

# Rethinking the technology foundation for digital transformations

The McKinsey Perpetual Evolution™  
approach will help established companies  
keep pace with digital innovation

February 2017

Authored by:  
Oliver Bossert  
Jürgen Laartz

Why are some Internet retailers able to make crucial changes to their ecommerce websites in hours while it takes brick-and-mortar retailers three months or more to do the same? How come just a few car manufacturers can rapidly make online updates to their products in the field, be they to their infotainment systems or to fuel and engine performance—a practice that is becoming crucial in a world of servitization, where manufactured products come with digital services attached to them? And how is the new wave of cloud-based enterprise software vendors able to make software updates to their products in days or weeks, rather than the months it takes traditional enterprise software vendors to do so?

## Rethinking the technology foundation for digital transformations

Given that they compete front and center today with such digital natives, established companies must be able to operate with that kind of speed. But most of them can't, and to a large degree it's because of their enterprise architecture—that is, how they have designed their technology to support their business strategy. To compete against the best digital natives, traditional companies now need to adopt a much different approach to enterprise architecture. We refer to it as Perpetual Evolution™ because it allows them to continually upgrade both their digital business capabilities and the technologies underneath them.

For companies whose enterprise architectures lock them into legacy business processes and technologies, competing against pure digital companies with perpetual evolution architectures should raise red flags. Established companies in nearly every industry are racing to digitize their business models, product and service offerings, and the business processes that support them. Some are spending enormous sums to ward off companies like Amazon, Netflix, Uber, Spotify and PayPal which began life on the Internet and have rapidly gained share in sectors ranging from retail and entertainment to banking and transportation.

Unburdened of having to connect their new digital systems to aging technologies, today's digital natives can build greenfield digital business processes unconstrained from prior work, in online marketing, sales, distribution, and other areas. That is a key reason why they're thriving in this era of digitization, product servitization, and dramatically reduced software release cycles—all at a time in which those three trends thwart the incumbents.

In this article, we explain what an enterprise architecture of Perpetual Evolution™ is, how it contrasts with the architecture approaches of the past, and why it has become necessary. We then explore what companies must do to shift their enterprise architecture from the old to the new and the benefits they can get in doing so. Finally, we discuss what it takes to operate with this new architecture.

## Enterprise architecture approaches, old and new

In the physical architecture, in which everything from homes to skyscrapers are mapped out, building design approaches depend on the technologies and materials available for construction. Skyscrapers, for example, couldn't be built until the arrival of high-strength steel and elevators in the late 1800s. The same principle holds true in the ways companies design products, work processes, and the information systems that support them—the ability to capitalize on new digital technologies requires new approaches to enterprise architecture.

The enterprise architecture approaches in most large companies today reflect a bygone era, a time in which it was not necessary for companies to rapidly shift their strategies, continually evolve their products and services, and constantly improve their business processes. Until this decade, digital technologies such as mobile devices, Internet of Things sensors, and big data and analytics platforms weren't crucial to being able to compete in the marketplace.

A way to think about the enterprise architecture practices of the past, current and future is to look at six core elements of the enterprise stack—business operations, business capabilities, IT integration platform, IT infrastructure services and information and communications technology:

