevant. The complexity of ES makes it difficult to maintain without vendor support. Therefore, when purchasing enterprise software, it is wise to take a protectionist perspective toward your company. Consider long-term organizational goals and their commensurability with your vendor's business plan. For example, what percentage of its total business does your industry represent? How many resources is it likely to allocate to the refinement of the product you purchased? Is the vendor committed to continued development of an industry-specific product? This represents a strategic alliance between yourself, vendors, consultancies, and implementers. No longer are you the master of your own destiny. We have heard it said that buying an ES is like "getting into bed" with the vendor, but the relationship is not merely a one-night stand because you cannot go your separate ways the morning after go-live!
5. **Innovate.** Consider whether increased efficiency is having any impact on flexibility and innovation in the organization. Consider the idea that another wagon might be more appropriate to hop up on — and if one isn't coming around, then maybe there is a bicycle that will get you where you are going more effectively.

6. **Disseminate.** Outsourcing your ES training is a mistake. Your users are your people, and you must take the time to determine the most effective way of disseminating information to them and fostering a learning environment around the ES. Some creative options include:

- Retaining project staffing and infrastructure well into the first year of ES production
- Using intranet and instant messaging as communication mediums
- Encouraging the development of communities of practice where ideas can be shared among people engaged in similar practices
- Utilizing special training teams that visit different locations and encourage broader discussion about the ES functionality once users have started to use the basics for their job
- Providing help desks with "power users" who can help out with transactions that are done so infrequently that one-time training often does not help because users forget the procedures

In sum, implementing an ES is hard work no matter how you look at it, but multiple organizations have embarked on such projects without dire consequences. Perseverance and resilience of organizational stakeholders is necessary if you are going to make an ES work within your organization. The lesson is that you need to recognize that anything worth doing is worth doing until it doesn't need to be done anymore.



William M. Ulrich, President and Founder, Tactical Strategy Group, Inc.;
Senior Consultant, Cutter Consortium

Application Package Software: The Promise Vs. Reality

INTRODUCTION

Application package software, which is licensed from third-party software providers to fulfill a specific business purpose, has been in use for decades. While the software has matured, implementing and benefiting from third-party application packages remains a challenge. There have been numerous claims made about the value and benefits of using application packages; some are accurate, some are not.

To cut through the marketing hype and rhetoric, Cutter Consortium conducted a survey of application package users that focused on their expectations and experiences. This article provides an analysis of these survey findings along with related conclusions on the successes and challenges in deploying application software packages. It also presents a set of recommendations for organizations needing to streamline future package selection and deployment initiatives.

WHAT IS APPLICATION PACKAGE SOFTWARE?

Application package software, or simply an application package, is a collection of software programs that have been developed for the purpose of being licensed to third-party organizations. Application packages are generally designed to support commonly performed business functions and appeal to multiple types of user organizations. Although a package may be tailored to a user's specific needs through parameters or tables, the software itself is not individualized to a given organization in the same way that custom-designed, custom-coded software would typically be tailored.

Examples of application packages include accounting systems, human resources software, and enterprise resource planning (ERP) software. Application software, within the context of this discussion, does not include all-purpose tools such as Excel, Quicken, or Word. Spreadsheets, databases, and word processing software are all-purpose tools that perform application functions — in the hands of sophisticated users.

APPLICATION PACKAGES: A BRIEF HISTORY

Application packages perform business-specific functions, as opposed to operating system or environmental software such as IBM's zSeries operating system or Windows NT. Most important is the fact that while operating or environmental software upgrades are typically transparent to the business community, application package upgrades are not.

Application packages first became available during the mid-to-late 1960s when financial accounting and payroll software was made available for lease from companies such as McCormack & Dodge. Early application packages focused on accounting or financial solutions. Application packages eventually offered manufacturing, customer management, human resources, and various other functions.

Many of these systems were designed using the same principles as inhouse legacy applications. Early packages focused on a single function within a corporate or government hierarchy, such as accounting, and were built using older software languages and programming techniques. This has caused application packages to become inflexible and increasingly hard to fine-tune to customer requirements.

Application packages must be integrated into inhouse application and data architectures. As a result, inhouse programming teams have had to modify application packages, and this has resulted in difficulties in reintegrating vendor upgrades back into the package software. Falling behind current releases of a given package increases the challenge of upgrading packages exponentially.

Over the past decade, a variety of ERP packages were released that appealed to executives frustrated with inadequate responsiveness from inhouse programming teams. Many times, the decision to acquire and install these packages is driven by senior management, based on promises from vendors stating that the package will provide a low-cost way to rid themselves of legacy

systems while delivering new business functionality. The promise of application packages, however, is contrasted by reality in many situations — as our survey shows.

SURVEY BACKGROUND AND ANALYSIS APPROACH

The survey was commissioned to determine the benefits and challenges associated with deploying and integrating application packages. There were 76 respondents from a wide range of industries, government agencies, and nonprofit centers. The main goal of the survey was to cut through the anecdotal discussions about packaged software and expose the realities of what organizations are actually doing when it comes to deploying and benefiting from application packages.

The approach taken in analyzing the application package survey involved comparing the perceptions and expectations of what organizations thought a given application package could provide versus the actual results experienced by respondents. This focus included:

- Adaptability of the business to align the package with business processes
- Ability of IT to integrate the package with other applications and related data
- Overall success in terms of implementing and benefiting from the package
- Customization requirements versus the perception of how much the package would need to change prior to implementation
- Overall satisfaction with a given package from a best practice perspective

APPLICATION PACKAGE CATEGORIES

The survey sought to determine which types of application packages an organization has licensed and deployed. Application package categories represented in the survey range from enterprise software to more specialized, off-the-shelf offerings (see Graph 1 in the Survey Data section):

- **ERP** software is a cross-department, enterprise-wide package that provides an integrated set of functionality to user organizations. This is the most commonly used software, used by 70% of survey respondents.

- **Human resource management (HRM)** software manages internal payroll, benefits, insurance, and other functions related to administration of inhouse staff. HRM software is in use by 45% of respondents.
- **Customer relationship management (CRM)** software addresses customer management functionality and is primarily used by customer-facing departments. CRM software has grown in popularity and is in use by 40% of the respondents.
- **Supply chain management (SCM)** software supports the complex supply chain management function and is commonly deployed by manufacturing and other organizations reliant on third-party goods and services. SCM software is in use by 16% of the respondents.
- There are a variety of other specialty products on the market. For users of any of these products, the survey included a category called **commercial off-the-shelf software (COTS)**, in use by 28%.

SURVEY PROFILE

The makeup of survey respondents includes a wide range of organizations: manufacturing firms (16%), government agencies (12%), financial institutions (12%), publishing/media firms (10%), consulting companies (12%), transportation/distribution providers (6%), healthcare (3%), and other industries.

