kentoh/Getty Images

Perpetual evolution—the management approach required for digital transformation

Oliver Bossert and Jürgen Laartz

Companies that commit to continually updating their enterprise architectures can deliver goods and services as fast as Internet-born competitors do.

Internet retailers can make crucial changes to their e-commerce websites within hours, while it takes brick-and-mortar retailers three months or more to do the same. Cloud-based enterprise software suppliers can update their products in days or weeks. By contrast, traditional enterprise software companies need months.

Why can't established companies move as quickly as their Internet-born competitors? In part, because they are limited by their enterprise architecture, which is the underlying design and management of the technology platforms and capabilities that support a company's business strategies.

The enterprise architecture in traditional companies typically reflects a bygone era, when it was not necessary for companies to shift their business strategies, release new products and services, and incorporate new business processes at hyperspeed. Consider that until this decade, mobile devices, the Internet of Things, and big data and analytics platforms weren't crucial for competing in the marketplace. Companies did not have an acute need to continually infuse new IT-enabled business capabilities into their operations.

They do now.

To compete against digital-born companies, traditional companies need to adopt a much different approach to designing and managing enterprise architecture—a model we call "perpetual evolution," because it emphasizes continual changes to and modular design of business capabilities as well as the technologies behind them. This approach encompasses a range of widely known enterprise architecture frameworks but links them together in a new way. It compels executives to take a comprehensive view of their digital capabilities and technologies but to manage them in a way that mitigates or removes interdependencies and emphasizes speed. Indeed, our work with companies exploring digital transformations suggests that a shift to the perpetual-evolution model can result in faster product-development cycles and greater operational efficiencies—outcomes that are in sync with customers' expectations.

An enterprise architecture built for perpetual evolution differs from a traditional one in six important ways. When considering business processes and activities, IT and business leaders emphasize end-to-end customer journeys rather than discrete product- or service-oriented processes. They use multiple operating models rather than one. When considering the application landscape, IT leaders design and develop applications to be modular and work independently rather than being tightly coupled with other applications or systems. The enterprise architecture features a central integration platform that boasts lightweight connections rather than a heavyweight bus.[1] The IT organization deploys an application-development model in which developers and IT operations staffers work closely to test and launch new software features quickly (DevOps). And the general view of information and communications technology is as a commodity rather than a strategic factor (Exhibit 1).

In this article, we compare the perpetual-evolution model with existing approaches to designing and managing enterprise architecture, and we explore what's required to shift to this newer approach. The companies that do can unburden themselves of their legacy business processes and mind-sets. They can build the systems and capabilities required to thrive in this era of digitization, enhanced service delivery, and dramatically reduced software-release cycles.

Comparing old, new management approaches

A good way to understand the evolution of enterprise architecture is to consider how companies have traditionally treated its core elements—business operations, business capabilities, the IT-integration platform, IT-infrastructure services, and the underlying information and communications technologies. How would those elements look different under a perpetual-evolution model?

Business operations

Companies have typically designed their business operations using technologies and methodologies with an eye toward simplifying

[1] A connection layer that contains most of the business logic (or rules of computing).

EXHIBIT 1

Digital transformation requires a different model for managing enterprise architecture.

Traditional model	Elements of enterprise architecture	Perpetual evolution model
Focus on product- or service-centered processes	Business operations	Focus on customer-centric journeys
Reliance on one operating model	Business capabilities	Use of multiple operating models (working at 2 speeds)
Emphasis on interdependency	Business applications	Emphasis on decoupling applications
Use of heavyweight bus[1]	IT integration platform	Use of lightweight connections
Software development managed centrally	Infrastructure services	Software developers and IT operations jointly build new products and features (DevOps)
Managed as precious asset	Information and communication technology	Managed as commodity

Technology stack

[1] Connection layer that contains most of the business logic (or rules of computing).

Source: McKinsey analysis

internal processes. They may build systems that automate internal transactions such as “order to cash” and “service inquiry to resolution,” for instance, and only update those systems incrementally.

