



## Radically Simple IT

By designing and deploying enterprise systems in a different way, Japan's Shinsei Bank turned IT from a constraint into a launchpad for growth.

by David M. Upton and Bradley R. Staats

**E**NTERPRISE IT PROJECTS - both custom and packaged "one size fits most" systems - continue to be a major headache for business leaders. The fundamental problem with these systems is that for the most part, they're constructed using what programmer and open source champion Eric Raymond dubbed a cathedral approach. Like the great edifices that Europeans erected in the Middle Ages, enterprise IT projects are costly, take a great deal of time, and deliver value only when the project is completed. In the end, they yield systems that are inflexible and cement companies into functioning the way their businesses worked several years ago, when the project started. Despite recent improvements in the flexibility of packaged software, companies often find it exorbitantly expensive and difficult to modify their enterprise systems in order to exploit new business opportunities.

Instead of building systems that are legacy from the day they are turned on, managers can and should develop systems that can be improved - rapidly and continuously - well after they've gone live. Over the past decade, we've studied the design and implementation of enterprise IT systems and assisted numerous firms with the process. Through our work, we have identified an approach that not only reduces a company's costs but supports the growth of existing businesses and the launch of new ones. We call it a "path based" approach, because rather than attempting to define all of the specifications for a system before the project is launched, companies focus on providing a path for the system to be developed over time. The approach's premises are that it is difficult and costly to map out all requirements before a project starts because people often cannot specify everything they'll need beforehand. Also, unanticipated needs almost always arise once a system is in operation. And persuading people to use and "own" the system after it is up and running is much easier said than done.

In our research, we discovered a standout among the companies applying the path-based method: Japan's Shinsei Bank. It succeeded in developing and deploying an entirely new enterprise system in one year at a cost of \$55 million: That's one-quarter of the time and about 10% of the cost of installing a traditional packaged system. The new system not only served as a low-cost, efficient platform for running the existing business but also was flexible enough to support the company's growth into new areas, including retail banking, consumer finance, and a joint venture to sell Indian mutual funds in Japan.

The path-based principles that Shinsei applied in designing, building, and rolling out the system - forging together, not just aligning, business and IT strategies; employing the simplest possible technology; making the system truly modular; letting the system sell itself to

users; and enabling users to influence future improvements - are a model for other companies. Some of these principles are variations on old themes while others turn the conventional wisdom on its head.

### Born of Necessity

Shinsei came into being when Long-Term Credit Bank, founded by the Japanese government to assist in the rebuilding of the country's industries after World War II, went bust in 1998 with nearly \$40 billion in nonperforming loans. The firm was nationalized, then sold in 2000 to Ripplewood Holdings, a U.S. private equity fund, and renamed Shinsei, which means "new birth." Ripplewood executives coaxed Masamoto Yashiro, the former chairman of Citibank Japan, out of retirement to lead Shinsei. In addition to deciding to revamp existing commercial-banking operations, Yashiro formulated a plan for revolutionizing retail banking in Japan by offering a value proposition that was unique in the country at that time: high-quality products and services provided on a convenient, easy-to-use, low-cost basis. The strategy called for Shinsei to offer services that were then uncommon in Japan, including ATMs available 24/7 free of charge, internet banking, online foreign-exchange trading, online bilingual banking, and quick service supported by real-time database reconciliation (meaning customers' accounts were updated immediately after each transaction).

Yashiro felt that the bank needed to move quickly to seize the opportunity in retail banking. However, the firm's existing IT systems were antiquated and could not even support the bank's existing corporate business adequately. To address these issues, Yashiro hired his former colleague Dhananjaya "Jay" Dvivedi, who had led IT operations at Citibank Japan, to be chief information officer. Upon taking the job, Dvivedi quickly surrounded himself with a talented core team, most of whom had worked for him previously. Since the re-

covering bank had limited investment funds, Yashiro gave Dvivedi the mandate to revolutionize IT but with the understanding that his team needed to do it "fast" and "cheap." Recognizing that they could not fully know what the retail operation would need, the two men agreed that the goal should be to build a system that could scale with growth and adapt to new opportunities that the dynamic business would create.

The conventional choices for building a major enterprise system were two: the "big bang" approach of replacing the current system with an entirely new system and processes all at once or the incremental method of improving or replacing the existing system one small piece at a time. Dvivedi and his team were leery of taking the big bang route, believing it was too risky given the

#### Article at a Glance

Japan's Shinsei Bank used a path-based approach to build an enterprise IT system that would provide a low-cost, efficient platform for running its existing business but was flexible enough to support the company's growth into new areas.

