



## The next generation of in-house software development

How leading-edge companies are streamlining applications development.

Sam Marwaha, Samir Patil, and Ranjit Tinaikar

*Web exclusive, February 2006*

Most companies recognize the many advantages of buying software applications off the shelf rather than developing them in-house. By purchasing packaged applications (such as those from Microsoft, Oracle, SAP, and other vendors) or by using applications that vendors create, host, and make available over the Web (for example, those from Salesforce.com), companies can acquire effective business solutions with the benefits of standardization. In this way, they can keep up with of the innovations created by focused specialists.

In spite of those advantages, custom-built applications are still very much a part of the IT landscape. Companies in many sectors spend well over half their applications budgets on custom software, used largely to enhance, support, and operate customized systems. For large companies in competitive, fast-moving industries such as telecommunications, financial services, high tech, pharmaceuticals, and media, those outlays can run into hundreds of millions of dollars. Banks, for instance, frequently build custom applications to support new financial products or to manage risk. Pharmaceutical companies regularly build custom applications to support R&D and marketing activities.

Even when a company uses off-the-shelf applications, it frequently customizes them, typically by adding modules that provide a distinctive capability. A high-tech company, for example, installed a suite of packaged enterprise-resource-planning applications but also built a customized cash-management application to supplement the ERP product's plain-vanilla finance functionality. Unfortunately, even a little customization here and there quickly adds up. Worse, when companies must revise systems to meet new business needs, changing interconnected, customized applications can be difficult and costly, and upgrades to customized applications usually cost more than those to packaged software.

Nevertheless, some pioneering companies have found a way to capture the benefits of packaged software in a customized-applications environment. They have adopted the approach of software vendors, which package and sell applications aimed at the common needs of many customers rather than of individuals, by writing an application once and then selling it many times. In this way, pioneering banks and media and pharmaceutical companies have reduced the complexity and cost of managing applications and speeded up the deployment of new or updated ones.

An approach some companies have used to turn elements of custom-applications support into packaged activities involves standardizing the maintenance, support, and software-management activities that groups of applications share. Applications that support field service work, for instance, may undertake similar functions (such as call planning) or incorporate similar types of tools (such as calculators, diagnostics, or checklists for customers). A company can define the management of common elements among applications as standardized "products" designed to provide for the needs of many applications rather than one. Similarly, it can assemble new, custom-built applications from common, internally built modules of functionality and then reuse services developed by its teams to undertake common tasks such as authenticating users or accessing attributes of customers or products. Like software vendors, such a company writes the code once but uses it frequently—a tactic made possible by creating application interfaces that incorporate broadly accepted standards such as those from the [Web Services Interoperability Organization](http://www.w3.org/2001/sws/).

The upside of this approach is now quite clear. One company that adopted these support and maintenance principles reduced spending on applications maintenance by 30 percent—amounting to 60 percent of the entire applications budget—and speeded up the deployment of new applications, thereby completing in two to three weeks what once took two to three months.



Focusing in-house applications management on such products can be challenging and clearly isn't right for all companies. The payback will be greatest in fast-moving sectors. Companies prepared to try will be tempted to seek solutions for the problems of today's applications portfolios rather than those of tomorrow's—problems that are inevitably harder to pinpoint and understand. But treating support services as products is all about building for the future. Companies will need to change how they organize support resources, perhaps even overhauling IT governance structures to ensure the appropriate funding, oversight, and accountability of a very different applications environment.

### **Turning applications management into a product**

To understand what the leading companies are doing, it's worth dividing the expenditures for custom-made applications into two parts, each requiring a different strategy. One part is identifying and codifying into software the business rules that give a company its competitive edge—rules like "If a physician is unaware of our product, give her the benefits message" or "If patient compliance is an issue, discuss tools for improving compliance." This is a development task. The second part of spending on custom applications—accounting for 50 to 70 percent of their total costs—involves managing them over their lifetime. This task includes selecting the applications tools to use, determining the service levels users require, rolling out applications, operating them at target service levels, maintaining them, providing ongoing user support for them, and eventually retiring them.