In addition, respondents provided input for varying cross-sections of their organizations. For example, 30% of respondents represented their departments, 22% represented a single division, and 46% represented the entire company. IT organization size ranges from very small (fewer than 50 employees) to very large (more than 1,000). However, 37% of respondents have between 50 and 500 IT professionals, while 12% have more than 500 IT professionals. (For more details on survey demographics, including location and revenue, see page 22.)

The respondent profile indicates that application packages are in use by companies and government agencies that range from very small to very large. These organizations cross a variety of industries and regions that have large inhouse support infrastructures as well as smaller support structures. In other words, application packages are not limited to organizations of a particular industry, demographic, or size.

APPLICATION PACKAGE DEPLOYMENT EXPERIENCES

This section discusses the experiences of the organizations that implemented one or more application packages. This analysis is focused on responses from those that either deployed or attempted to deploy a package and the issues they encountered.

The first question used to qualify respondents was to determine the degree to which an organization has deployed an application package (see Graph 2). Twenty-eight percent of respondents have either fully deployed an application package or its core package functionality, while 25% have partially deployed package functionality. Another 4% have deployed packages that were later pulled from production, while 3% have failed in their attempts to implement a package. The remaining 12% have never deployed a package. (Note that these 12% did not enter responses in the remaining survey questions.)

This sampling not only provides the foundation for the remainder of the survey but also suggests an important aspect of package utilization. Only 28% of respondents fully deployed an application package. This is evidence of the fact that organizations do not fully utilize application packages. This means that management expectations should be adjusted to realize that, for whatever reason, a company may only utilize a portion of the capabilities contained within a given package.

Implementation Challenges

The implementation challenges associated with application packages are best exemplified by the degree of modifications required to the package. While the business process and organizational alignment issues are addressed later in this article, the physical customization of a package can be driven by many factors. When asked about the degree of customization required, only 2% of respondents did not change the package at all, while another 10% only modified it slightly (see Graph 14).

On the other hand, nearly one-quarter of respondents modified their packages a great deal, while another 54% modified them somewhat. When asked whether their organizations had planned to make these changes, only 14% of respondents said they had anticipated extensive changes, while another 36% expected moderate changes (see Graph 16). In comparison to the 78% that applied a great deal or moderate changes to their packages, only 50% *expected* to apply this degree of change to their packages.

The survey additionally sought to determine when these changes were applied, as shown in Graph 15. Close to half (44%) of the respondents applied all or most of their application package changes prior to implementation. Another 47% say that they applied changes both before and after implementation. Only 9% applied all or most of their changes after implementation. From a planning perspective, this is an indicator to management that package customization is typically required prior to the deployment of a given package.

Survey findings show that the perception of senior management that packages can be dropped into an organization with little change is typically inaccurate.

When asked if the degree of package customization exceeded expectations, nearly half of respondents (49%) say that the changes exceeded what was anticipated and what was budgeted for the project (see Graph 17). Another 45% believe that the changes were about what was anticipated/budgeted, while only 6% think that the changes were less than anticipated/budgeted. These findings show that the perception of senior management that packages can be dropped into an organization with little change is typically inaccurate.

These findings additionally show that management must be more realistic about the degree of package changes required and must build these requirements into plans and budgets. Further, management should increase budgetary allocations for package implementations in many cases to reflect the reality that almost half of survey respondents spent more than they had anticipated on package deployment.

Integration Challenges

A second challenge related to implementing an application package is the impact to the surrounding environment. The survey found that 31% of respondents expended a great deal of effort integrating application packages with existing inhouse applications, and another 46% spent some effort on existing systems integration (see Graph 12). In addition, over half (53%) spent a great deal or at least some effort on integrating a package with other application packages.

Another 45% of responding organizations expended a great deal or some effort to integrate a package with

middleware applications, while an additional 36% spent a great deal or some effort to integrate the package with business process automation software. Finally, 42% of respondents had a great deal of difficulty in integrating existing data into a package, while another 39% experienced some difficulty. This is an essential step with almost any package, and these findings indicate that data integration challenges are nontrivial.

Even more revealing was the fact that 34% of the surveyed organizations found these migration and integration tasks to be extremely difficult (13%) or quite difficult (21%), as shown in Graph 13. Another 37% found at least some difficulty in integrating an application package into the existing systems environment. Five percent of respondents found that integrating a package was not a difficult task.

These findings expose one package implementation challenge that can be extremely difficult but that management may not have anticipated. Integrating an application package into complex computing environments layered with legacy systems and data stores, other packages, middleware, and business process automation tools is not a trivial exercise. These findings further suggest that management may not have fully mapped the capabilities of the packages that have been required with the existing IT environments into which these packages must be integrated.

Management may not have fully mapped the capabilities of the packages that have been required with the existing IT environments into which these packages must be integrated.

USER BENEFITS: EXPECTATIONS VS. REALITY

While implementation challenges are important from a planning and budgeting perspective, the functionality and usability of an application package from a business perspective is even more critical. This section discusses the perceived business benefits anticipated by organizations and compares and contrasts these perceived benefits with the results achieved by the organization as the application package was deployed.

Projected Business Benefits of Package Deployment

Based on the survey findings, the main objectives of buying an application package are to improve

organizational performance and to ensure consistent quality across the business. These findings are based on the percentage of respondents citing the following survey items as important or very important (see Graph 4). Note that the percentage of respondents for each category is shown in parentheses:

- Desire to improve organizational performance (90%)
- Quality assurance across the business (85%)
- Benchmarking against functional best practices (50%)
- Security and integrity of the data resource (72%)
- Benchmarking against industry best practices (37%)
- Fear that competition is leaving you behind (29%)

These findings indicate that organizations are seeking productivity and quality improvements across lines of business with an eye toward capitalizing on industry best practices and improving overall security and systems integrity. This is reinforced by the fact that a related finding indicates that the adoption of best practices motivates 70% of the respondents to seek package-based solutions (see Graph 3).

Respondents were asked to share what they consider to be best practices based on several survey selections (see Graph 5). Responses suggest that superior business processes that will be implemented through using the software is a key factor (60%), along with having software templates to avoid “reinventing the wheel” by building custom applications (60%). Other factors include bringing external practices into the organization by using the application package (39%), regulatory factors (33%), and risk management (31%).

Actual Benefits of Package Deployment

As shown in Graph 6, a great deal or quite a lot of the actual benefits gained from package deployment include operational efficiencies (51%), customer satisfaction (34%), strategic advantages (33%), revenue collection/generation (32%), cost reduction (27%), and resource reallocation (24%). These findings show that gains in operational efficiencies and customer satisfaction are very real for organizations that deploy application packages. Given these findings, management should strongly consider these benefits as key justifiers when building the business case for an application package.

