



## When IT's customers are external

IT organizations can minimize the risks of customer-facing projects by adopting the approach of software product companies.

Simon P. MacGibbon, Jeffrey R. Schumacher, Ranjit S. Tinaikar

*Web exclusive, January 2006*

Managing a large IT implementation is challenging for any company, but the risks increase when the end users are external customers and the price of failure is damage to the bottom line. Careful planning of the design and rollout is thus crucial. No one knows this better than software product companies, such as Microsoft and Oracle, which live or die by how well they introduce products that both meet their customers' needs and keep their cost base competitive. They realize the importance of understanding what customers want, of market research and customer support, and of maximizing value by striking a balance between satisfying buyers and controlling costs.

In these respects, software product companies have much to teach internal IT organizations, whose project-management practices usually fall short when it comes to developing and executing customer-facing systems. Too often the IT organization assumes that someone in marketing or sales has done the up-front market research or that budget-busting modifications to the features of systems are customer driven and absolutely essential. Such faulty assumptions can have serious consequences: whereas internal customers usually have no choice but to adopt a new solution—even a poorly designed one—external customers can go elsewhere. During the recent refinancing boom, for example, one mortgage company had to forgo 10 to 20 percent in potential extra revenues because its loan origination system couldn't handle the necessary volume. Rather than deal with a slow, unwieldy system, the company's "customers," namely, the real-estate brokers who sold its loans, simply used other lenders.

### A market-oriented approach

Traditional project management focuses on prioritizing, scheduling, and deploying a well-defined set of system capabilities using standard processes and tools. These are vital steps, but they must be adapted to the special needs of customer-facing projects.

The market-oriented approach of software product companies differs in two ways. First, these companies focus on customers, using market research and follow-up support to ensure that users will like and adopt a new product because they find that it meets their needs. Second, the companies sell to highly competitive markets, so they must keep their costs in line. They therefore design products simple enough to ensure that development and maintenance costs can be controlled, yet sufficiently comprehensive to meet their customers' requirements. This balancing act ensures that products are not overcustomized and that solutions target the highest-value buyers.

### Researching customers and usage

Since customer satisfaction and the adoption of products are tightly linked, product companies invest heavily in market research to gain a detailed understanding not just of functional needs but also of how the solution will fit in with adjacent systems and environments.

This research can uncover potential problems early on. A large vendor of customer-relationship-management (CRM) systems, for example, assumed that few of its target customers used a certain type of integration platform. Research showed the opposite—the platform was disproportionately dominant. In response, the vendor developed connectors to enable customers using that platform to integrate its CRM system, averting a potential disaster.



The failure to anticipate such external dependencies is common among IT organizations, even though fixing the mismatch can require costly upgrades or product modifications. A product company such as Siebel Systems, by contrast, employs marketing specialists and has an explicit process to establish an understanding of the broader environments where its products are going to be used. The design stage incorporates not just the basic functional requirements of a product but also the features and capabilities that will help customers integrate it into their suite of systems to create a solution. Another effective approach is to give beta versions of software to customers with the aim of eliciting feedback and an early buy-in before the launch of the main product.

Market research must be a continuing rather than a one-off effort. Successful product companies monitor customer adoption trends across the entire life cycle of a product and adjust development plans accordingly. Without up-to-date information, an IT project can miss the mark. One financial-services company successfully navigated through two system releases for institutional customers without hiccups, but with the third release the system ground to a halt, overwhelmed by unexpected levels of usage that exceeded its capacity. Investigation revealed that the customer adoption group had raised its forecast of transaction volumes but that, somehow, this vital piece of information hadn't made it to the system-development and testing teams, which continued working with the original forecast assumptions. The company fixed the problem by establishing an explicit communication link between the customer adoption people and the development and testing teams at the highest level of the project-management hierarchy.

### **Balancing costs with customization**

Most projects carried out by internal IT organizations suffer from budget overruns that occur for two reasons: excessive customization and changing requirements. In an effort to keep external customers—especially big, powerful ones—happy, many IT groups end up overcustomizing solutions and adding needless costs and complexity. One large consumer credit company almost doubled the time and money originally allocated to a new platform for institutional customers by trying to satisfy all of their operational and technology needs.

Product companies avoid this problem and keep their costs competitive by identifying common customer needs and then developing a basic product around a standard menu of features and functions. Additional options are bundled into a tiered offering aimed at a few higher-value customers and those willing to pay a premium. The mortgage company mentioned earlier started with ambitious plans for its new loan origination platform. Ideally, it would have supported a wide range of products (mortgages, home equity lines, and personal loans) and distribution channels (wholesale, branch, and direct). But faced with escalating costs and development time as the project team struggled to reconcile these competing priorities, the CEO imposed a financial cap. This move helped to get the project done by forcing the IT group to narrow its focus to the product and channel that had the highest growth potential and the greatest current volume, respectively.

*At one database company, the reuse of components is so central that it gives an award to whichever team "stole the most code"*

Unplanned or changing requirements should be accommodated by instilling process discipline. To reduce the number of changes ordered, many product companies use process-improvement methodologies, such as the capability maturity model (CMM), a set of process standards widely adopted by the software-development industry to improve the quality and productivity of the development process. Leading providers of offshore software-development services, for example, must have attained CMM Level 5—the highest—to be competitive. They also increase their flexibility by designing components for reuse. Eontec, for instance, has a product for retail banks that reuses services such as check reorders and fund transfers across channels. Most of the large providers of enterprise-resource-planning solutions, such as SAP and Oracle, draw on inherent commonalities of service (such as invoice generation) to develop standard but flexible packages that can be adapted to different customer situations. In *Secrets of Software Success*, the authors



report that at one provider of database products, the reuse of components is so integral to the corporate culture that an award for "the team that stole the most code" is given at the company's annual developer conference.<sup>1</sup> The financial-services company mentioned earlier, aiming to support a range of home-financing products and their complex contracts flexibly and cost effectively, created a set of reusable modules for common products and contracts.