In most companies today, the IT and business teams involved in developing a custom application decide independently on the type and version of the applications tools and deployment environment it will use—the server, the database, the portal, and so on. Decisions about service levels (such as availability) and policies on data storage also are made ad hoc. Such applications end up as complex beasts to manage, typically requiring a host of maintenance and support processes as customized as the applications themselves. There will be differences—sometimes major—in how they meet the demand for split-second response times or what must be done to change the reports that they produce.

More companies are setting standard levels for infrastructure services—including standards for server availability, storage capacity, and network performance. Yet the way teams manage end-user enhancements, maintenance, and support for applications still varies widely. These variations lead to spotty utilization, reduce productivity, delay the rollout of new applications, and impair the ability to meet user expectations consistently.

In response to such problems, companies have tried to reduce labor costs by, for instance, offshoring the maintenance of applications and consolidating them in shared service centers. Consolidation improves utilization (fewer machines and people are needed) but rarely raises the productivity of IT maintenance, because the diversity of maintenance and support activities doesn't go away. Similarly, offshoring can reduce per-hour labor costs, but companies (either in their captive offshore centers or their outsourcing vendors) typically do little to reduce the underlying diversity in the maintenance processes for applications. Indeed, without investment in a level of on-site support or strict adherence to processes, geographic separation may only exacerbate the problem.

But these processes can be standardized, as some leading companies are demonstrating. A company may have thousands of applications, but we've found that, for most organizations, they can be clustered into fewer than a dozen archetypes—our term for applications grouped by their key commonalities. Archetypes naturally vary by company, but broadly speaking they can include applications that support a large number of customers primarily on the Web, provide data-intensive analytical support, track performance and provide dashboard summaries for senior executives, or offer highly regulated transaction support.

The important point is that, within these archetypes, the requirements for applications management are strikingly similar, so maintenance and support can be standardized. All "road warrior" applications, for example, need tools that support offline work, the offline-online



synchronization of data, PDA form factors, and early-morning and late-evening technical assistance. Companies essentially need to standardize these processes into products, designed once and then used over and over again for different applications, within a particular archetype. When a company needs a new customized application, the team that develops it chooses a product that determines how it will be deployed, maintained, operated, supported, and even priced for internal users. This approach draws on the model used by software vendors and application service providers such as Salesforce.com.

Considerable standardization may be involved. One company, for instance, defined five such products that address the management challenges of each application archetype and are designed to derive the maximum benefit from standardization (exhibit). Together, the five addressed 60 percent of the company's applications, accounting for 80 percent of its applications budget. This company didn't attempt to squeeze all existing applications into the new model. Instead, it has looked forward: IT managers estimate that over 90 percent of all its new applications-development projects will take advantage of one of the five models from the beginning.

#### EXHIBIT

### Standardizing processes into products

5 applications-management products (disguised example from multibillion-dollar company)<sup>1</sup>

#### Tiered price levels based on labor and platform costs

#### Products

	Departmental	Simple Web site	Road warrior	Enterprise application	Highly regulated
<b>End-user demographics</b>	• Small group of internal users	• Specialized user groups (including external)	• Mobile-user base	• Users in manufacturing department	• Corporate users
<b>Service level offered</b>					
Level 1 support	• 18 hours per day, 5 days	• 9 hours per day, 5 days	• 24 hours per day, 7 days	• 24 hours per day, 7 days	• 24 hours per day, 7 days
Latency issue	per week	per week	per week	per week	per week
Availability	• No	• No	• Yes	• Yes	• Yes
Deployment cycle	• Medium	• Low	• High	• High	• High
Other	• Weekly	• Weekly	• Weekly	• Monthly • Full disaster	• Annually • Full disaster
<b>Application platform</b>	• IIS <sup>2</sup> • mSQL <sup>2</sup> • Windows	• Apache • MySQL <sup>2</sup> • Linux	• IIS <sup>2</sup> • Oracle • Windows	• SAP • Oracle • Solaris	• IBM Websphere • Oracle
	Virtualized Intel Blade servers		Intel servers	RISC servers	
<b>Application examples</b>	• Department intranet	• Recruiting Web site	• Online/offline sales system	• Supply chain management	• Financial-reporting system

<sup>1</sup> Approximately 60% of new applications fit into 1 of the 5 models; 40% are highly variable and require customization.