When asked of the usefulness of a best practices model as embedded in the application package, respondents indicate that the following are either very useful or quite useful (see Graph 7):

- Improving internal operations (55%)
- Facilitating business process automation or integration (53%)
- Facilitating cross-organizational collaboration (46%)
- Improving customer resource management (38%)
- Improving strategic data management and use (36%)
- Improving knowledge production and management (32%)
- Improving supplier/distribution chain management (30%)

These findings further confirm that improving internal operational efficiencies is a benefit of deploying application packages along with business process automation and organizational collaboration across business lines. These second two items are related so there is no surprise that they ranked close to each other.

These findings additionally support the notion that packages are useful for managing customer resources and strategic data management. Ranking slightly lower, however, are the best practice benefits related to knowledge management and supplier/distribution chain management. Clearly, the internal operational gains are a bigger advantage over external customer- and distribution-related gains.

Application Package/Business Process and Organizational Alignment

The flip side of the business-benefits equation has to do with the adjustments that an organization must make to its entrenched business processes and organizational infrastructures.

When asked about the extent to which respondents modified organizational and business processes to fit the package, three-quarters modified their business processes somewhat (53%) or a great deal (22%), as shown in Graph 8. Only 2% of respondents did not modify business processes to adapt to the package.

A followup question sought to clarify the types of modifications that organizations enacted as a result of implementing an application package (see Graph 9). Respondents enacted process flow changes to complete business activities a great deal (30%) or somewhat (35%). Half of the respondents enacted changes to organizational roles and responsibilities, while another 36% enacted a great deal or moderate degree of changes to organizational reporting relationships. Departmental or functional changes were applied by 35% of the responding organizations.

These findings support the notion that if management plans to deploy an application package, then organizations must plan on adapting their business processes and organization infrastructures to those packages.

Organizations applied these changes both before and after the implementation of the application package: 24% of respondents made organizational changes prior to implementation, 19% made changes after implementation, and 57% made changes both before and after implementation (see Graph 10). This indicates that organizational adjustments resulting from the application package implementation project are ongoing activities that management must plan for and manage carefully.

More importantly — as indicated in Graph 11 — is the degree of difficulty posed by these organizational changes. If, for example, there are simply organizational adjustments needed to conform to an application package, management would be able to budget for and deliver these changes without significant business disruption. Difficult changes, however, are likely to cause more business disruptions.

A significant percentage of survey respondents claim that these changes were either quite difficult (30%) or extremely difficult and time-consuming (17%). Only 6% of respondents that applied organizational changes report that organizational changes were not difficult. This is significant when one considers that 98% of respondents applied organizational changes as a result of implementing an application package. In other words, virtually every respondent had to apply organizational changes, and almost all of those respondents consider these changes to have been moderately difficult, quite difficult, or extremely difficult and time-consuming.

User-Driven Impediments to Package Deployment

One key indicator as to the usability of an application package and its ultimate success is the receptiveness of the business community and the impediments encountered within the user environment. The survey uncovered several interesting points in this regard.

Almost half of the respondents (47%) say that users attempted to introduce “add-ons” to supplement the functionality of the package (see Graph 19). This indicates that almost half of those organizations implementing a package had users that did not feel the package was good enough in its off-the-shelf form.

An even more telling finding is that more than 60% of respondents report that users resisted the removal of existing application systems that replicated the functionality found within the package. In other words,

users favored their existing applications over the functionality provided by the application package. This is a critical indicator that management may not have investigated the applicability of the package to business requirements or not understood the business requirements in the first place.

One additional challenge is the issue of user-based “shadow” systems. Over half of all respondents (51%) had problems with user-base applications written in Excel, Access, 4GL, or other custom software. These systems are rarely discussed, yet are significantly problematic given that almost all business units perform certain key functions beyond the reach of IT. These systems are an impediment to package deployment and should be identified during the initial planning and assessment for package deployment.

Time and budget allocations for package customization efforts are exceeding the project parameters anticipated at the onset of the project.

Realizing Application Package Benefits: At What Cost?

The bottom line for determining user benefits involves the difficulty in realizing benefits from an application package and the extent to which best practices embedded within an application package have actually helped an organization meet its goals.

Regarding the difficulty in realizing benefits, respondents report that benefits from the application package have been extremely difficult to achieve (10%) or quite difficult to achieve (30%), as shown in Graph 18. Only 18% of respondents report that it has been easy to realize benefits from their application package, while 42% report that it has been neither easy nor difficult to realize benefits from their application package.

The fact that 40% of respondents report that realizing package benefits has been quite or extremely difficult and only 18% report that it has been easy to realize benefits corresponds to earlier findings related to the degree of organizational change and package customization required.

Finally, respondents were asked to what extent best practices as embedded within the application package have helped the organization meet its goals (see Graph 20). Only 12% claim that the package helped a great deal, while 48% report that the package helped

somewhat. In contrast, 18% indicate that the package did not help much or at all, while another 22% are neutral on the subject of how much the package helped meet organizational goals.

Clearly these results did not live up to expectations given the survey findings related to the degree of application package customization and the degree and difficulty of organizational changes needed to adapt business processes and infrastructure to accommodate the application package.

CONCLUSIONS AND RECOMMENDATIONS

The lessons learned from the application package survey are significant and point to a series of recommendations that organizations can pursue to achieve greater benefits from their application package investments.

Summary

A summary of conclusions from this report is presented below:

1. The majority of responding organizations are users of ERP systems but additionally use a combination of other application packages. These include CRM, HRM, and various other off-the-shelf software packages.
2. Application packages are, according to this survey, only fully implemented 28% of the time. While this is not necessarily a sign of success or failure, executives should note that a package is unlikely to deliver or be deployed to the full breadth of capability that a given package provider may be suggesting.
3. Package customization requirements were extensive and exceeded the expectations of survey participants. This finding suggests that the time and budget allocations for package customization efforts are extending beyond the project parameters anticipated at the onset of the project.
4. Integrating application packages into complex computing environments is difficult and time-consuming. These findings could be the result of not fully mapping package capabilities to existing IT environments, including business data, existing systems, middleware technology, and business process automation solutions.
5. One telling finding is that users continue to favor their existing applications over the functionality provided by the application package. Users also like to hold on to their shadow systems that have been user-developed to surround core applications.

This is likely an indicator that management may not have investigated the applicability of the package to the business requirements, or it may not have understood the business requirements in the first place that the package was to address.

