



# All legacy systems go

Virtually every large organization is looking for a way to modernize its legacy systems safely, efficiently and cost effectively. Dell is making it happen.



Advances in technology come so quickly these days that businesses have a hard time keeping pace. In the last 10 years, roughly 701 percent of all applications used by Fortune 5000 companies run in legacy environments built 20, 30, even 40 years ago, according to one analyst group. Another top tier analyst firm estimates that a typical corporation spends between 60 percent and 80 percent of its IT budget simply to maintain existing mainframe systems and applications.

And these legacy systems don't only absorb resources; they can also hamstring a company and keep it from moving forward. Most aging systems have reached a point where they can no longer adapt, grow and do what an evolving business needs them to do. It's possible that, with constant tinkering, companies could squeeze more out of their old systems, but today's IT workers tend to have completely different skill sets than their predecessors, and they have a difficult time supporting and operating applications and technologies installed when they were in grade school.

Of course, the cost of replacing and re-architecting these old systems can seem high. But the cost of doing nothing can be much higher. For example, there's the financial institution that was hemorrhaging market share to an online competitor because its outdated technology could not process transactions as cheaply or efficiently. And there's the Canadian province that was losing track of its prison inmates because its old, inflexible case-management system couldn't do the job.

Modernizing critical legacy applications is hard. You want to focus on your core competency, and not be in the business of maintaining custom code. To realize lasting benefits, you simply cannot afford to take the easy route. You need a modernization solution that aligns your technology with your business needs for better, lasting results. And, above all, you need assurance of project success, especially when it affects your critical applications.

“We thought the tools would be important, but they were more than important. They were absolutely essential. We got exactly what we wanted with very specific methodologies and disciplined application of that methodology... It’s a nice balance, I think - a blend of theory and practice applied to technology appealed to us.”

*Peter Huggard  
SVP, Canadian Software  
Operations, P2E*

## New Systems, New Problems

Of course, it’s perfectly natural for a company to hesitate when faced with replacing systems that have served them adequately for years or decades. It’s not just the cost involved. It’s the possibility that the new systems will not perform as advertised, that they may actually do more harm than good. There are plenty of examples of organizations that have plugged in new systems, stood back and watched the sparks fly.

The FBI came under withering criticism for delays and cost overruns that accompanied its efforts to modernize its case-management system. Among other things, the FBI was attempting to replace green screens with an intuitive Web-based user interface, according to a recent article. “After more than three years and \$334 million expended on development and maintenance...the FBI does not have a current schedule or cost estimate for completing the project,” says a new report issued by the Department of Justice’s inspector general.

The Department of Homeland Security, which has already failed three times at modernizing its financial systems, is on the verge of a fourth losing effort—at a cost of \$450 million. And the state of Indiana is currently in court with its technology partner over a disastrous initiative to update its welfare administration system.

## The Right Way to Upgrade

Why do many modernization projects fail to produce results? The number one reason is that organizations fail to capture the decades of knowledge inherent in their legacy applications and the associated extensions or augmentations that have evolved to address changing business requirements. When applications are replaced, all too often that “tribal” knowledge is lost.

Over the life span of a typical legacy

system, users develop coping mechanisms. These may take the form of workarounds, including: manual procedures, repurposing unused data fields, spreadsheets, cheat sheets, etc. Workers come up with all sorts of ingenious ways to bridge the gap between what the legacy system can do and what the business demands of them. Dig inside a typical legacy system and look at a line of COBOL code written 30 years ago—it probably does not resemble what is actually being done in the business. That’s usually because workers have augmented that code with additional processes and procedures over the years. Often the changes aren’t documented, they’re passed down and around from employee to employee.

If your company could capture this tribal knowledge and transfer it to your new system, you have a much better chance of a trouble-free modernization program. That’s where Dell can help. Dell can mine the tribal knowledge that resides in your system or application by employing proprietary methodology

## Dell Re-architect Solution

### Legacy Language Experience

COBOL, RPG, PL1, Natural, VB, Mumps, Maper, Assembler and more.