Under a perpetual-evolution model, business operations and digital systems must be designed with an outward-facing view—that is, focused on the customer experience online and offline. Priorities have changed. The customer used to be an element in a product- or company-centered process; now the products and services are an element in the customer journey. To be sure, companies’ inward-focused view isn’t obsolete. Enterprises need to maintain core transactional processes and systems, whether they are accounts payable and receivable, order management, procurement, or something else. And they must also make sure those business processes and technologies remain efficient. However, businesses’ operations and IT systems must now reflect all phases of and elements within the customer journey—not just the exact moment of purchase. And the experience must be continually updated. Individual companies are becoming part of larger industry ecosystems that are focused on supporting end-to-end customer journeys. In the old world of TV manufacturing, for example, companies designed their business operations and IT systems to follow the product to retailers. Today’s digital TVs have become platforms for manufacturers to provide a range of TV-related services to the home, such as identifying shows consumers might want based on their viewing habits, targeted advertising, and more. As a result, TV manufacturers’ business operations and IT systems must encompass the end user’s TV viewing experience, not just the retailers’ requirements. And because end-user preferences will be ever-changing, business operations and activities must be adapted on the fly.

Note that B2B companies are not immune to this trend, especially those that embed digital technologies into their products to sell predictive maintenance, performance improvement, and other services—for example, construction equipment, aircraft engines, power turbines, and drilling equipment. Companies' enterprise architecture must be able to support customers for the entire time in which they use products and services, even in real time.

Business capabilities

As we mentioned earlier, until this decade, companies have not had an acute need to continually infuse new IT-enabled business capabilities into their operations—for instance, identifying the product a customer is most likely to buy next. Rather, they introduced these capabilities into their enterprise architectures slowly and periodically. Business applications that support these capabilities, such as enterprise-resource-planning (ERP), product-lifecycle-management (PLM), and customer-relationship-management (CRM) systems, were managed as tightly coupled systems; making changes in one often required making big changes in others.

In today's fast-changing digital world, however, companies must be able to continually improve business capabilities without fear of disrupting entire systems. One way to do so is to group processes and systems into two categories: digital business capabilities that are differentiating for the customer experience, and those that support transactional capabilities. We call this a two-speed architecture, and it is a critical element of the perpetual-evolution model because it helps companies direct their resources appropriately.[2]

Consider a retail chain that sells a growing proportion of its products through its website. The company cannot take months to enhance its product-recommendation engine when a digital-born competitor can do that in days or weeks. It must have an architecture that makes business capabilities systems-agnostic. It shouldn't matter, for example, what kind of core systems the retailer has; its new or enhanced product-recommendation approach should be able to be implemented and changed easily. These digital business capabilities become the basis on which to compete in an online world.

IT-integration platform

The first two elements of enterprise architecture we have discussed are focused on front-end operations and activities, whereas the other four involve consideration of the back end of companies' enterprise architectures.

Under a traditional model of enterprise architecture management, companies' IT-integration platform would typically feature a single heavyweight enterprise service bus. This setup can make it difficult for companies to operate digitally in real time. The number of connections increases exponentially in a digital environment, and when all service calls have to pass through the heavyweight bus, the connection layer can become a bottleneck. So companies can have a hard time, for instance, offering website visitors faster page-loading times. Such delays can represent billions of dollars in lost revenue.

The perpetual-evolution model, by contrast, emphasizes lightweight connections to improve

[2] Oliver Bossert, Martin Harrysson, and Roger Roberts, "Organizing for digital acceleration: Making a two-speed IT operating model work," October 2015, McKinsey.com.

transmission performance and address the problem of latency—the time it takes for companies to deliver web pages to online customers who demand instant responses at every click. The functional elements of the purchasing experience, such as payment- or promotion-management applications, can be decoupled from one another—although when a change does not affect a single service but the entire platform it can still be managed on a slower development track. In this way, companies can upgrade core applications within CRM, ERP, PLM, and supply-chain-management systems module by module (or service by service) without having to make whole-system replacements. The application-migration process can happen faster, and any risks—of downtime, for instance, or the introduction of system bugs—can be kept to a minimum.

IT-infrastructure services

In most traditional companies, IT-infrastructure services (the hardware, software, and network resources required to support an enterprise IT environment) are centrally managed by an independent team. After application developers and code testers finish their tasks, they turn over their assignments to a production team, whose complex testing and handover processes could delay the delivery of a new system to the market for weeks or months.

Under a perpetual-evolution model, DevOps becomes central to a company's ability to test new digital business capabilities and bring them to market rapidly. The concept of DevOps has firmly taken hold in many companies. It involves bringing together IT developers with IT-operations staffers to codevelop new software products and features. Because both sides have skin in the game—with no organizational siloes or middlemen between them—they can address problems proactively. Under this approach, companies are seeing increased productivity within their software-development teams, faster release of digital products and services, and improved customer experiences. Our experience suggests, for instance, that companies can reduce the average number of days required to complete code development and move it into live production from 89 days to 15 days, a mere 17 percent of the original time.