### **Providing follow-up customer support**

High-quality solutions and strong customer support win the customers' trust, as product companies know. They reap the dividends in the form of wider adoption of their products, increased customer satisfaction, and future repurchases or upgrades.

Trust can be built in a number of ways. Siebel uses extensive beta testing to improve its products' usability and quality before releasing them. Intuit's TurboTax, a tool for calculating and submitting taxes, includes context-sensitive feedback options in its product in order to collect market reactions online. Netscape distributes free beta versions of its Web browser to gain critical mass for new releases. Companies that manage postmerger IT integration ramp up their help desk support when they first roll out customer-facing components, such as branch teller and call-center applications for retail banks.

The mortgage company took a test-and-learn approach to implementing its loan origination platform. Two pilot groups in one geographic location tested the platform under real market conditions to see which features and functions delivered measurable benefits to customers. Meanwhile, taking another page from the release models of software product companies, the mortgage company forecast how the new platform would affect call-center volumes and beefed up support correspondingly so that customers would come away satisfied with their experience and the solution.

Most new software solutions require a level of process change and "re-skilling" on the customer's part. Large IT projects typically bring customers up to speed with training documents, user manuals, and perhaps formal training programs. Software companies go further and see the postsales period as an opportunity to strengthen the relationship and build trust by offering on-site consultants, specialized help desks, Web-based communities, deployment managers, and feedback channels. Few IT projects—even customer-facing ones—approach follow-up and support with the degree of discipline shown by product companies.

### **Making it work**

Like a product company's R&D group, an IT organization must collaborate with marketing and sales to make sure that a solution offers value to customers. To do so in a disciplined way and to keep the focus unremittingly on the customer, it's necessary to reengineer the traditional software-development steps of project initiation, design, coding, testing, and deployment.

A customer-facing project is best initiated and sponsored not by IT but by the business unit that "owns" the customer relationship. It is also essential for marketing and sales to be involved in researching customer needs and for the IT organization to allow enough time—up to three or four months—to gather this intelligence during the project initiation phase. At this point, as well, the project team should identify the company's highest-value customers and their vital requirements, which will then form the core of a set of standard features. This approach keeps costs down by minimizing customization. In addition, detailed research helps IT to prioritize its development efforts and to make the right trade-offs in the subsequent design phase. Research carried out by a large asset manager, for instance, revealed that the company's inflexible approach to record keeping was a source of dissatisfaction for its largest, most profitable customers, leaving it vulnerable to competitors. Armed with this information, the company made upgrading its record-keeping platform a priority.




## Maurizio Ammannato

---

During the design phase, it is standard IT practice to involve the end user—difficult enough for internal projects but even more so for customer-facing ones, where end users cannot be strong-armed into cooperating. Project teams thus must secure the customers' involvement by using methods ranging from formal contractual agreements to informal advisory boards. One provider of global payment services formed a group of key-customer banks to help develop and roll out a payment-messaging technology. An auto-financing company with a strong culture of dealer-centered service and sales formed a dealer advisory board to test ideas for online loan origination and to win advance buy-in for new features. When a few large customers account for the bulk of a company's revenue, it is usually wise to have them participate even more actively in the design phase. Specific controls should also be introduced at this point to ensure that the marketing group or salespeople on a project team get written design approval from these customers before development starts.

The focus on the customer must be sustained into the coding and testing phases, when the development team should survey the technology environment of leading customers and work with their IT departments to ensure that the new system will integrate well. This concern is a vital one if customers have to send or receive information through financial or administrative systems that tie into the solution. An agency that buys, pools, and securitizes loans from lending institutions, for instance, must download detailed product and contract information from these business partners, a requirement that obliges the agency to integrate its systems with theirs. Final tests should include simulations to show how well a system handles specific transaction types or volumes, with customers verifying the correct conditions to test.

At the deployment stage, the project team should ensure that operations run smoothly by dispatching specialized deployment engineers (from IT) and relationship managers (from marketing) to customers' sites. A help desk is also vital during the first few months. After that, a formal satisfaction-and-review process will gather feedback for future releases.

Risky and challenging though customer-facing IT projects might be, they also offer opportunities to deepen and strengthen relationships with customers. By adopting a market-oriented approach and maintaining a strong customer focus throughout the development cycle, the IT organization can greatly increase its chances of success. 

### About the Authors

**Simon MacGibbon**, a consultant in McKinsey's global sales and marketing practice, specializes in customer life cycle management and the telecommunications industry. He is based in San Francisco. **Jeff Schumacher**, an associate principal, leads McKinsey's customer life cycle management technology practice in San Francisco, where he is based. He specializes in sales and marketing. **Ranjit Tinaikar**, a principal, coleads McKinsey's global IT strategy practice and credit operations practice. He is based in New York.

This article was first published in the Winter 2005 issue of *McKinsey IT*.

### Notes

<sup>1</sup> Detlev J. Hoch, Cyriac R. Roeding, Gert Purkert, Sandro K. Kindner, and Ralph Müller, *Secrets of Software Success: Management Insights from 100 Software Firms around the World*, Boston: Harvard Business School Press, 1999, p. 118.