### Projects at a Glance

Over 30 successful projects with greater than 100 million lines of code completed

### Industries Served Highlights

Local, State and Provincial Governments

Oil and Gas in Canada and the USA

Retail and Commercial Systems

Financial Services & Insurance

Healthcare & Life Sciences



and a fit-for-purpose toolset. But it's not simply a matter of analyzing millions of lines of code. Dell addresses your legacy system holistically, as an ecosystem of software, hardware and human beings that together drive your business operations.

Part of the process includes working with representatives of your user community. They are the true experts in business processes at the everyday level. Another part involves understanding your current means of operation and nuances of your legacy environment by capturing and mapping business rules and discovering how users have evolved and changed system processes through manual workarounds necessary to perform their jobs.

When all user workflows are captured, we modernize these activities. This involves tracking all requirements and quickly identifying reusable, missing and unused assets in the legacy application. We work with you to replace the many lines of legacy code with middleware or off-the-shelf components. We re-architect the remaining business-relevant custom code that differentiates your business into re-usable services on top of your optimal service oriented architecture.

### Case Study 1 - A Well-Oiled Transition

In every modernization project, one challenge is paramount: Ensure that the modernized system retains all the legacy functionality needed to run the business. That was the hurdle faced by P2 Energy Solutions (P2E), a software and service provider for the oil and gas industry. The company needed to re-architect and enhance its core application—and move to a Web 2.0 user interface—in order to stay competitive in its market.

P2E had more than 20 years of functional enhancements and tribal knowledge invested in its legacy system, which helps oil and gas companies

manage their financial assets and make payments. Among other tasks, the application works out who must be paid when oil leaves a well and journeys through a network of underground pipes to the refinery. This can be a complicated equation, because the owner of a particular plot of land must be paid a fee based on a large number of variables, including the amount of oil traveling through the pipeline, individual contracts, geographical location and relevant pieces of legislation. To calculate the correct figures, the P2E system was embedded with scores of business rules.

P2 Energy Solutions needed to transfer all of this accumulated information from its legacy system to its new system. It also needed to migrate a host of industry-specific functionality from the old system to the new system. P2E could simply not afford to lose any of that knowledge. It was the core of its business.

After reviewing several options, including an in-house build and international outsourcing, P2E was introduced to Make Technologies. After some deliberation, P2E decided to go with Make Technologies because it appreciated the effectiveness of the company's integrated modernization tools. The structure and predictability of the modernization process was one of the key decision drivers for P2E's selection of Make Technologies.

Make Technologies' solution, based on its modernization software, applied in conjunction with its methodology, ensured the success of P2E's modernization efforts. The project was completed faster, more easily and less expensively than P2E's most optimistic estimates. The cost saving versus either a Greenfield development or off-the-shelf software was dramatic. At the end of the day, P2E's modernization cost

	The Dell Data Workbench Migrates data continuously and in parallel with application re-architecture.
	The Dell Code Generator Generates code from Dell Designer into a service-based architecture, boosting productivity and quality.
	The Dell Designer Creates modern application models based on legacy requirements discovered with Dell Analyzer, and incorporate necessary enhancements.
	The Dell Analyzer Reverse engineers requirements by capturing how users actually use the software and discovering the connections between application functionality, code and data in an interactive and collaborative environment.
	The Dell Repository enable companies to catalogue and understand all the assets in their legacy applications by creating a 360 degree view of application usage, including how people operate these applications, what data they use, and how your applications have been modified over the years.

*The Dell Re-architect Solution doesn't just modernize the code, it also migrates the data – assuring that all of the features your organization depend on remain intact and complete.*



came to less than 20 percent of what a competitor paid to develop a similar system.

