Photo credit: Getty Images

Why utility boards should care about IT architecture

Eelco de Jong, Anand Mohanrangan, Aditya Pande, and Parker Shi

Many utilities struggle to match the customer and commercial expectations set by digital leaders. Investing in faster, more flexible IT architecture can accelerate their digital modernization.

Digital companies make decisions using real-time data from many sources, reconfigure processes to meet new customer needs, deploy new functionality every few weeks, and update systems every few days. Imagine how a utility with these capabilities could respond to a power outage. Using live operational metrics, meteorological information, drone images, customer data, and social-media feeds, it could warn customers in advance about likely outages, dispatch crews before the event, and issue regular updates on when power will be restored. Should an outage be widespread or protracted, the company could scale systems seamlessly and maintain outage services flawlessly.

This probably sounds like science fiction to an executive at an average utility with IT systems dating back decades. Those systems will have grown bigger, more cumbersome, and harder to maintain over the years, with millions of lines of custom code written in legacy programming languages such as COBOL by developers who have long since moved on. For the many utilities that have undergone M&A, the blending of IT systems will have created yet more complexity. And when the stormy season arrives, it can bring unplanned disruption to utility business services that are critical for managing customer outages. Little wonder that leaders battling so much uncertainty are on edge, hoping that systems stay on.

Serving customers better will inevitably mean updating this IT architecture. But replacing a whole stack of legacy systems is costly, slow, and risky. Fortunately, digital innovators have shown there's a better way, one that delivers rapid impact and more long-term business value at lower risk.

Five inconvenient truths that hold utilities back

The challenges utilities face are similar to those tackled in recent years by banks, telecom operators, and retailers, among others. Taking lessons from their experience, we describe five inconvenient truths that hold utilities back—and suggest how to address root causes and secure long-term impact.

Inconvenient truth #1: It doesn't make sense to replace one dinosaur with another

Responding to customer and industry trends, many utilities are investing in new websites, field-force functionality, CRM tools, and the like. These usually involve a one-shot replacement of a legacy system—a customer-information system, an outage-management system, or some other piece of IT infrastructure. But the trouble with replacing millions of lines of legacy code with an equivalent system in a modern language is that it simply saddles the utility with another dinosaur a decade down the road.

Research we conducted in collaboration with the University of Oxford shows that these big-bang programs, however effective they may be for engineering projects, are highly risky when it comes to IT. Among a cross-industry sample of 2,175 IT projects, including 452 costing €10 million or more, 64 percent suffered cost overruns and 78 percent overran their schedules (Exhibit 1).