Information and communications technology

Information and communications technologies (ICT)—the combination of all the company's audiovisual, telephone, and computing networks—have tended to be costly. Companies deployed them carefully as expensive (but necessary) assets. However, advances in connectivity, cloud computing, and other technologies have made it easier for companies to adopt a perpetual-evolution mind-set and model for managing ICT. They can use cloud-technology services, for example, to turn IT into an affordable resource, regardless of company size. Indeed, even start-up companies can get up to speed in their target markets quickly by renting computing power and storage space from cloud vendors. ICT is now a commodity, and prior investments are no longer necessarily a big competitive advantage or barrier to market entry.

Establishing a perpetual-evolution architecture

Managing changes systematically across all elements of the technology stack will enable companies to move to an architecture of perpetual evolution. Most companies, however, still view each as a separate system or capability rather than as critical interconnected components of architecture. We have found five principles to be critical for changing this mind-set:

- Free up development teams from unnecessary dependencies.
- Be consistent; focus on change across all areas of the enterprise architecture.
- Break down silos …
- … but maintain a strict separation of the platform team from other teams.
- Recognize that transformation of enterprise architecture must be an ongoing process.

Free up development teams from unnecessary dependencies

Companies must be able to change elements of their digital products and processes quickly, thus keeping up with competitors' ability to generate new and innovative customer experiences on demand. To do that, companies must free up their development teams from unnecessary dependencies (Exhibit 2). They can do this by deploying DevOps models and decoupling applications from larger platforms. Teams would no longer have to wait for sign-offs, handoffs, and preparation of test environments when writing code. Those tasks would be managed within the team, with immediate input from development and operations specialists. Such freedom could help development teams reduce their software-release times from months to hours.

Eliminating dependencies is crucial if companies want to design and sell new digital capabilities to ever-more targeted customer segments, each of which will have different needs. Let's use the example of an auto manufacturer that has embedded digital technologies into its cars that enable customers to make online updates to navigation, infotainment, and other systems. To ensure perpetual evolution, the automaker needed to design those systems so it could isolate the business capabilities it wants to offer customers—for example, a certain navigation capability or a specific new feature of the infotainment system—and so it could change or update these elements independent from one another.

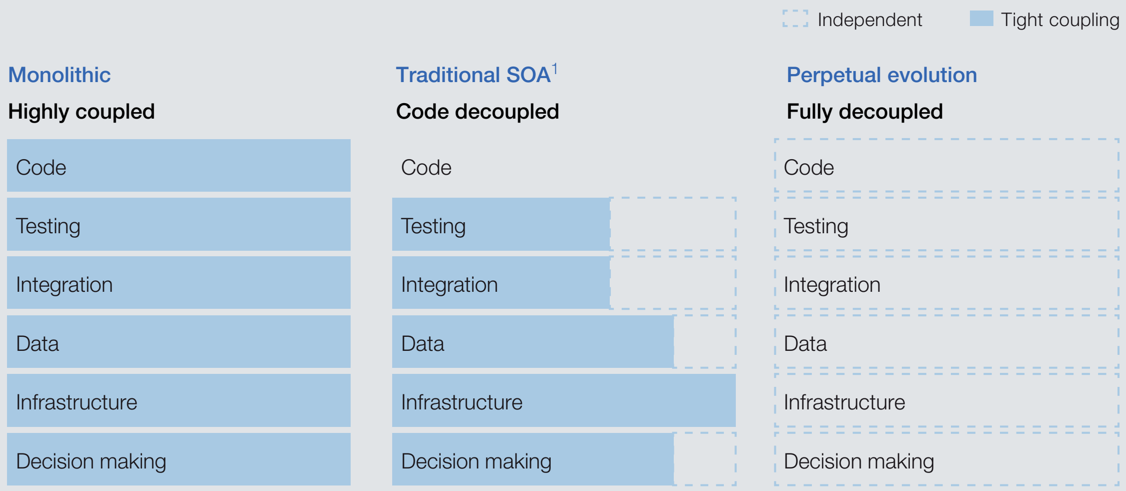
Be consistent; focus on change across all areas of the enterprise architecture

Coding isn't the only place to worry about dependencies. Dependencies also crop up in testing, integration, data, infrastructure, and decision making. By the latter, we mean the individuals who must sign off on the implementation of new business capabilities—is it the team chartered to build and enhance them, or senior management? If, after the capabilities are developed, senior management must approve them before they are put into the marketplace, you can bet it will take those new capabilities a long time to come to market.