The approach addresses the three main challenges of an IT project: It is difficult and costly to map out all requirements before a project starts because people often cannot specify everything they'll need beforehand. Unanticipated needs almost always arise once a system is in use. And persuading people to adopt and "own" the system after it is in operation is much easier said than done.

The path-based principles that Shinsei applied in designing, building, and rolling out the system - forging together, not just aligning, business and IT strategies; employing the simplest possible technology; making the system truly modular; letting the system sell itself to users; and enabling users to influence future improvements - are a model for other companies.

bank's cash constraints and knowing all too well the problems endemic to such projects. The incremental course, however, which would probably take three to five years, would be far too slow. So they decided to blaze a third path. They would put into place a new, modular infrastructure that at first would function in parallel with but eventually would supersede the current infrastructure. According to traditional IT thinking, this was madness. Much bridging software would have to be developed to span the old and the new, which would require an enormous effort.

But Dvivedi knew from his prior work and his conversations with other CIOs that technical problems were almost never the reason that new IT systems flopped. Human problems were. People typically resist adopting new systems, often because the cost (the effort) outweighs the benefits. To address this, Dvivedi used simple but innovative technology solutions to avoid the wrenching go-live experience. For example, by mimicking the old system's look and feel at least for a while, Dvivedi and his team were able to speed adoption of the new system.

While the retail unit has yet to break solidly into the black (due to the expenditures required to build the business and Japan's difficult economic and regulatory environment), the new enterprise IT system was instrumental in helping Shinsei quickly become a significant player in the retail banking market in Japan. By June 30, 2007, Shinsei had more than 2 million retail customers, up from fewer than 50,000 in 2001, when its retail business was limited to wealthy clients. *The Asian Banker Journal* named Shinsei the best retail bank in Japan in 2004 and 2005, and

David M. Upton (dupton@hbs.edu) is the Albert J. Weatherhead III Professor of Business Administration and Bradley R. Staats (bstaats@hbs.edu) is a doctoral candidate in the technology and operations management unit at Harvard Business School in Boston

*Nihon Keizai Shimbun*, one of Japan's most influential business newspapers, proclaimed Shinsei to be number one in customer satisfaction among banks in Japan in 2004, 2005, and 2006.

Let's take a closer look at Shinsei's path-based approach.

### **Don't Just Align Business and IT Strategies - Forge Them Together**

The notion that business strategy and IT strategy should be aligned and, therefore, that business users should be involved in the design of enterprise systems has been widely accepted. However, doing this has proven fiendishly difficult, for several reasons. For one thing, IT leaders struggle to truly understand the business context. What's more, business leaders do not invest the

**The first step is to focus on the foreseeable business objectives, not the existing environment. Too often, firms spend their time thinking about how current processes work.**

time required to appreciate the power and the challenges of technology and tend to treat the IT staff as second-class service providers. Even when the two groups meet to discuss a project, those occasions tend to be isolated, onetime events, rather than part of an ongoing discussion. Like it or not, however, information systems are an integral part of business strategy in almost all industries today. If business leaders view the IT staff as an ancillary player rather than a partner, then knowledge transfer between the two groups will suffer, resulting in missed opportunities and suboptimal performance.

Complicating the situation, when companies adopt standard packaged software, they often end up adapting the business to the technology. To be sure, this sometimes means that businesses abandon inefficient processes and institute best practices that are embedded in the software. But more

often than not, managers sacrifice idiosyncratic, competitively powerful capabilities that the system could make possible because developing them would add to the time and cost of carrying out an already expensive, time-intensive project.

So what should a company do to integrate its IT and business strategies? Before the planning work on a project begins, general managers need to be sure that the IT staff understands the business and takes a central role in the organization. One way to make sure this happens is to have the head of IT report to the CEO or COO (as Dvivedi does at Shinsei) rather than the CFO, as is common at many major corporations. The perception of importance often becomes the reality.

Business managers also need to have an understanding of what IT can do. At Shinsei, Yashiro and his successor, Thierry Porte<sup>2</sup>, invested substantial amounts of their time in learning about IT. Porte<sup>2</sup> speaks with Dvivedi frequently, meets with him formally a minimum of once a week, and visits the company's IT and operations center at least once a month. Porte believes that as CEO, "I have to be able to explain IT to my customers and employees so that we can meet our customers' business needs and deliver value going forward."