6. Organizations are seeking productivity and quality improvements across lines of business to leverage industry best practices. In an attempt to meet these requirements using a package strategy, organizations had to modify their business processes in favor of the business processes embedded within the application package. In addition, respondents had difficulty implementing changes to their business processes. This finding not only signals a potential for budget overruns but also points to why only portions of certain packages are implemented and why users prefer their legacy applications over the package application.
7. Virtually every respondent had to apply organizational changes, and almost all of those respondents considered these organizational changes to be somewhat difficult, quite difficult, or extremely difficult and time-consuming. In other words, companies are being forced to change how they operate to align themselves with these application packages. This may be fine with some companies, but other organizations may have not understood that a software package was dictating how they were going to operate.
8. Realizing the benefits of these packages was a challenge for many respondents. In fact, 40% of respondents report that realizing benefits has been quite difficult or extremely difficult, and only 18% report that it has been easy to realize benefits. This corresponds to earlier findings related to the degree of organizational change and package customization required.
9. Organizations are not meeting the objectives envisioned when they initially decided to obtain an application package and are not gaining bottom-line value commensurate with the challenges associated with the implementation effort. This is the most critical finding because it suggests that there is a serious gap between what organizations think an application package can deliver and what it ultimately does deliver.
10. There are real benefits to be derived from application packages as long as organizations can manage expectations and cost-effectively deliver this value to the business community.

These findings expose the great conundrum of the application package strategy. Executives want to take advantage of best practices through the acquisition of application packages, but implementing these best practices across an enterprise tends to be more difficult, more time-consuming, and more costly than first anticipated. In addition, the business users, supposedly the beneficiaries of these best practices, are either demanding changes to these packages or resisting these packages in favor of legacy systems and user-based systems.

The bottom line is this: Organizations are making major investments to customize application packages, integrate those packages into surrounding IT environments, and adapt business processes and organizational infrastructures to adapt to these packages. Yet the benefits accrued from these investments are delivering less than what the business community had anticipated in a variety of best practice categories.

Recommendations

What can be done about this? Here are four recommendations that could address the obvious challenges of application package selection and deployment initiatives:

1. Spend more time on up-front analysis, mapping business process, organizational, data, and functional requirements for your organization to the capabilities of the package. This analysis should include a review of business processes, a current systems assessment, analysis of existing data requirements, examination of user-based applications, and a mapping of the findings to available package options.
2. Organizations must do a better job uncovering business requirements prior to package acquisition. This involves a needs analysis with the business users who will be using the application package. Too many times the business community is not involved in selecting a package or ensuring that the package maps to business requirements.
3. Management should review increasing budgetary allocations for application package selection and implementation projects. This is based on the fact that close to half of the survey respondents spent more than they had anticipated on package deployment.
4. Executives must adjust expectations of what packages can deliver and the level of investment and organizational retooling necessary to achieve these objectives.



From the Editor, Gabriele Piccoli

CONCLUSION

It's a New World ... So Grab Your Old Weapons

This issue of *Cutter Benchmark Review* focuses on a topic of interest to all organizations, bar none: the use of prepackaged software and the management of the implementation process of off-the-shelf applications. Our academic contribution this month is provided by Erica Wagner, Assistant Professor at the School of Hotel Administration at Cornell University, and Sue Newell, Cammarata Professor of Management, Bentley College. Providing our view from the field is Bill Ulrich, Cutter Consortium Senior Consultant and President of Tactical Strategy Group, Inc.

One of the survey results I find most interesting is that only 28% of respondents fully deployed an application package. This is, of course, by design. Like how the user who complains that he only employs a fraction of Excel functionalities forgets that Excel is designed not only for him but for millions of other users (some like him and some not), so, too, management often forgets that its organization is unique — as is every other one implementing the same off-the-shelf application. Expecting to fully deploy the software just because it is available — not because every component fits well with the organization's processes — is simply planning for heartbreak. Heartbreak, ranging from mild to deadly, comes in the form of user dissatisfaction, countless post-implementations requests, overtime and overbudget implementations, and so on. I do understand that sometimes you need to give a little to get a little, as when the firm values the integration offered by an ES and is willing to forego deployment of the best module for each of its functional areas. But don't expect management to understand this "seemingly obvious" fact on its own.

Moreover, with nearly half of our respondents reporting that changes went beyond what was anticipated, Bill stresses that we need to be proactive in managing management expectations. It is our responsibility to ensure that management understands what to expect during and after implementation. I can certainly tell you that many in the intended audience of my new textbook (i.e.,

future senior managers) typically do not realize how dramatically large organizational off-the-shelf applications (and the issues they bring about) differ from the personal productivity software they are most familiar with when it comes to IT.

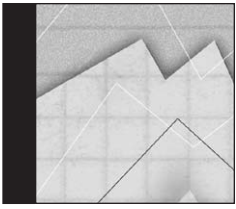
To compound the problem, management often severely underestimates how difficult and costly customization is going to be. Yet, as Erica and Sue show, while many respondents faced unplanned changes, nearly 95% managed to create a system that was in production. So Erica and Sue have a very provocative suggestion: How about we stop treating the go-live moment as the watershed of failure or success? They recommend including post-implementation time. Erica has done some further writing in this area explaining the reasons for enduring project failure and challenging the traditional system development lifecycle.¹

For a publication that prides itself on multiple views, there is remarkable consistency between the analysis and guidelines in our two contributions. This is perhaps best captured by Bill when he says: "The bottom line is this: Organizations are making major investments to customize application packages, integrate those packages into surrounding IT environments, and adapt business processes.... Yet the benefits accrued from these investments are delivering less than what the business community had anticipated in a variety of best practice categories."

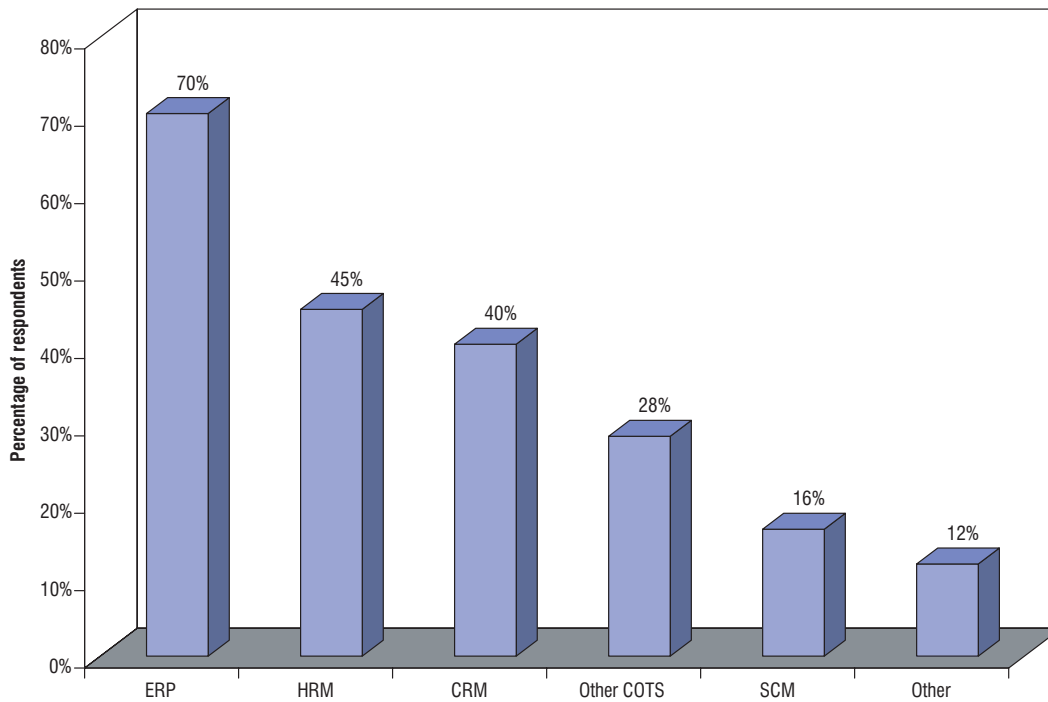
It may be the brave new world of "buy and make," as I called it in the introduction to this issue. Yet the above reminds us very much of the traditional challenges associated with custom systems design and development.