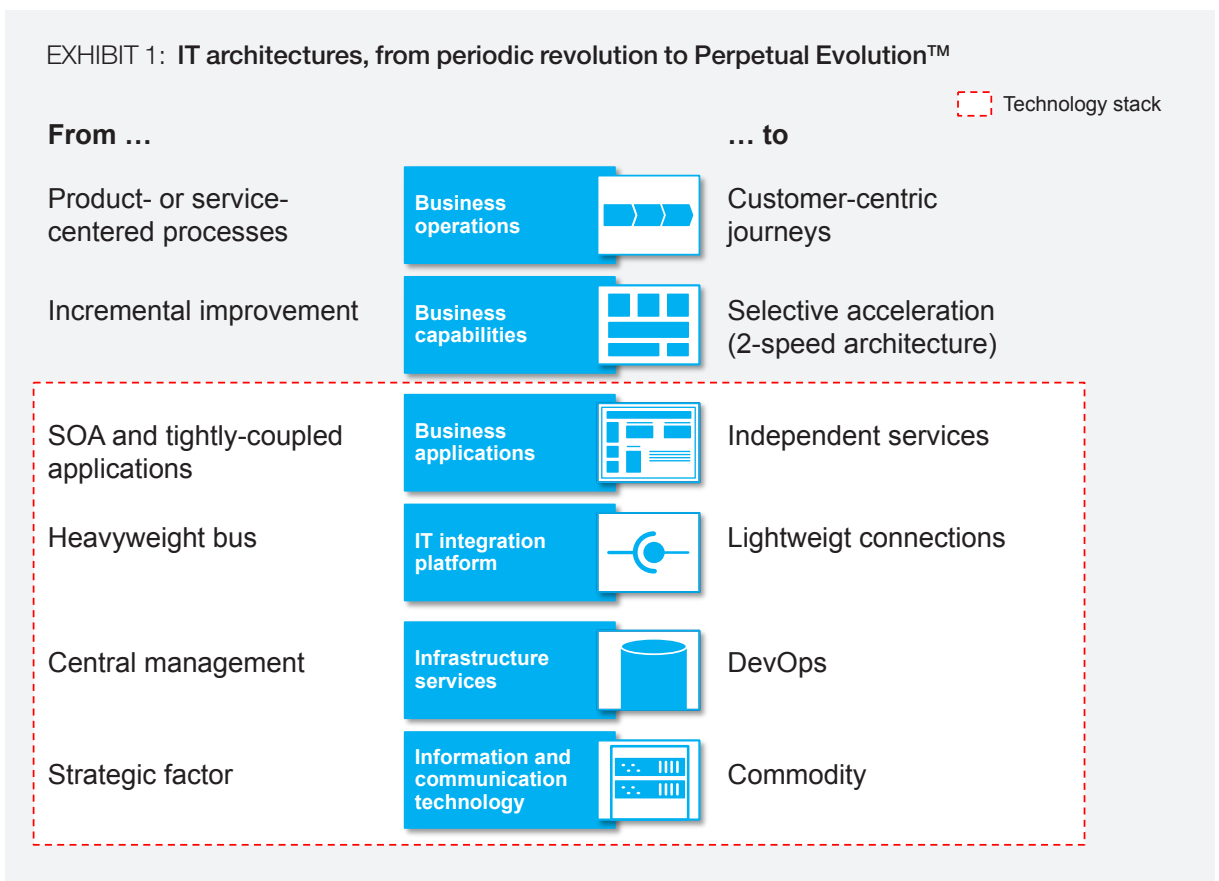
- **In designing their business operations**, companies developed technology and methodology that support an inward-looking view—systems that automated and incrementally improved business processes such as order to cash, and service inquiry to resolution.
- **Companies did not have an acute need to continually infuse new IT-enabled business capabilities into their operations.** (By business capabilities, we mean isolated business activities that accomplish a distinct business outcomes such as an activity within a business process. Finding the product a customer would most likely buy next is one such activity.) They introduced those capabilities slowly and periodically. Even today, most companies have not rebuilt their applications as business capabilities.<sup>1</sup>
- **Business applications** such as ERP, SCM, PLM and CRM were tightly coupled systems. Making changes in one part of the application often required making big changes in other parts.
- **The IT integration platform** was what we call a heavyweight enterprise service bus (ESB)—heavyweight in that most business logic was written into the bus. That has turned ESBs into big monoliths that make it difficult today for companies to operate digitally in real time—e.g., to offer web visitors low latency page loading times. Customers who won't put up with such delays in the online experience are costing online retailers billions of dollars in lost revenue.
- **IT infrastructure services** were centrally managed by an independent team. After the developers and code testers were through with their tasks, they threw it over the wall to production, whose complex test and handover process could delay the delivery of a new system to the market for weeks or months.
- **Information and communications technology** was costly, and thus it had to be deployed carefully as a costly (but necessary) expense that had to be minimized. This is a key reason for the intense pressure to consolidate, outsource and offshore IT to drive costs down.

<sup>1</sup>The Paris Guide to Infrastructure. McKinsey Quarterly.

These architectural practices were successful in a world in which companies from time to time had to make major, multifaceted changes in their business and IT infrastructure. Those changes happened every 10 to 15 years, and they typically were the result of implementing a major enterprise system such as ERP, SCM, PLM and CRM.

In effect, these architectural approaches enabled what we call periodic revolution in the business and IT architecture—“big-bang” changes every decade or so. But operating in today’s digital world requires a much different architecture. The new architecture must enable companies to make continual changes in their business process and IT infrastructure. As such, it is an architectural approach that we refer to as Perpetual Evolution™.