In project development, the first step is to focus on the foreseeable business objectives, not the existing environment. Too often, firms spend all their time thinking about how existing systems do a job and the current processes that are used to complete a task. This results in a paving of the old dirt paths. Anyone who has tried to drive the confusing streets of Boston or London will

understand the consequences of this approach.

Having identified the foreseeable business objectives, managers must build an IT strategy that fits them. This should be an ongoing effort: There must be a constant interaction between the business and IT groups about business goals and IT decisions and constraints. By engaging in iterative discussions, the two sides gradually come to speak the same language. As business users educate systems people about their needs and IT people put prototypes in front of them, new potential solutions will emerge.

The close relationship between the IT and business groups helped Shinsei fully harness technology in its drive to transform the customer's experience at the bank's branches. For example, Shinsei tellers were equipped with two screens - one facing the teller and the other facing the customer. The customer screen was the same as that on the internet banking site, while the teller screen displayed that content plus additional information about the customer. When a customer came in for a transaction, the teller would literally show the customer how to execute the transaction herself (without telling the customer that she was being trained to conduct future transactions on a personal computer at home or on an ATM). Tellers were cashless. If a customer wanted to deposit or withdraw money, the teller would walk over to an ATM with the customer and again execute the transaction with the customer's participation. Customers were never forced to serve themselves, but through its IT systems and other infrastructure (branches, ATMs, tellers), Shinsei could provide a high-level of service while training and encouraging customers to handle certain transactions themselves.

Tellers were also able to see firsthand the kinds of transactions that were troublesome for customers, generating a raft of ideas to improve the systems. For instance, when customers were conducting tasks such as opening an account

and transferring funds, they had to fill out paper forms and hand them to tellers, who then entered the information into the system. Tellers noticed that customers often chose the wrong forms or filled them out incorrectly. To fix this problem, Shinsei changed its process. Tellers now take the necessary information from customers, enter it into the system, and present a computer-generated confirmation to the customer for immediate verification.

### Strive for Extreme Simplicity

William of Ockham, the medieval philosopher, said theories should be as simple as possible. The same principle applies to enterprise IT systems: They should be designed with as few standards (such as network protocols, operating systems, and platforms) as possible - ideally one of each. While organizations usually start with a handful of standards, most allow them to multiply over time as a result of acquisitions and individual businesses' initiatives. In addition, the technology chosen or developed to satisfy specific needs should be as simple as possible, should assume there will be technical failures and have ways to mitigate them, and, as much as possible, should be reusable.

Minimal standards. Standardization of a small set of components is critical to a path-based approach. Just as Southwest Airlines has reaped benefits from flying only Boeing 737s, simplifying the IT infrastructure allows a company to reduce complexity, deepen specialized expertise, and increase the potential for reusing elements of the system - all of which accelerates development and lowers maintenance costs. In addition, standardizing components allows organizations to devote less time to maintaining quality and more to building new functionality.

Indeed, one of the biggest sources of added value that external big-bang software providers can bring is often that they limit the standards that are used in an organization to their own proprietary set. Most IT managers do



not have the power or long-term discipline to hold the line themselves. That means business managers must have an understanding of technology to appreciate the trade-offs and help avoid the exceptions that can balkanize IT systems.

Dvivedi ruthlessly drove standardization at Shinsei. He solicited input from relevant players but did not wait for consensus to act. One of his most radical decisions was to eliminate Shinsei's mainframe systems, the traditional backbone of a bank's IT, and replace them with Intel-based servers. This was a significant change from the prevailing practice among banks in Japan and most financial services firms around the world, which loved the high throughput speeds and dependable uptime of mainframes. The problem was, mainframe systems were expensive and difficult to keep up (the annual cost of maintaining a typical mainframe is 15% to 20% of the original purchase price). The software for older mainframe-based systems is usually written in arcane proprietary languages, which are difficult to untangle even if you can find programmers conversant in the code. The switch to a server-based platform immediately saved the bank \$40 million in expenses

seï to address business issues as they arose. This involved a straightforward process that they followed consistently. After first identifying a business issue, the team would break it down into its constituent pieces and would then determine a technology solution for each one. To do so, they first delved into their tool kit of standard modules and components to see if a solution already existed. If they did not have a solution in-house, they looked outside for an off-the-shelf solution. If none were available, they would turn to one of five or six independent Indian software-services partners to develop the capability.'