What's an IT professional to do then? Polish those relationship skills and continue the good fight of senior management education. I am sure that I speak for the whole Cutter crew and the CBR team when I say that we'll continue the fight by your side.

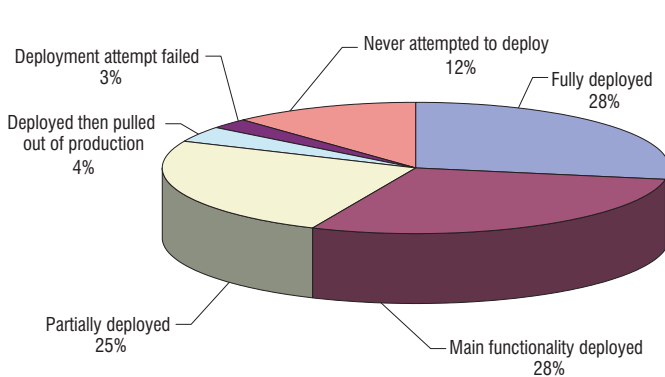
¹The paper (Wagner, Erica, and Gabriele Piccoli, "A Call to Engagement: Moving Beyond User Involvement in Order to Achieve Successful Information Systems Design") is forthcoming in *Communications of the ACM*, but I can share it if you e-mail me at gpiccoli@cutter.com.



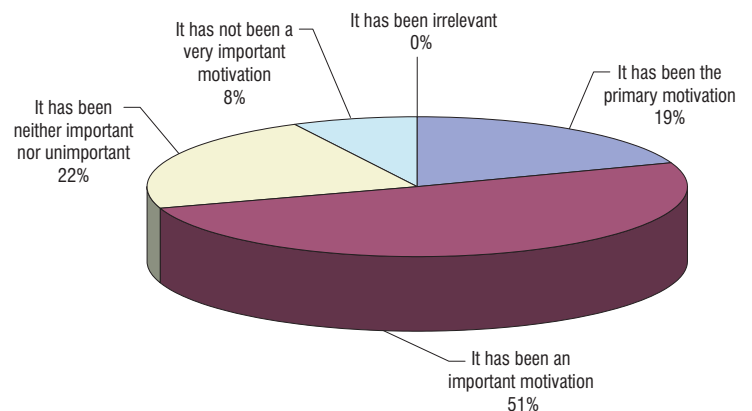
Application Package Software



Graph 1 — Which types of application package software has your organization implemented?
(Please select all that apply.)



Graph 2 — To what degree has your organization deployed an ERP or other type of application package to support a business process?



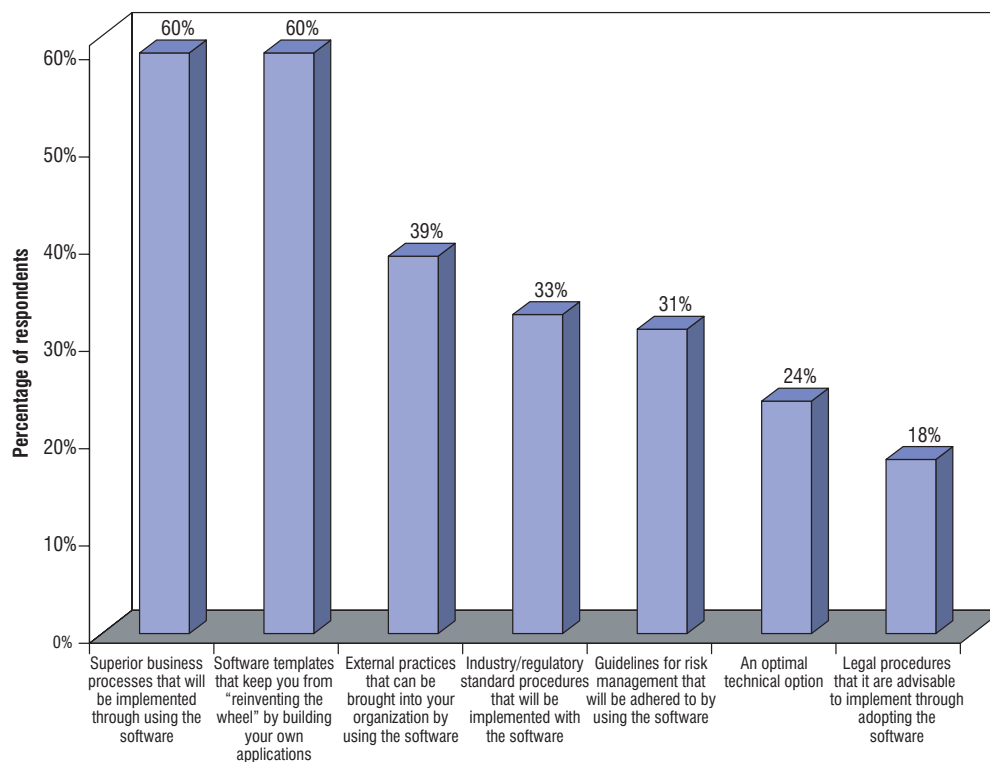
Graph 3 — To what extent have the decision makers in your organization acquired application package software because they believe it will enable the organization to adopt best practices?

SURVEY DEMOGRAPHICS

This survey investigated the extent of organizations' adoption of application packages, the reasons for adoption, the benefits expected and the benefits realized, software changes, and any difficulties associated with adoption. Of the 76 responding organizations, 45% are based in North America, 16% in Europe, 11% in Asia, 9% in Australia/Pacific, 7% each in South America and Africa, and 5% in the Middle East. Twenty-six percent have more than 5,000 employees, 24% have between 1,000 and 5,000 employees, 33% have between 100 and 1,000 employees, and the remaining organizations have 100 or fewer employees. Twenty-four percent have annual revenues of more than US \$1 billion, 20% have annual revenues between \$100 million and \$1 billion, 38% have annual revenues between \$10 million and \$100 million, with the remainder having annual revenues less than \$10 million. Annual IT budgets range from less than \$100,000 (7%) to more than \$10 million (25%), with 38% having annual IT budgets between \$100,000 and \$1 million and 18% between \$1 million and \$10 million (12% do not know the dollar amount of their annual IT budget). Over 50% of the respondents hold senior management/policy making or IS/IT management titles, with project management, consulting, and QA management being among the other titles.