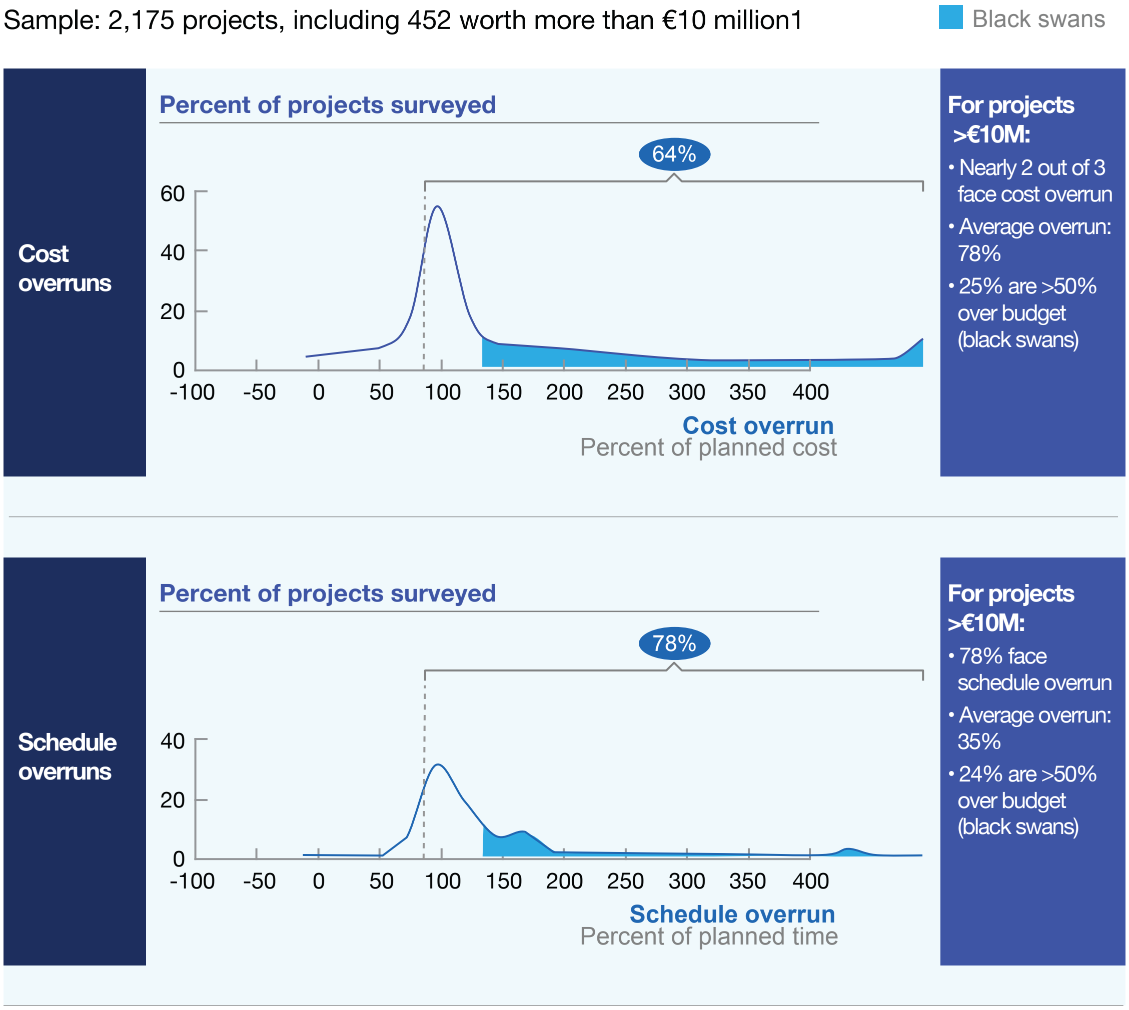
Soon after one Asian bank replaced its legacy system with a comprehensive platform from a vendor, it hit a problem. It wanted to re-engineer customer engagement in the branch, but its teller system and middleware couldn't accommodate the changes, and seemingly simple enhancements, such as adding more currencies to the trading system, could cost a year of effort. Even as the bank identified more and more needs its system couldn't meet, it had to spend more and more to keep the platform operating. Eventually, after having invested more than $1 billion in a system expected to last decades, it decided to start again from scratch. This time it opted for a modular approach, with systems that could be designed, piloted, refined, launched, and updated separately without knock-on effects for other systems or parts of the business.

One of the advantages of a modular approach is that a utility can test and release hundreds of lines of code at a time instead of millions. This takes some of the complexity out of software development and allows new hires to become productive much earlier. Instead of spending months familiarizing themselves with the system, they can focus on a few microservices and get to work straight away.

Admittedly, a modular approach does make the technology architecture itself more

EXHIBIT 1 Major IT projects often suffer cost and schedule overruns

Sample: 2,175 projects, including 452 worth more than €10 million1



For projects >€10M:

- Nearly 2 out of 3 face cost overrun
- Average overrun: 78%
- 25% are >50% over budget (black swans)

For projects >€10M:

- 78% face schedule overrun
- Average overrun: 35%
- 24% are >50% over budget (black swans)

complex: instead of one system, you now have many. And as organizations speed up their adoption of new technologies and try to be more agile and open, that complexity will continue to increase. However, as leading organizations are finding, the risks and costs are manageable and easily outweighed by the business value delivered by an accelerated rather than big-bang approach to IT architecture modernization (Exhibit 2).