<sup>2</sup> IIS (Internet Information Services) is a server from Microsoft; mSQL (mini structured query language) is a database commonly used for midsize projects; MySQL is a popular open-source database.



Companies that took this path have found several benefits of standardizing applications-management activities as products:

- Reduced work for application "owners" (such as business units) and developers in sorting through management issues
- Reduced costs and enhanced cost transparency derived from per-seat, per-application prices
- Standardized, fine-tuned processes for applications of the same archetype
- Dramatically higher resource utilization
- Increased manageability of service levels
- Concentration of skills around fewer technologies, leading to better training and development
- Full value for companies that consolidate applications in shared services<sup>1</sup>

### Turning business functionality into products

Several leading companies are at the stage of defining archetypes to reduce spending on applications management. Some of these companies have begun to experiment with applying the lessons of packaged software—write once, sell to many—to the *development* of customized applications as well (see sidebar, "[The development and delivery process](#)").

In essence, such companies have taken the writing of reusable code, an idea as old as computing, into a new era. The task is to create software that performs a particular function—calculating an interest rate or targeting a customer, say—and then to reuse it in any new application that might need this functionality. In the past, locating such code and deciding how best to use it were difficult because this knowledge could be shared effectively only within a tightly knit team.

Today companies bring some order and standardization to the process by using Internet-based service-oriented-architecture (SOA) standards.<sup>2</sup> These IT advances help companies to codify business functionality in ready-to-use software building blocks much more easily and quickly, to scale up the kinds of functionality suitable for reuse in applications, and to ensure that such building blocks are employed more effectively across project teams and organizations—and maintained in a more standard fashion after an application has been deployed.

Consider the problem these companies are solving: because the applications that support old ways of working are difficult to change, many companies have found it hard to adapt their business capabilities to keep up with market developments. One health insurance company, for example, has a different claims engine for each of its three lines of business. When the applications were developed, making each one unique was logical because these businesses had very different needs. But that uniqueness has become a drag on innovation. When the insurer began to introduce more consumer-directed health plans, it had to change all three systems—and alter them again when it later introduced a new marketing approach that involved different methods for tracking customers and promotions. Each change was difficult, expensive, and time consuming.

In a company with reusable components of business functionality, such changes are a lot easier. A major bank created a library of reusable modules—essentially off-the-shelf products—that codified business functions for analytic modeling. The bank uses these products repeatedly in applications that support the trading of a wide range of financial instruments, such as equities, bonds, derivatives, and foreign exchange.

Other companies are following suit. A leading media business defined ways to codify elements (including customer profiles, lifetime value analyses, promotions, and campaign management) so that the software for them could be built once and reused. Naturally, this approach saves time and therefore money. For early adopters, the benefit that really counts is a reduction in the time needed to develop an application: they are finding that they can roll one out 20 to 40 percent faster when they use common applications products. Furthermore, the reduced expense could eventually allow companies to leverage the advantages by using easy-to-assemble applications to



test new business strategies. If the strategies work, the companies can scale up the applications; if they don't, little has been lost, because such applications are inexpensive to build and easy to discard.

It is still early to make definitive predictions. But internal, off-the-shelf components of business functionality might allow IT to deliver its true promise: promoting business innovations instead of being the boat anchor holding them back.

### **Lessons from early adopters**

As we noted earlier, this approach isn't suitable for all companies. Those that must roll out new applications quickly and constantly—banks and media companies, for instance—will benefit from it. But those in slow-moving sectors may have little need. If applications don't have to change much, writing them once is good enough—it's not worth all the work required to standardize custom activities. For businesses that can benefit from product-oriented approaches, we offer the following lessons from early adopters:

- Build the products "prospectively," mindful not just of the existing base of applications but also of future needs.
- Organize groups to deliver products effectively against business needs and not just technology outcomes.
- Pay attention to organizational factors that will ensure proper governance and realize the business benefits.