Exactly how is an architecture of Perpetual Evolution™ different? Let’s examine its implications for those six core elements of the stack. (Exhibit 1)



### Business operations: designed with an outward view

Business processes and digital systems must be designed with an outward view focused on the customer experience online and offline—rather than on an inward view of a company’s operations. Priorities have changed. While the customer used to be an element in a product- or company-centered process, today the products and services are an element of a customer journey.

To be sure, companies' inward-focused view isn't obsolete. Enterprises need to keep their core transactional processes and systems, whether they are accounts payable/receivable, order management, procurement or something else. And they must also make sure those business processes and technologies remain efficient.

However, increasingly an architecture that enables companies to perpetually evolve their businesses cannot focus only on the journey that customers use in purchasing their offerings. For many companies, architecture needs to reflect a larger journey. Take, for example, an automobile insurance company's architecture. More and more policies are purchased on websites that aggregate the entire auto purchase: the car, the insurance, and more.

These larger ecosystems are end-to-end customer journeys, and you see them happening in many B2C industries in which consumers must buy several products and services to accomplish a single task: buying or renting homes, taking a vacation, and many more.

Note that B2B companies are not immune to the trend, especially those that embed digital technologies into their products (e.g., construction equipment, aircraft engines, power turbines, drilling equipment) to sell predictive maintenance, performance improvement and other services. That's the whole servitization trend we mentioned earlier. That requires a company's enterprise architecture to encompass the entire time in which customers use their products, even in real time, not just the moment of purchase.

Even B2C companies are getting into the servitization game. Take the case of manufacturers of high-definition smart televisions. In the old world of TV manufacturing, companies designed their IT systems to follow the product to the customer (in this case, to a retailer). 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. The impact on TV makers' enterprise architecture is that they must design systems that encompass the end customer's TV viewing experience.

### **Business capabilities: the foundation of digital competition**

Digital systems that provide the narrow functionality (such as one-stop checkout) necessary for a superior customer journey are the foundation of an architecture of perpetual evolution. In fact, the best digital native companies organize their teams of business and technology professionals by such distinct business capabilities. For example, Spotify groups teams into squads, each of which controls a set of business capabilities.

To compete in a fast-changing digital world, established companies must first move to a model for digital capabilities. They must be able to continually improve the digital business capabilities that comprise their customer journeys, and add new capabilities when necessary, by adopting an architecture of Perpetual Evolution™.

That, in turn, means grouping processes and systems into two categories: digital business capabilities that are differentiating for the customer experience, and those that are supporting transactional capabilities. We call this a two-speed architecture.<sup>2</sup> These digital business capabilities become the basis on which to compete in an online world.

<sup>2</sup>Organizing for digital acceleration: Making a two-speed IT operating model work. June 2015. McKinsey.com



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 native online retailer can do that in days or weeks. The only way to do that is having an architecture that makes businesses capabilities systems-agnostic. It doesn't matter, for example, what kind of core systems the retailer has; its new or enhanced product recommendation approach can be implemented and changed easily.

So now after moving to a capability-based model in the first two levels of business operations and business capabilities, the enterprise architects need to ingrain a highly modular software skeleton in the four levels below them in the stack. We start with the business applications level.

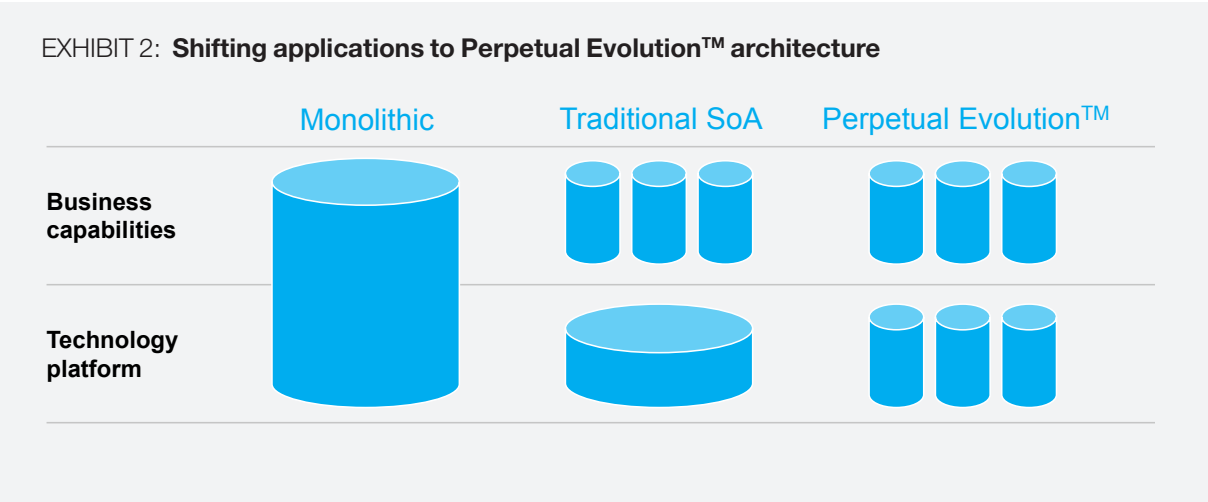
**Business applications: shifting from SOA and tightly coupled software to independent services**