Inconvenient truth #2: Opportunities to create business value are often overlooked

Utility leaders tend to operate their IT systems like their physical assets. They plan for a year, build for three, run the new entity for decades until it falls apart, and then replace it, aiming to recoup the cost over five to ten years. The trouble is that a replacement system created in this way will reflect old ways of doing business and miss the opportunity to harness new technologies to simplify processes and embrace new possibilities.

Moreover, utilities are all too often content to cover the cost of their new system instead of aspiring to generate twice or three times as much in new value. By adopting a more radical and ambitious vision, they could improve not only the efficiency of their operations and the stability of their legacy systems, but customer satisfaction as well, enabling them to capture additional business value from new revenue streams. The greatest benefits come when this value is of board-level relevance—such as a lift in customer satisfaction or a drop in outages—and when it is captured not after a six-year program, but six months into a project and regularly thereafter.

Smart utilities start by identifying key business outcomes—game changers—and assessing the value they drive. Then they work out which capabilities they need to realize these outcomes, which ones are missing, and whether the gaps are best filled by processes or technology. Finally, they identify the minimum viable set of IT changes needed to deliver the desired results.

This approach allows utilities to make progress while identifying areas that need deeper changes: rewriting some bits of legacy COBOL functionality as microservices, for instance, while flagging others for longer-term replacement. In the old days, this would have involved collecting thousands of business requirements and spending years developing solutions; now it's about IT and the business working closely

EXHIBIT 2 An accelerated approach to IT architecture modernization can deliver significant value

	Category	Examples of impact achieved
Sources of value from an accelerated approach	Improve effectiveness	• Improve conformity to service-level agreements • Reduce the most severe incidents by 50%+
	Improve efficiency	• Reduce spend on core maintenance by 10–20% • Reduce time to market for simple changes from months to weeks or days
	Reduce risk of modernization efforts	• Avoid the cost and scheduling risks of a typical mega project
	Enable technology evolution	• Reducing investment in core IT allows increased spending on initiatives to grow the business, from 20% to almost 40%

together to build functionality incrementally and keep it aligned with strategic value.

Inconvenient truth #3: Firefighting by heroes compensates for structural challenges and weak business engagement

At most utilities, the IT people who have been around longest know the most. They probably built the legacy system and have operated it ever since. They know how to make just about anything work, how to resolve technical issues in the most expedient way, and what shortcuts to take to meet aggressive deadlines from business leaders. For these heroic efforts, they are justly recognized across their organizations.

But however effective this firefighting may be, it tends to obscure fundamental issues such as weak technology processes, poorly disciplined change management, the build-up of multiple overlapping technologies (in integration layers, for instance), and the presence of numerous emergency shortcuts (such as direct reads to a core-system database to check for payment status). Over time, issues like these increase the fragility of the legacy system and the cost of replacing it. Often the root cause of such issues lies in a tendency for IT to accommodate every business request that comes its way without highlighting the technical complexities involved or proposing simpler alternatives. Another common cause is cultural shortcomings such as a lack of accountability and a siloed IT organization (Exhibit 3).

Many organizations believe that a major system change is the answer to problems like these. Yet system changes rarely address underlying IT processes or change the middleware that integrates core systems with channels and other systems. Organizations that tackle this challenge successfully tend to take a multipronged approach. They set up teams to upgrade middleware and data-access layers and replace the legacy system with a consistent set of technologies based on the latest thinking about scalable and flexible technology design. They also strengthen key technology processes to ensure that the organization reduces errors in running and operating systems.

Inconvenient truth #4: Running legacy systems and designing new architecture require different skills

While legacy IT talent’s understanding of business processes and the working of the old system will be invaluable in developing the new one, it won’t be enough. That’s because modern approaches to system development, such as distributed databases and concurrent read/writes, aren’t compatible with the best practices these seasoned experts deploy to maintain legacy systems.