The invention and deployment of Shinsei's entirely new ATM network is an excellent illustration of this approach. In 2000, other Japanese banks charged fees for using their own ATMs as well as those of competitors and offered ATM services only during branch business hours. Shinsei realized that if it wanted to offer free transactions 24 hours a day, it could not build a replica of others' costly ATM networks. Accordingly, the project team of business and IT people precisely identified the functionality that they needed to offer and then broke down the issues involved as far as they possibly could. That allowed

ability than a leased line at one-tenth of the cost. Altogether, the new approach allowed Shinsei to offer customers ATMs that were always available and provided greater functionality than competitors' traditional networks at a fraction of the cost.

Another example of an ingeniously simple solution is the IT team's cure for failures stemming from memory leaks. Any operating system may be prone to memory leaks - which happen when an application does not release the memory that it used once it has finished a task. Eventually, the operating system may run out of memory, become unstable, and crash. The commonly prescribed remedy is to design sophisticated memory management and debugging tools to try to prevent all memory leaks. However, such tools cannot reliably plug every leak. Shinsei decided to simply assume that memory would leak within its servers and to have them perform a staggered reboot at frequent intervals - a crude but effective and inexpensive solution.

Whether Dvivedi's staff developed a component internally or asked an outside vendor to provide it, the component had to be reusable in other projects at Shinsei if at all possible. To make sure of that, the team clearly specified the required function of the component and the standard interfaces that would ensure it could "speak" to any other existing or future modules. For instance, business and IT personnel at Shinsei realized early on that credit checking was a key process in a number of services that the bank offered. Therefore, the team developed a reusable module for credit checking that could be deployed across products. Today, more than 90% of the technology components are used in more than one place.

**Modularity, not just modules.** While the prevailing view that big IT programs and systems should consist of modules is hardly new, the concept of modularity is often misunderstood. Just because a software developer claims that the various parts of its applica-

### Business managers must have an understanding of technology to appreciate the trade-offs and help avoid the exceptions that can balkanize IT systems.

annually. Along with selecting a single server platform, Dvivedi and his team chose other standards, such as Dell PCs, Microsoft Windows, the internet, IP phones, and standard messaging between business systems.

Simple, **reusable solutions.** Not wanting the new bank (the retail as well as the commercial and investment-banking units) to be shackled by the capabilities of the existing technology, Dvivedi worked with his team to create an architecture that would allow Shin-

them to solve problems in new, more cost-effective ways. For example, traditionally, ATMs were connected to a bank's back-end systems through expensive leased lines. The team realized that the internet could serve the same role. The uptime, or reliability, of an internet connection, though, was worse than that of a leased line. The simple solution: Design the system to expect and deal with failure by installing two internet connections from two different providers. This yielded better reli-

tions are modules does not mean that they are actually modular. Modularity involves clearly specifying interfaces so that development work can take place within any one module without affecting the others. Companies often miss that point when developing enterprise systems. For example, we know of an automobile company that had teams working on multiple modules of a new enterprise system and claimed to have a modular design. However, one team was in charge of interfaces and was constantly changing them. Every alteration by this group forced all the other groups to spend huge amounts of time redoing the work they had already completed. Rather than limiting the impact of changes by embracing modularity, this company had actually amplified problems!

A truly modular architecture allows designers to focus on building solutions to local problems without disturbing the global system. With small, modular pieces, the organization can purchase off-the-shelf solutions or turn to inside or outside developers for a certain piece, accelerating the speed of development. Modular architecture also makes it easier to upgrade the technology within modules once the system is up and running.

Breaking down and solving problems in this way offers a number of advantages beyond speed. It allows the IT team to concentrate on obtaining the lowest-cost solution for each part and (by partitioning work) reduces the impact of a single point of failure. Clearly specifying the functions of modules and the interfaces makes it easier to build a module that can be reused in other applications.

The modular approach was a critical part of achieving the bank's strategy, as Dvivedi described it, "to scale up and expand into new activities with ease, to be able to service the needs of the organization as it grows from a baby into an adult...and avoid building capacity before we need it." Take loan-processing capabilities. The project team rolled

out the capabilities in small stages for three reasons: to prove to management that the computer system would perform as promised, to avoid overwhelming managers and users with too much automation all at once, and to be able to address any technical issues quickly as they arose. Accordingly, the team initially sought to show that the system could correctly approve credit for a

**A truly modular architecture allows designers to focus on building solutions to local problems without disturbing the global system.**

small number of loans (20 to 30 a day). Then the team developed the capacity to fully process 200 to 300 loans a day. As the business grew, Shinsei eliminated manual work to reach a capacity for processing 6,000 loans a day.