The goal in adopting a Perpetual Evolution™ architecture in this layer of the stack is to be able to upgrade core applications such as ERP, SCM, PLM and CRM module by module (or service by service)—without having to upgrade to a whole new version of those applications. (See Exhibit 2.) This can be explained based on three steps:

**Step 1:** In a monolithic architecture all elements are tightly coupled. The core platform and services are in a single release container.

**Step 2:** In a more service-oriented approach the functional elements, such as payment or promotion management are decoupled from one another. Still when a change does not affect a single service but the platform it will go in a longer release window.

**Step 3:** Perpetual Evolution™ is introducing true modularity of the underlying platform, such as deployment, data management, and analytics, so that also technical capabilities can be changed independent from one another.



**IT integration platform: lightweight integration**

Heavyweight buses such as an ESB no longer are the answer to all problems. Companies need lightweight connections between their services in areas where granularity and performance are of high importance. That's the only way to deal with the problem of latency—the time it takes for companies to deliver web pages to online customers who demand instant responses at every click.

And it is the architectural prerequisite for decoupling services without making the bus a bottleneck. Without lightweight buses calling an increasing amount of Web services, IT integration platforms become big bottlenecks in the customer experience, and a major factor behind online customer defection.

### **Infrastructure services: DevOps takes down the wall**

The concept of DevOps has firmly taken hold in many companies—grouping together IT professionals previously separated in the functions of software development and IT operations. DevOps becomes central to a firm's ability to test new digital business capabilities and bring them to market rapidly. In doing so, companies should be able to tweak business capabilities and put them online in hours or days, rather than the weeks or months it takes to develop an enhancement, throw it over the wall to testing, and then put it into a long production queue.

However, to make the DevOps model work companies need to put their DevOps team members into the teams charged with updating and building new digital business capabilities. In other words, there is one team for each business capability, and on that team are representatives from systems development and operations.

### **Information and communications technology: a big expense becomes a less-expensive commodity**

Technology services from cloud vendors turns IT into an affordable resource for companies of many sizes. All of a sudden, the big, costly IT infrastructure that companies have taken years to build is no longer a competitive barrier. Startup companies can climb into the game quickly, as firms like Netflix (renting computer power from Amazon Web Services for many years to support its streaming video services) and many other digital natives have done. Information and communications technology is now a commodity, and prior investments are no longer necessarily a big competitive advantage and barrier.

Binding together these six fundamental changes in the stack will enable companies to move to an architecture of perpetual evolution of their business and IT. That will enable them to loosen up their asset base of business and system capabilities—they can change out parts quickly, add new parts in no time, and replace other parts with the latest and greatest functionality.

## **Establishing a perpetual evolution architecture**

So how do companies transition their architecture from the old to the new? How do they move from the design philosophy that guided their present IT asset base, to an approach that governs the infrastructure they need now—one by which they can make perpetual evolutions of their business in a digital world?

The obstacles are many, but all are surmountable. From our experience helping companies across industries and geographies adopt a Perpetual Evolution™ architecture, we have found five moves to be critical:

- Freeing up the team from unnecessary dependencies
- Not overinvesting in one area by consistently eliminating dependencies
- Setting boundaries in technology—not on paper
- Maintaining freedom within capability teams through strict separation from the platform
- Regarding the platform as one of continuous change