So utilities need to find a way to keep their old hands on board while introducing new thinking and call out blind spots and unconscious biases. Bringing in an external hire with recent architecture experience can help strike the right balance between skepticism and a can-do attitude. Another proven approach is to team seasoned IT staff with young hires fresh from digital companies. One utility hired recent graduates and paired them with experienced legacy-system developers who supported them through a year of classroom and on-the-job training. Within six months, the new hires were able to address issues independently; after two years, they worked as developers in their own right. The approach also freed up seasoned experts to help with implementing new systems.

EXHIBIT 3

Cultural issues can impair an organization’s ability to build and operate systems reliably

Quotes

“Some people are insecure or don’t feel they trust their peers.”

Key observations

- Teams look to deflect accountability for system-wide issues
- Unclear ownership of data and middleware

Quotes

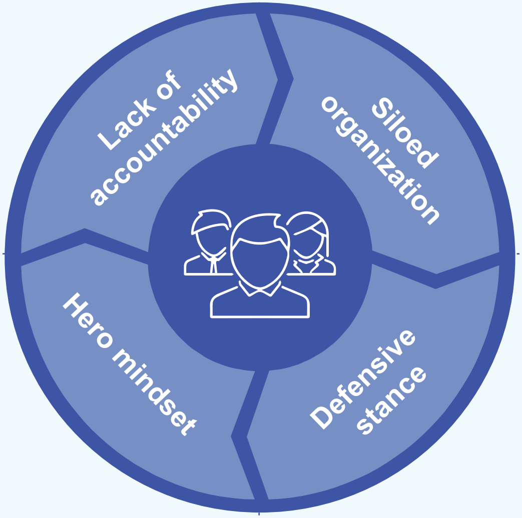
“We find a problem but no one investigates the root cause”

“Once we throw it over the wall it’s their responsibility“

“There are standards at the enterprise level, and the BU architects feel that there are standards for their BU. And sometimes these can be at odds”

Key observations

- Teams look to deflect accountability for system-wide issues
- Unclear ownership of data and middleware



Quotes

“Whenever there is a system outage, everyone jumps on a bridge and we do whatever it takes to solve the problem”

Key observations

- When system outages occur, teams revert to an “all hands on deck” approach
- No proactive steps are taken to prevent system disruptions

Quotes

“My system was fine during the hurricane”

“It gets down to what group you are in, rather than guiding principles”

Key observations

- Owners are concerned about showing that their system didn’t cause failure rather than working together to prevent issues

The culture feedback loop leads to a highly siloed organization that impedes change and slows defect resolution

Inconvenient truth #5: Architecture transformation hasn't yet made it to the top of the CIO's agenda

Architecture is the outcome of the tough—and sometimes irreversible—choices that technologists make in designing a system. It's often criticized as too visionary, out of touch, and difficult to implement or, conversely, as too tactical and focused on current transactional needs. Tight deadlines frequently force project managers to take shortcuts they plan to remediate later but never do. And so the technology "debt" mounts ever higher.

To make matters worse, architecture functions are notoriously understaffed, and report far too low down in the IT organization—typically two or three levels below the CIO. It's hardly surprising, then, that modernization plans are rarely ambitious and seldom make it onto the agenda of the CIO, let alone the executive team or board. But this must change if companies want to move forward. In particular, the CIO needs a detailed understanding of architecture needs and gaps, a ring-fenced budget, and a SWAT-style team to make things happen.

One retail CIO elevated IT architecture to report directly to him. He got enterprise and project architects working together to ensure new systems were properly implemented and set aside an annual budget to fund modernization and pay off the technology debt. Updating IT architecture is like doing preventive maintenance at a power plant: if it doesn't happen, the plant will collapse.

These inconvenient truths should ring bells for most utilities, but they need to be aware that some large retail utilities, especially in Europe, have already recognized the challenges and made great strides in modernizing their technologies. Having done so, many quickly realize that they need to address regulatory constraints that may require an even bigger rethinking of their technology platform.