	Not at All Important	Not Very Important	Neutral	Important	Very Important
Desire to improve organizational performance	6%	0%	5%	30%	60%
Fear that competition is leaving you behind	19%	16%	36%	21%	8%
Benchmarking against industry best practices	9%	22%	31%	31%	6%
Benchmarking against functional best practices	8%	15%	27%	37%	13%
Quality assurance across business	9%	3%	3%	51%	34%
Security/integrity of data resource	8%	5%	16%	36%	36%

Graph 4 — How important were these reasons for your organization to implement application package software?



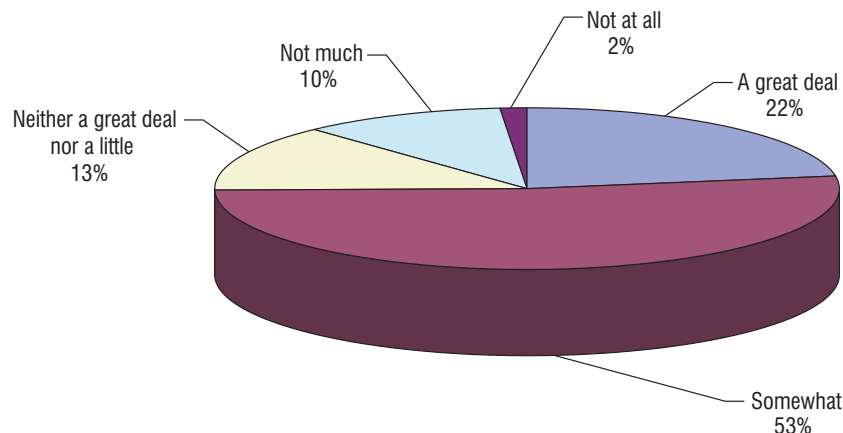
Graph 5 — What is your definition of a software-based "best practice"? (Please select all that apply.)

	Not at All	Not Much	Neutral	Quite a Lot	A Great Deal
Resource reallocation	10%	30%	36%	22%	2%
Business cost reduction/cost deferment	8%	16%	49%	24%	3%
Revenue collection/generation	16%	16%	36%	27%	5%
Operational efficiencies	5%	16%	28%	39%	12%
Customer satisfaction	6%	22%	37%	31%	3%
Strategic advantages	8%	22%	37%	24%	9%

Graph 6 — To what extent has the application package software that you adopted helped the organization achieve the following benefits?

	Very Useful	Quite Useful	Neutral	Not Very Useful	Not at All Useful
Facilitating business process automation/integration	13%	40%	22%	19%	5%
Facilitating cross-organizational collaboration	19%	27%	30%	18%	6%
Improving customer resource management	5%	33%	37%	16%	9%
Improving internal operations	15%	40%	27%	15%	3%
Improving supplier/distribution chain management	8%	22%	42%	19%	9%
Improving knowledge production/management	10%	22%	39%	22%	6%
Improving strategic data management/use	12%	24%	43%	16%	5%

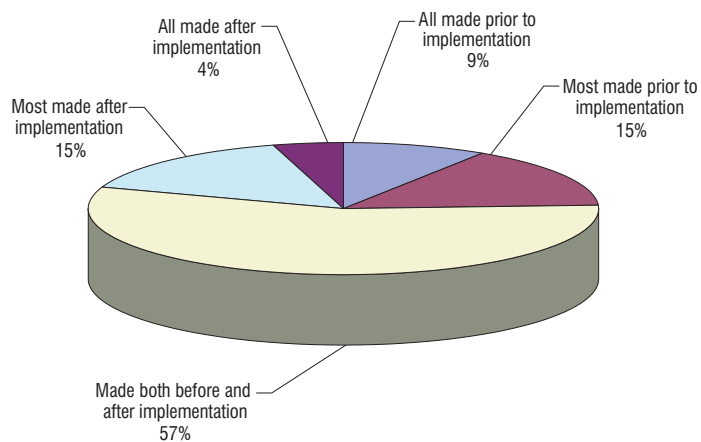
Graph 7 — How useful has a best practice model, as embedded in the application package software, been in relation to the following?



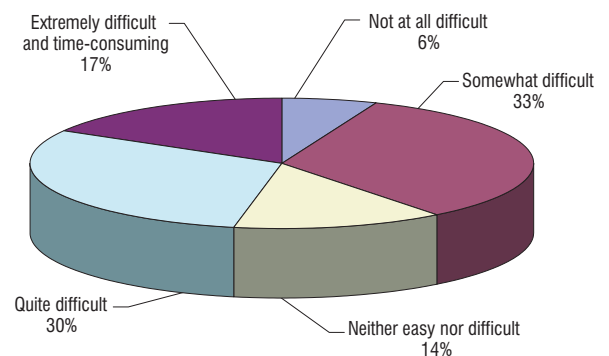
Graph 8 — To what extent did you modify organizational and business processes to fit the application package software?

	A Great Deal	Somewhat	Neutral	Not Much	Not at All
Reporting relationships	9%	27%	17%	33%	14%
Departmental/functional units	11%	24%	15%	38%	12%
Reward/pay mechanisms	5%	11%	30%	41%	14%
Organizational roles and responsibilities	12%	38%	18%	26%	6%
Process flows necessary to complete business activities	30%	35%	21%	14%	0%

Graph 9 — To what extent did you modify the following aspects of your organizational structure and business processes to fit the application package software?



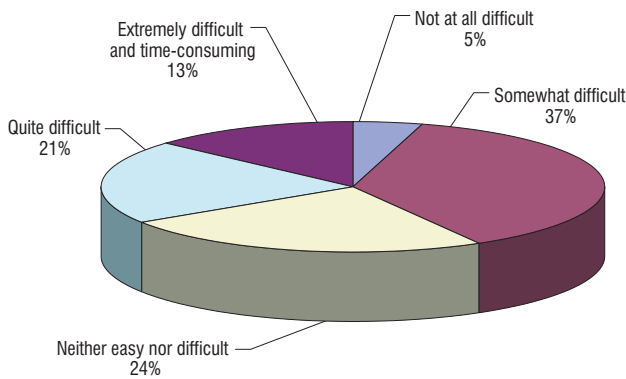
Graph 10 — Did your organization make these organizational changes prior to the implementation, or were the changes made once the system was implemented?