Thanks to the modular structure of the automated system, Shinsei can simply replace one part (the loan-application or credit-checking functions, for example) without affecting the rest. What's more, modularity has allowed Shinsei to change its IT when appropriate or necessary without having to risk upsetting customers. It can keep the customer interfaces (such as web pages or the format of the ATM screen) the same while changing the back-end systems.

### **Give (Some) Power to the People**

Many of the failures in large IT projects stem from organizational resistance—users torpedoing new systems—rather than from technology that does not work. Sometimes the problem is that the company tries to force the system on its people, and they rebel; more often, people simply do not see a compelling reason for making the effort to learn how to operate the new system.

In general, firms should not have to sell new systems to users; rather they should build systems that users willingly

embrace. This is not to say that every technology will be adopted enthusiastically by every member of the organization. But if a system is universally hated long past the "get to know you" stage, it is likely that the system needs significant improvement or should be scrapped.

When Shinsei rolls out a new system, it starts the process by offering

an interface, or screen format, that is similar to that of the old system. A Shinsei employee starting to use the new system for inputting loan applications could navigate to a page that was an exact replica of the old system's data entry screen. However, if the new loan-application form required information (a customer's mobile phone number, for instance) that was not included in the old screen, the employee would have to go to the new data entry screen to type in the information. By going back and forth in this fashion, the user would become accustomed to the new system. Only after the vast majority of users make the transition to the new system does Dvivedi's team turn off the old data entry screens. This approach will add to costs in the short term, but Dvivedi believes it is a small price to pay for the quicker and more enthusiastic adoption of the new system.

**Continuous improvement.** Dvivedi's belief in user power extends beyond the rollout stage. It also applies to continuous improvement. One bedrock principle is that any continuous-improvement effort will fail without the committed involvement of users. Shinsei actively solicits users' ideas for improving its enterprise systems, involves them in daily experimentation, and strives to make them feel that their input matters. It realizes that if people do not feel their

ideas are being listened to and acted on quickly, they stop offering them.

To that end, Dvivedi and his team created a system for addressing feedback and requests from customers, busi-

**When continuous improvement is an integral part of daily work, the need for catchy slogans to inspire the workforce and heroic problem solving greatly diminishes.**

ness users, and technical users. In recent months, such comments and requests have averaged about 100 a day. Both suggestions and system failures generate electronic work orders, which are routed to the relevant personnel and escalated to higher levels in the organization if they stay open for too long. When an issue has been resolved, the person who raised it is notified.

For example, the feedback system helped business leaders detect that something was amiss in the mortgage business. When customers applying for home loans complained that they had already sent requested documents to Shinsei that the system showed as outstanding, managers were automatically notified of the problem, and a team of business and IT personnel was assigned to find the root cause and address it. When the team studied the issue, it found that the actual problem was that the bank had been sending customers a list of the documents to submit, but customers weren't certain what the documents were (for example, they did not know what their deeds looked like). As a result, customers submitted the wrong documents. The team's remedy: Have the IT system automatically identify the unique set of documents that a customer had to submit, and then send the customer samples of those documents, no more and no less. By making sure that customers send the correct documents the first time, Shinsei has reduced the time it takes both customers and the bank to process the documents and increased customer satisfac-

tion. All this may not sound like a major breakthrough - and that is precisely the point. When continuous improvement is an integral part of daily work, the need for catchy slogans to inspire the

workforce and heroic problem solving greatly diminishes.

Companies currently spend about 5% of their revenues on IT. While there is a large variance in this number, there is an even greater variance in the benefit companies get out of their IT. If anything, the already daunting task of choosing the right IT systems - those that can support the business strategy, provide a competitive advantage, and serve as a platform for growth - is getting more challenging. That's because

the choices are greater, are changing faster, and are growing more complex with the advent of cheap processing power, network capability, and sophisticated IT vendors in developing economies.

In such a world, businesses must focus on building IT systems that cannot fail to improve. This outlook can be very disquieting for the traditional manager who thinks that constructing a major IT system is like putting up a warehouse: You build it; then it is finished. But that does not work for IT anymore. If you take that approach to building enterprise systems, you will get rigid, costly systems that are outdated from the day they are turned on. If you adopt the path-based approach, you'll get flexible systems that can change as the business demands and can shift IT from being a simple platform for existing operations to a launchpad for new functions and brand new businesses. 

Reprint R0803J  
To order, see page 135.