Such dependencies are a feature, in effect, of earlier approaches to enterprise architecture. All the elements of the enterprise architecture were tightly coupled. Different modules used the same code base, so a change in one area prompted time-consuming dependency checks to determine how other areas might be affected. The installation of new software depended on the schedules of software testers and resources. Even when developers decoupled software functionality, they often coupled the data, which created dependencies. And when developers intended to decouple the integration layer from applications, teams still too often hardwired

EXHIBIT 2

The perpetual-evolution model eliminates dependencies among elements of the technology stack.



[1] Service-oriented architecture.

Source: McKinsey analysis

business logic into the heavyweight bus, also creating dependencies. When software was ready to move into production, the handover from the development team to the infrastructure team often slowed things down. They were now working on the production team's schedule, competing against a long queue of software releases. Perhaps most important, awaiting senior management's approval for a new software system or functionality upgrade before it went into production could set things back by weeks.

To be sure, companies' movement over the past few decades toward services-oriented architecture (SOA) plus a decoupling of code from the other five elements of the enterprise architecture have been major advancements. Companies can now design web services around specific business capabilities. Yet in most companies, the testing, integration, data, infrastructure, and decision-making activities remain tightly coupled. Companies must explore the use of web services so that new software features can be launched independent of any others, and independent of any piece in the IT stack. In fact, their ultimate goal should be just that, rather than to create a focused service.

Break down silos …

IT architects have often been stereotyped as "people drawing funny boxes in charts." For their part, software developers have been viewed as the people who write code for the modules that those "funny boxes" represent. This division of labor has all too often led to both groups operating in their own worlds rather than working closely together. A company that wants to be digitally competitive will need enterprise architects more than ever. However, those architects can no longer can maintain an arm's-length relationship with developers. They must work closely with them to make sure the architectural rules of perpetual evolution—not just the code—are written into software. Architects need to be part of the teams focused on a business capability or group of related capabilities. They will find themselves working alongside product managers, developers, marketers, testers, production people, legal help, and others.

… but maintain a strict separation of the platform team from other teams

Every company informally manages a part of their IT architecture as a platform, and it organizes other parts according to business capabilities (for example, microservices associated with customer onboarding or marketing campaigns). To shift to a perpetual-evolution architecture, companies must draw explicit boundaries between these two parts of the architecture. Then they must enforce those boundaries through strict oversight and other governance processes.

A company's digital business capabilities enable it to make rapid changes to products and processes, therefore IT professionals must shift their focus along these lines as well. They should define the parts of the IT platform according to the business capabilities they support, rather than as technologies. Defining an IT capability as "service integration" will help the company identify the technologies in the organization with comparable functionality. It will also help the company create more meaningful roles, such as "service integration architect," rather than "XYZ product architect."

Recognize that transformation of enterprise architecture must be an ongoing process

By drawing clear boundaries between business capabilities and technology platforms, companies will be able to isolate the fast-moving parts of their infrastructure (the business capabilities) from the slower-moving ones (the platforms). Nonetheless, they cannot ignore the need to continuously improve their platforms. Companies must make sure they can update pieces of their platforms continuously as well. For many on the senior-leadership team, this will require a significant change in mind-set; traditionally they have been focused on requesting and approving "big bang" system changes. IT leaders and enterprise architects will need to educate the C-suite about the benefits of the perpetual-evolution model, which emphasizes continual monitoring and continual renewal, across all elements of the technology stack. They may need to introduce new forms of reporting and communications, for instance, to help business executives understand the need and to keep track of outcomes.

◆◆◆

To stay competitive in a world in which providing a great customer experience has become paramount, companies in nearly every industry must continually innovate digital products and services, as well as the business processes that support those products and services. They can gain greater agility if they abandon rigid enterprise-architecture-management practices of the past and adopt a new approach that enables perpetual evolution—changing out elements of enterprise architecture quickly, adding new parts in no time, and incorporating the latest and greatest functionality. This shift in methodology can help traditional companies keep pace with digital-born competitors.

Oliver Bossert is a senior expert in McKinsey's Frankfurt office, and **Jürgen Laartz** is a senior partner in McKinsey's Berlin office.

Digital/McKinsey

June 2017

www.McKinsey.com
@DigitalMcKinsey
www.facebook.com/DigitalMcKinsey/