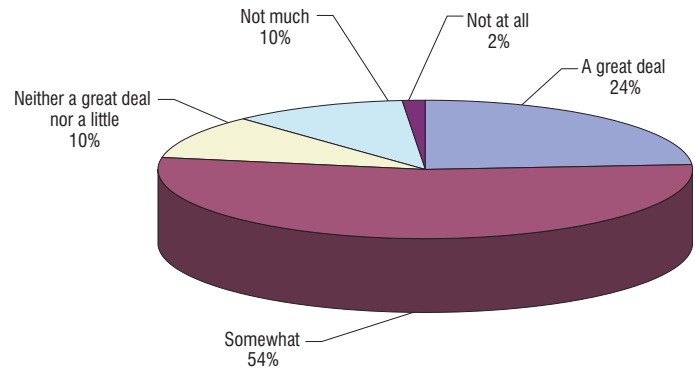
Graph 11 — To what extent were the organizational changes a difficult challenge?

	A Great Deal	Somewhat	Neutral	Not Much	Not at All
Integration with existing applications	31%	46%	9%	10%	3%
Integration with other package applications	16%	37%	15%	24%	8%
Combined package application with middleware technology	12%	33%	15%	24%	16%
Migration and integration with existing data	42%	39%	12%	5%	3%
Retooling of business process automation software	15%	21%	21%	21%	22%

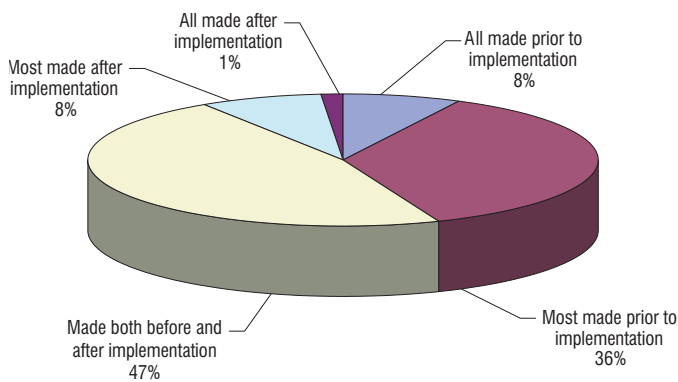
Graph 12 — To what extent did your organization expend effort on the following activities linked to your application package software implementation?



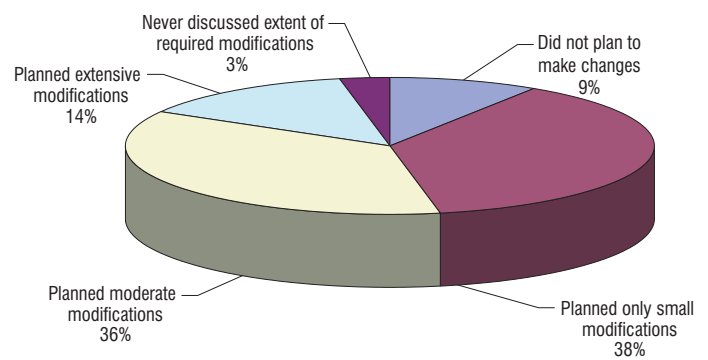
Graph 13 — To what extent was the migration and integration of existing data a difficult challenge?



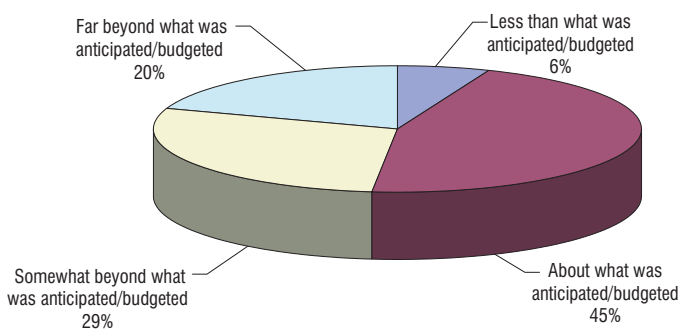
Graph 14 — During the application package software implementation, to what extent did you modify the software through customization (i.e., beyond the standard configuration process) to fit existing organizational processes and structures?



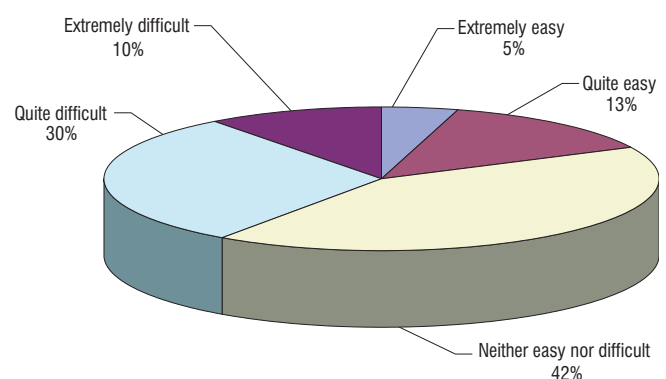
Graph 15 — Did your organization make these software or configuration changes prior to the implementation, or were the changes made only once the system was implemented?



Graph 16 — To what extent did your organization plan to make software or configuration changes prior to the implementation?



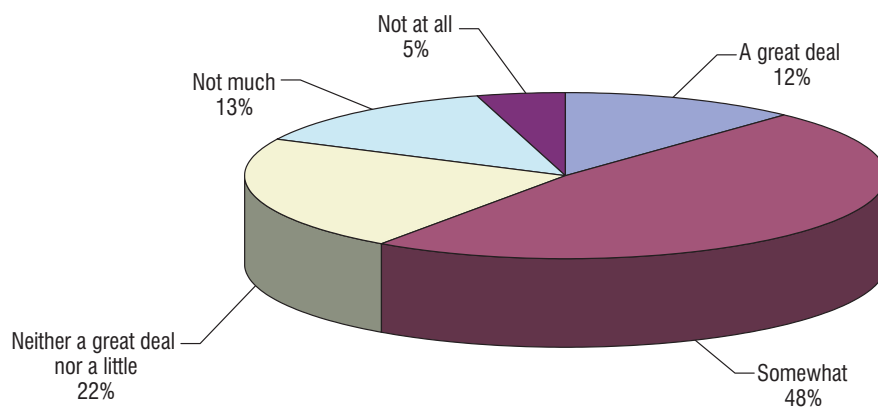
Graph 17 — To what extent did the degree of software changes or configuration changes exceed your expectations?



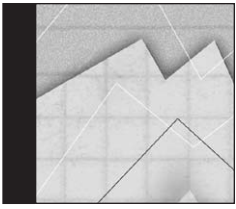
Graph 18 — How difficult has it been to realize the benefits that were anticipated when you adopted the application package software?

