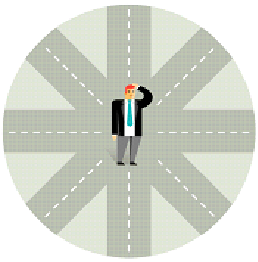
Deloitte.
CIO Insights and Analysis from Deloitte

CONTENT FROM OUR SPONSOR

Please note: The Wall Street Journal News Department was not involved in the creation of the content below.

A Path Forward for Legacy Systems

Outdated technologies can hamper IT's efforts to support modern business needs, but CIOs have options for overcoming the problem.



Digital technologies are creating a world of new opportunities for businesses today, but legacy technologies can hold companies back. Not only are most of these older systems relatively inflexible, but they're also increasingly expensive to maintain—particularly given the dwindling pool of workers trained in their upkeep. For organizations that still rely on legacy technology, the promise of digital transformation can seem like a far-off dream.

As partners at Deloitte Consulting LLP, Haissam Issa and Marlin Metzger are intimately acquainted with this problem. Issa leads the organization's systems integration practice globally, and Metzger leads its application modernization offering. In this interview, they share their perspectives on the challenge and ways to overcome it.

It would be difficult to find a large company today that doesn't still depend at least to some extent on legacy technology. What are some of the implications of this dependence for IT and for business as a whole?

Issa: These stable systems have supported companies' core businesses for decades, and although they're typically inflexible, in general they've worked well. For that reason, CIOs have often kicked the can down the road, so to speak, leaving the problem of upgrades for the next tech leader to tackle. They've been reluctant to make any big IT changes for fear of compromising their core business.

Now organizations are increasingly digital. They are also changing very rapidly, and these legacy systems can lack the flexibility and scalability to keep up—especially given today's tech-savvy end consumers. They are often handcuffing the business with what we call legacy technical debt. It's a real problem. Modern concepts like microservices, the cloud, and mobile computing are simply beyond many legacy technologies. Meanwhile, thanks to what's sometimes called the silver tsunami, many IT professionals who have been supporting these older systems are approaching retirement, creating a further sense of urgency about modernizing legacy technologies.

How would you summarize the costs of this burden, both financially and in terms of business competitiveness?

Metzger: There are a few ways to look at this. Take banking, for example. An internet bank, built from the ground up with modern technologies, can generally offer lower-cost products than a traditional bricks-and-mortar bank. With a traditional bank, when money is transferred from one account to another, there's often a delay during which the bank indicates the transaction is "pending." That's typically because of the legacy technology running in the background to effect the transfer. A purely digital bank wouldn't face that delay.

Now think about retail. Clearly, most retailers have higher peak demand for resources around Christmas than they do in May. Modern technologies like the cloud give them scalability and the ability to spin up resources as needed. That's a huge advantage over the legacy scenario, where they'd likely have to invest in a substantial amount of additional computing power to get that extra capacity.

Core-related expenditures are the single biggest line item in the IT budget at many companies. Given hardware costs, third-party software costs, and the flexibility to make changes, moving away from legacy technologies and into a modern environment can deliver significant savings.

What are some of the potential benefits of modernization?

Metzger: Flexibility, scalability, elasticity, cost savings, and the ability to avoid an imminent skills shortage probably top the list. Overall, modernizing helps enable the IT function to meet today's and tomorrow's business needs without worrying about what part of a legacy system might break along the way.

What are the main options for companies seeking to address this problem, and what are their pros and cons?

Issa: Aside from building a brand-new system from scratch, which is generally very costly and time-consuming, or "borrowing" technology that was developed for other organizations—which is sometimes an option in the government space, for example—most companies have two choices:

Off-the-shelf software. Plenty of offerings can address standard needs such as CRM or HR. Typically, though, these can't address a company's core business, or essential competitive elements such as a proprietary pricing algorithm. That leaves a big gap.

Automated refactoring. Another option is to use an automated refactoring service that migrates the technology onto a new platform and a new programming language. One big benefit of this option is that companies can compare the output of the new system with that of the legacy one—if they are identical, the core business has been preserved, and modernizing can begin from there. However, if the current legacy system can't be modernized to meet the company's needs, then refactoring is not a good option because the result will have the same problems. In this situation, companies may be best served by developing a system from scratch.

What does automated refactoring entail? What does the process look like?

Metzger: Automated refactoring can preserve a company's core systems while raising their performance to current standards. Essentially, it takes an incremental approach to replacing, or refactoring, the dated user interfaces, code, and data structures while modernizing the related business practices. Code written in COBOL, Natural, or PL/1 can be converted to Java or .Net, for example; data can be shifted from VSAM, IMS, or ADABAS to IBM DB2, Oracle, or another relational database management system. Because it's fully automated, it can be faster, less expensive, and more reliable than a manual upgrade. And there's no "code freeze" or operational downtime to disrupt the organization's day-to-day business at any point in the process. At the end of the day, it can leave the enterprise better able to handle current needs such as the cloud, mobile, and analytics.

Can you give an example of a company that has successfully used automated refactoring?

Issa: We worked recently with a client in the transportation industry whose legacy systems limited its agility, increased costs and time to maintain code and implement changes, and hurt its ability to produce timely reports. Through automated refactoring, the system's 700 screens, 3,000 code modules, and 120 ADABAS files were modernized over a period of two years. The result was an estimated 60 percent savings per year in personnel, hardware, software, and facility costs.

Any tips or suggestions for CIOs struggling with this issue?

Metzger: A good way to begin is by taking the time to fully understand the legacy environment. CIOs can sometimes make the mistake of assuming they know everything about the system because they know the core, but often there are also dozens of other smaller apps that tie everything together.

It's also important to establish a vision for the future environment, including whatever mix of cloud and on-premises software makes sense. Then the company can look at each of the apps in its current inventory and outline the options—it's never one-size-fits-all. From there, the company can start planning a road map based on business priorities and divide the modernization effort into small, agile releases.

Related Content

- "Core Modernization Benefits Finance"
- "Reimagining Heart-of-Business Core Systems"
- "IT Modernization Tips From, For Federal CIOs"

May 9, 2017, 12:01 am