### **Build the right products prospectively**

Defining the right products to manage applications begins with analyzing how they cluster according to common needs. It is then more fruitful to focus on applications in the pipeline rather than those already in place. In fast-moving sectors, the useful shelf life of most applications is typically less than five years. By focusing on the future, companies in these sectors avoid putting effort into applications near retirement; thus, within three years, as much as 90 percent of the portfolio of custom applications can be standardized.

Admittedly, it is possible to standardize the support and maintenance of existing applications so as to reduce costs. Further, it is easier for a company to see concrete ways of reengineering existing support processes (a remediation project) than to plan the organization of support and maintenance for new applications that haven't yet been clearly defined, because budgets, risks, and organizing principles are unclear. Nevertheless, our analyses suggest that it isn't worth recalibrating the management of most existing applications. Significant effort is necessary to reengineer processes and to help people accept and abide by new standards. Since the applications portfolio turns over quickly, the costs of reengineering are barely recovered, if at all. Reengineering applications with a longer life cycle can make sense, but only a few of them exist in the portfolios of companies in fast-moving businesses.

Similarly, the best way to identify the components of common business functionality is to look at requirements prospectively rather than retrospectively. In particular, companies should focus on specific areas (such as the way prospective customers are valued) where there is a strong business interest in standardizing operations or rules across groups. This approach not only helps a company get to the truly reusable business rules but also piques the business's interest in using these products, thereby making it easier to secure ongoing support for standardizing such processes.

### **Organize to deliver products effectively**

To define and deliver applications-management products, early adopters are carving out a single, separate applications organization from the multiple existing ones. As this delivery organization proves itself, companies mandate adherence to the new, standardized approach. Typically, the





organization is global, focused on standardizing activities of the applications that support products globally. After all, there are more common elements within product groupings than in applications that are grouped geographically.

The pioneers are also forming groups to manage business functionality products, organizing them in close relation to business domains—that is, end-to-end business processes (such as order to cash or R&D). The developers become steeped in the business groups' activities and can see more clearly how to capture functionality in code. Also, business leaders can act as sponsors of these product groups, an approach that can put them on an equal footing with traditional applications-development teams. In some cases, these groups can even provide leadership in driving business process standards—leaving traditional teams to assemble solutions from the reusable code.


### **Pay attention to organizational factors**

Turning support and maintenance into standardized applications-management products is one thing; getting the organizational elements right to make the most of those products is another. Companies must pay attention to three immediate constituencies: vendors, internal "buyers," and those who finance the budget for applications development.

Vendors have to understand what the company is trying to accomplish by standardizing support and maintenance, development, or both. A vendor's specialized software offerings or hardware should be incorporated appropriately into a prospective customized application. These technologies must therefore comply with the standards chosen for the management product supporting the application or be compatible with the modules assembled to make a new application. To achieve this goal, one company uses an architecture review process—looking at an application's technical design before launching development—as a way of forcing vendors to comply with its standardization efforts. Another company encourages compliance by certifying vendors, providing awareness education, and offering price incentives and vendor performance reviews.

A company must also persuade those who own an application internally (and across the global business) to look for and use off-the-shelf business functionality, captured in code either inside or outside the company. One way to do so is to create markets for these products. Goldman Sachs, for example, built a Web portal that allows developers around the world to share software components and documentation, both internally and with certified systems integration vendors.

Finally, those financing new applications must favor standardized business functionality products (on the development side) and management products (on the support side). People should be held accountable for moving toward these new, common approaches. Costs for developing functionality and management products are ascribed to a single applications project, but subsequent applications can reuse them, and this possibility has implications for the way projects are measured and their performance is graded. Companies may find it necessary to finance standardization efforts separately from other development projects and to establish pricing mechanisms for recouping the investment from business users of the applications. Funding must take into account both efficiency and relevance, and pricing should balance recovering the investment with encouraging greater use.

Companies that live or die by how quickly they can roll out new, innovative business capabilities to their customers can benefit from making their customized applications more like products. 