### **Case Study 2- A Healthy Approach to Modernization**

Organizations these days are addressing system modernization in a variety of ways. Some are going down the path of code conversion, but this often provides only temporary relief. Others are desperately trying to customize off-the-shelf software packages to meet their unique business requirements.

Still others are opting to fix the problem once and for all with a modernization effort that actually delivers.

When New Brunswick Health, the provider of universal health coverage to the residents of New Brunswick, Canada first approached Make Technologies, it had already experienced challenges to successfully modernizing. The organization had previously attempted to replace its aging, unwieldy legacy system with off-the-shelf software. However, New Brunswick Health soon realized the software would have to be highly customized and that the project would ultimately cost millions more than it was willing to pay. So it scrapped that effort. A second attempt was equally unsuccessful.

But modernization wasn't a luxury for New Brunswick Health, it was a necessity. After decades of tweaks, tinkering and staff turnover, the system was a black box. Nobody fully understood the code base anymore or how to maintain the business rules. There were few employees left who knew the application and almost all of them were at the point of retirement.

Yet New Brunswick Health was reluctant to make enhancements to its legacy system for fear it would cease to accurately pay doctors.

Within months of engaging Make Technologies, however, New Brunswick

Health had a complete inventory of the existing legacy system. Armed with this knowledge, New Brunswick Health has been able to confidently execute a successful modernization program. Ultimately, 1 million lines of COBOL code have been rewritten in Java. The resulting Java code base is approximately 80 percent smaller than the legacy COBOL code base, which should translate into significant savings in maintenance costs.

The upgrade brings the New Brunswick Department of Health peace of mind. What's more, significant cost savings are expected. A re-architecting of its systems has allowed the organization to move off the mainframe and to implement more than 200 major enhancements previously thought to be unfeasible.

### **Case Study 3 - Shifting Gears Smoothly**

The province of Nova Scotia was stuck in neutral. It knew it needed to modernize its mission-critical Registry of Motor Vehicles and Vital Statistics systems. It was spending \$330,000 per month simply to maintain and host these legacy systems.

Its intentions were good. It had a long list of planned system enhancements. But they were too expensive and too time-consuming to implement in the current legacy environment.

For example, the provincial government had mandated an "alcohol interlock" system that can track drivers convicted of DUI offenses. A Breathalyzer is installed in a vehicle and the driver has to blow under the prescribed limit before the car will start. But this system enhancement alone would have cost the province \$1 million and taken a year to implement. It simply was not feasible within the limitations of the existing legacy system.

Push had come to shove for Nova Scotia. The province first looked at a variety of off-the-shelf vendors to modernize its

systems. Their estimates were mostly in the range of \$40 million. The province then turned to Make Technologies, which offered to do the work for a fraction of the price.

And Make Technologies delivered. The 2.4 million lines of code that made up Nova Scotia's legacy applications were reduced to 450,000 lines of Java and 190,000 lines XHTML code. The modernized applications were reduced in complexity from 4,138 screens to 651 screens and 1,071 reports to 496 reports. Make Technologies created a system with a simplified program base, resulting in significantly reduced maintenance costs. At the same time, all legacy functionality required by the government was fully supported in the modernized application.

Nova Scotia's monthly system maintenance costs dropped as much as 85 percent. As for the alcohol interlock enhancement, it was implemented four months after the modernization at a third of the original time estimate, and at less than half the original cost estimate.

### **Conclusion**

With intense pressure to cut costs, stay competitive, and adhere to regulatory requirements, virtually every large organization is looking for a way to modernize its legacy systems safely, efficiently and inexpensively. We focus on how your applications support your current and emerging business needs, along with your unique organizational requirements. Dell enables companies to reduce risk, integrate low cost off-the-shelf software, adopt standardized technology components, and minimize the costs and risks of implementation and operation. This high value approach positions your company to respond to market changes more swiftly and effectively going forward. We provide an open, evolving application structure that is self-documenting, easy to update, and is based on open standards that requires no proprietary hardware or software.