	Major Problem	Somewhat of a Problem	No Value Added or Subtracted	Somewhat of an Opportunity	Important Opportunity	N/A
User attempts to introduce “add-ons” to supplement the application	13%	34%	25%	15%	5%	8%
User resistance to the removal of existing systems that were intended to be redundant with the software implemented	21%	40%	24%	8%	6%	2%
User-based shadow systems	21%	30%	25%	10%	8%	6%
Software vendor upgrades that we have been forced to accommodate	3%	34%	33%	9%	6%	15%

Graph 19 — Please indicate the extent to which you have found the following items to pose a problem or an opportunity:



Graph 20 — To what extent have the best practices embedded within the application package software helped the organization meet its goals?



About the Authors

Sue Newell is the Cammarata Professor of Management, Bentley College and Visiting Professor of Management at Royal Holloway, University of London. She has a BSc in psychology and a PhD from Cardiff University. Dr. Newell is currently the PhD Director at Bentley. She has worked previously at Aston, Birmingham, Nottingham Trent, and Warwick Universities, all in the UK. Her research focuses on understanding the relationships between innovation, knowledge, and organizational networking — primarily from an organizational theory perspective. At Warwick, Dr. Newell was a founding member of ikon, and she continues to focus on research that explores innovation processes using knowledge and organizational networking perspectives. She is also involved in research that explores the implementation and use of packaged information systems. Her research emphasizes a critical, practice-based understanding of the social aspects of innovation, change, knowledge management, and interfirm networked relations. Dr. Newell has published over 60 journal articles in the areas of organization studies, management, and information systems as well as numerous books and book chapters.

Gabriele Piccoli is a Senior Consultant with Cutter Consortium's Business-IT Strategies practice and Editor of *Cutter Benchmark Review*. He is also Associate Professor of Information Systems at the School of Hotel Administration at Cornell University. His research and teaching expertise is in strategic information systems and the use of Internet technology to support customer service, organizational relationship, and internal operations such as virtual teaming and Web-based training.

Dr. Piccoli has held positions as Adjunct Professor of Information Systems at the AB Freeman School of Business at Tulane University, and as instructor at the EJ Ourso College of Business at Louisiana State University where he received his MBA, as well as his PhD in business administration with emphasis in management information systems. He also holds a Laurea in Economia e Commercio from Università di Pavia.

Dr. Piccoli is actively involved in the profession as a former manager of the ISWorld Mailing List, a global mailing list of about 3,000 academics and professionals in management information systems, and has served on organizing committees for ICIS 2001 and ICIS 2002. He is also a member of the ACM, AIS, and INFORMS. His research has appeared in *MIS Quarterly*, *Decision Sciences Journal*, *MIS Quarterly Executive*, *Communications of the ACM*, *Harvard Business Review*, *The DATA BASE for Advances in Information Systems*, and *Cornell Hotel and Restaurant Administration Quarterly*, as well as other academic and applied

journals. Dr. Piccoli has also served as ad hoc reviewer for a number of journals in the field and serves on the editorial board of the *Cornell Hotel and Restaurant Administration Quarterly*. He can be reached at gpiccoli@cutter.com.

William M. Ulrich is a Senior Consultant with Cutter Consortium's Business-IT Strategies and Enterprise Architecture practices. He served as Guest Editor of the *Cutter IT Journal* and has been a speaker at Cutter *Summits* and symposia. Mr. Ulrich is President and founder of Tactical Strategy Group, Inc., a management consulting firm specializing in information and business planning strategies. He has more than 25 years' experience in the information management field. His system transformation strategies have been the basis for migration projects worldwide. As author of TSRM, a widely deployed redevelopment methodology, Mr. Ulrich is viewed as the leading authority on systems transformation for the IT industry. He has advised *Fortune* 1000 companies, government agencies, high-tech companies, and consulting firms on information management, legacy transformation, IT organizational change, business continuity planning, supply chain management, market deployment, partner alliance, and Y2K strategies.

Mr. Ulrich previously served as a reengineering products director at a software company where he had international responsibility for the company's reengineering product solutions. Prior to that, he served in a senior management capacity at KPMG. Mr. Ulrich has lectured internationally to thousands of business and IT professionals and has consulted and testified as an expert witness on the use of intellectual property within the computer field. Mr. Ulrich has authored hundreds of articles appearing in major publications on IT and management trends. His latest book is entitled *Legacy Systems: Transformation Strategies*. He can be reached at wulrich@cutter.com.

Erica Wagner is an Assistant Professor of Information Systems at Cornell University's School of Hotel Administration. She earned her PhD from the London School of Economics and has significant practitioner experience in the fields of accounting and budgeting. Dr. Wagner's research interests focus on the ways in which complex software is "made to work" within different organizational contexts. Her research has appeared in numerous academic and applied journals including *Information and Organization*, *Communications of the ACM*, *Journal of Strategic Information Systems*, *Journal of Applied Behavioral Science*, and the *Cornell Hotel and Restaurant Administration Quarterly*. For more information on Dr. Wagner, visit www.people.cornell.edu/pages/elw32/index.htm.

About Cutter Consortium

Cutter Consortium is a unique IT advisory firm, comprising a group of more than 150 internationally recognized experts who have come together to offer content, consulting, and training to our clients. These experts are committed to delivering top-level, critical, and objective advice. They have done, and are doing, groundbreaking work in organizations worldwide, helping companies deal with issues in the core areas of software development and agile project management, enterprise architecture, business technology trends and strategies, enterprise risk management, business intelligence, metrics, and sourcing.

Cutter delivers what no other IT research firm can: We give you Access to the Experts. You get practitioners' points of view, derived from hands-on experience with the same critical issues you are facing, not the perspective of a desk-bound analyst who can only make predictions and observations on what's happening in the marketplace. With Cutter Consortium, you get the best practices and lessons learned from the world's leading experts, experts who are implementing these techniques at companies like yours right now.

Cutter's clients are able to tap into its expertise in a variety of formats including print and online advisory services and journals, mentoring, workshops, training, and consulting. And by customizing our information products and training/consulting services, you get the solutions you need, while staying within your budget.

Cutter Consortium's philosophy is that there is no single right solution for all enterprises, or all departments within one enterprise, or even all projects within a department. Cutter believes that the complexity of the business technology issues confronting corporations today demands multiple detailed perspectives from which a company can view its opportunities and risks in order to make the right strategic and tactical decisions. The simplistic pronouncements other analyst firms make do not take into account the unique situation of each organization. This is another reason to present the several sides to each issue: to enable clients to determine the course of action that best fits their unique situation.

For more information, contact Cutter Consortium at +1 781 648 8700 or sales@cutter.com.

The Cutter Business Technology Council

The Cutter Business Technology Council was established by Cutter Consortium to help spot emerging trends in IT, digital technology, and the marketplace. Its members are IT specialists whose ideas have become important building blocks of today's wide-band, digitally connected, global economy. This brain trust includes:

- Rob Austin
- Christine Davis
- Tom DeMarco
- Lynne Ellyn
- Jim Highsmith
- Tim Lister
- Lou Mazzucchelli
- Ken Orr
- Ed Yourdon