### **The development and delivery process**

Old

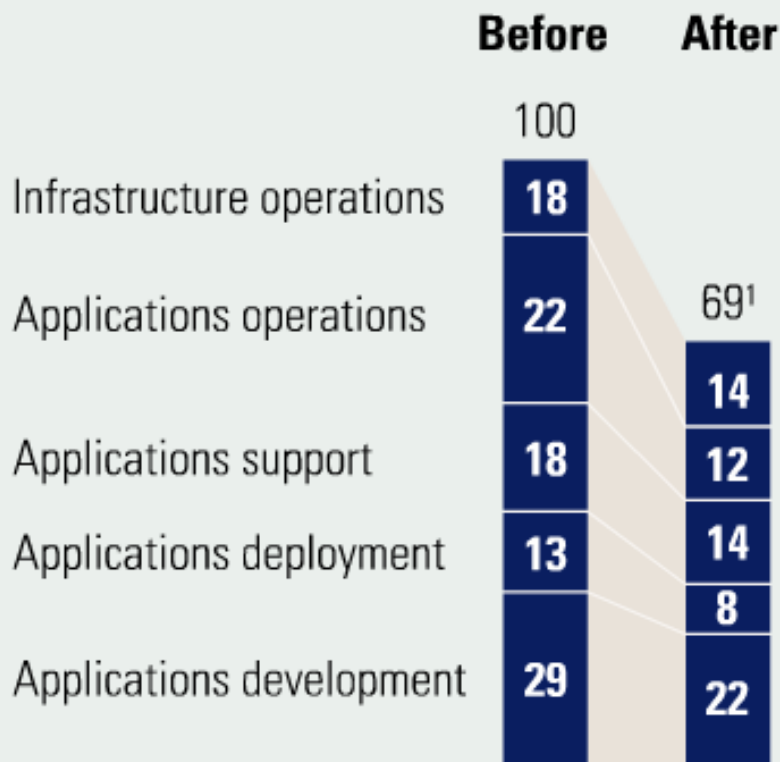
- The infrastructure staff and development vendor must design an infrastructure platform for every application.
- Dealing with exceptions to current processes (such as the needs of machines, remote access, and root privileges for the vendor) creates unnecessary overhead.



- Delays caused by procurement snags or infrastructure configuration issues typically run five to six weeks.
- Because the company has to pay vendors on a time-and-materials basis—and scale is limited—expenses mount.
- External resources are needed for routine operations such as database administration support.
- Time and attention are wasted on coordinating activities such as reboots and updates with other groups sharing infrastructure.
- Ad hoc systems upgrades and reboots often cause unplanned application downtime.

## The next generation of in-house software development

Annualized costs (disguised company example);  
index: costs before improvement = 100



<sup>1</sup> Figures do not sum to total, because of rounding.  
Source: Interviews; McKinsey analysis

New

- Predefined product options greatly reduce the time spent designing the infrastructure for the most common applications.
- A standardized process that can also handle exceptions creates a single point of ownership and accountability.



- An inventory of infrastructure options that comply with the reference architecture eliminates procurement delays.
- Leveraging vendor contracts gives the company a pool of resources at attractive rates.
- Routine tasks that were formerly undertaken by the vendor are coordinated, eliminating redundancy in the process.
- Applications operations are managed, freeing time to focus on more critical business tasks.
- Accountability for service levels helps the company identify key areas to improve processes.

### About the Authors

**Sam Marwaha** and **Ranjit Tinaikar** are principals in McKinsey's global IT practice. Sam is a coleader of the IT governance and organization practice and leads the US pharmaceutical-IT practice. Ranjit, a coleader of the IT strategy practice, specializes in retail-banking and consumer credit operations. **Samir Patil**, an associate principal, is a leader in the IT governance and organization practice. All three are based in New York.

This article was first published in the Spring 2006 issue of *McKinsey on IT*.

### Notes

<sup>1</sup>Managing applications support in this way is very similar to the approach other companies are taking to manage their infrastructure more efficiently and effectively. For more on this topic, see James M. Kaplan, Markus Löffler, and Roger P. Roberts, "[Managing next-generation IT infrastructure](#)," *McKinsey on IT*, Number 3, Winter 2004, pp. 2-9.

<sup>2</sup>Service-oriented architectures allow applications to be highly interoperable across a network by making it possible to access standardized computing or application services in a standardized way.