The digital modernization journey

Having learned from these inconvenient truths, utilities should be ready to plan their digital modernization. The CIO will need to set up a SWAT-style team to drive execution and rapid decision making, with a leader who is a direct report. Another vital step is to enhance governance and discipline around key IT processes, such as change and release management and infrastructure design. Third and last, the CIO will need to develop a business case and roadmap of changes that can be shared with the business. This roadmap is best planned in three phases:

Phase 1: Pilot journeys and 'hot spots' to capture early wins and gain credibility

In this phase, the utility takes one business process—ideally from a customer-facing journey such as payment—and brings in a small "business + IT" product team comprising process owner, designer, IT architect, engineers, and data scientists, supported by experts in agile, cloud, DevOps, and cybersecurity. The team quickly extracts functionality for the process from legacy systems and rewrites it as services on a modern integration layer, creating API end points to the legacy system where applicable.

We've found that many of the near-term issues caused by poor architecture can be addressed by focusing on a handful of "hot spots." Taking this more tactical approach can create early wins and earn the credibility needed to proceed with longer-term initiatives. Some of the hot spots we've seen companies tackle in this way include simplifying multiple middleware systems, creating scalable data stores to improve access to customer and outage data, and testing performance under storm conditions to develop better approaches to system resiliency.

Phase 1 demonstrates the value that can be created at speed through targeted architectural changes. The limited scope of each release means it carries much lower risk and allows utilities to commit funding in much smaller increments of, say, $5 million to $25 million.

Phase 2: Scale up journey implementation and replace selected systems

Once the utility has learned from the pilots and demonstrated early wins, it's time to address journeys that can be modernized without replacing the legacy billing and calculation engine. The first step is to identify elements of the architecture that are fit for purpose or modifiable, and then build on them to deliver customer journeys and functionality in the short to medium term. This effort will also highlight legacy elements that are difficult to modify or need rebuilding from scratch. Targeted changes such as creating a data lake can often unlock value quickly and create capabilities that will still be needed when the new system is implemented.

In parallel, the utility identifies the technology needed to stitch together all the journeys and new modules being delivered. This integration layer may include functionality for microservices and workflow and a business-rules engine. The utility also isolates the core transaction system so that it can, if necessary, be replaced more easily in phase 3. By the end of phase 2, the utility should have a short- to medium-term roadmap that specifies milestones, dependencies, sequencing, and duration for all planned activities.

Phase 3: Extend the transformation to the core transaction architecture

Many customer journeys and processes will by now have been implemented in a separate layer on top of the legacy billing and calculation engine. Sometimes, though, this engine has constraints that can't be addressed during phase 2, such as an inability to support net metering or time-of-use rates. So it's worth examining whether there is still value to be captured by replacing it, and if there is, building the business case.

That involves planning the phasing in of the replacement, taking into account the size of the customer base, the demand for new functionality, the need for customer education, and the complexity of testing and migration. Though still a significant undertaking, this replacement will be far less complex than a big-bang replacement would have been. Much of the remaining legacy back end will already have been decoupled from other systems while APIs were being built and functionality was being extracted into services. The separation between systems allows new customer-facing functionality to be delivered even while the back-end system is being replaced, taking much of the time pressure off the modernization effort.

Key questions to help CIOs get started

Answering these questions will help IT architecture replacements get off to a good start.

- How well defined are the nonfunctional requirements of key business processes?

How do major projects address them?

- How do business requirements drive technological complexity?

- How mature are key IT processes? How do inconsistencies in them contribute to issues with system stability?

- What are the hot spots in the technology landscape, from architecture to infrastructure? How do they affect stability and flexibility? Are they being addressed through major projects?

- What are the gaps between system design and actual usage?

- How does the culture of the IT organization enhance or hinder the modernization and maintenance of a reliable technology architecture?

◆◆◆

Replacing legacy architecture is one of the biggest technology challenges a utility can face. Adopting a modular approach doesn't only help reduce risk and improve returns; it also builds an organization's capability to deliver other systems and positions it to succeed in a digital era. ◆

Eelco de Jong is a partner in McKinsey's Charlotte office; **Anand Mohanrangan** is a senior expert in the Silicon Valley office, where **Aditya Pande** is a partner; and **Parker Shi** is a senior partner in the New Jersey office.