## Free up the team from having to deal with unnecessary dependencies

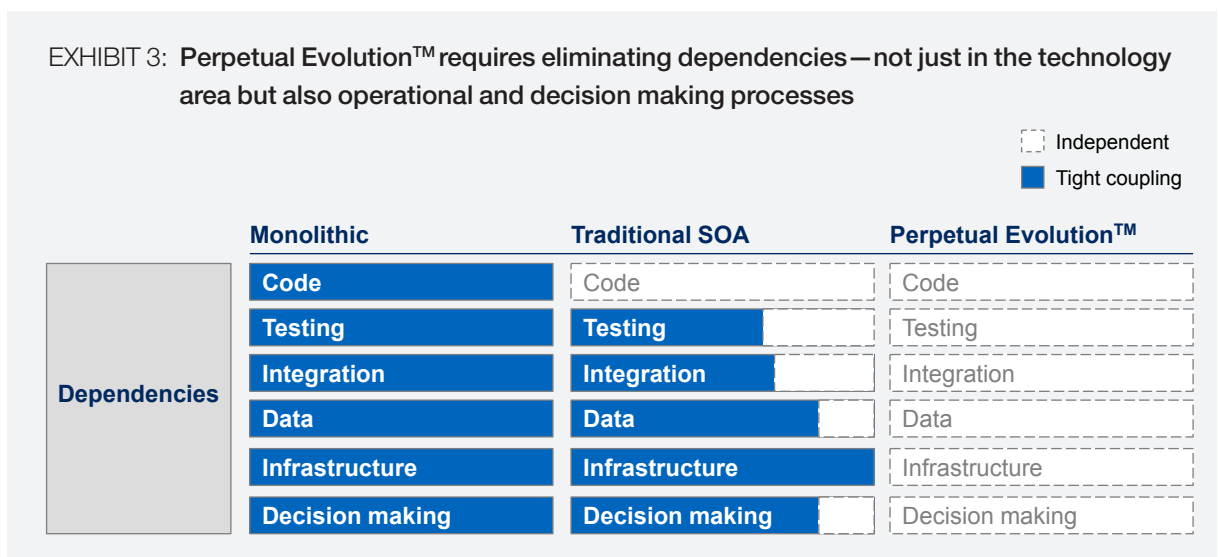
The concept of decoupling is a very old one in technology. However, in the past it was only applied to a limited area of the technology stack.

In the past when talking about decoupling, companies usually defined it as a technical challenge of the services or elements running on top of a rather large platform.

Today, all dependencies must be eliminated because it is the only way a company can change the pieces of its digital products and processes quickly, and thus keep its customer experience competitive. To do that, an architecture must eliminate dependencies among business capabilities and within the technology stack.

Consider an auto manufacturer that has embedded digital technologies into its cars that enable customers to make online updates to its navigation, infotainment and other systems. If the automaker didn't design those systems in ways that it could isolate business capabilities it wants to offer customers—e.g., a certain navigation capability or a specific new feature of the infotainment system—it won't be able to change or update these elements independent from one another. Eliminating such dependencies is crucial to companies that want to design and sell new digital capabilities to small customer segments, each of which have different needs.

With this said, it is clear that dependencies need to be eliminated along all six dimensions of system development, not just in coding. (See Exhibit 3.) In a Perpetual Evolution™ architecture however the modularity and the decoupling is not limited to the functionalities. It also extends to the technology platform and the operating model.



## Don't overinvest in one area: eliminate all dependencies

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 who must sign off on the implementation of new business capabilities—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.



These kinds of dependencies, across all six aspects of a new piece of software, have slowed companies down for years. Such dependencies are a feature, in effect, of earlier approaches to enterprise architecture. They are what forced big technology changes to be made only periodically.

In monolithic architectures, all six elements of the software stack were tightly coupled. The code base of different modules was tightly coupled, requiring time-consuming dependency checks. Installing new software depended on the schedules of software testers and resource. Even when developers decoupled software functionality, they often coupled in data. That created dependencies. And when they intended to decouple the integration layer from applications, teams still too often hardwired business logic into the ESB, also creating dependencies. When software was ready to put 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 importantly, awaiting senior management approval for a new software system or functionality upgrade before it went into production could set things back by weeks. This is the decision-making dependency.

To be sure, the last decade's movement toward services-oriented architectures (SOA) was a big advance: decoupling code from the other five elements. It helped companies design web services around specific business capabilities. Yet in most companies we know, the testing, integration, data, infrastructure and decision-making activities remained tightly coupled. That was why a new web service could take weeks or months before a company's top management approved it, IT production tested it, and rolled it into the field.

In creating Web services or what is also called microservices, companies must make sure each service is 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.

### **Set boundaries in technology (not on paper)**

IT architects have often been stereotyped as “people drawing funny boxes in charts.” For their part, software developers have been viewed as the people to write for the code for the modules that those “funny boxes” represent. Unfortunately, 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, they 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.

That means the 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.

### **Make sure capability teams have freedom (by strict separation from the platform)**

The teams working on specific business capabilities must not have to worry about whether their improvements affect their company's underlying technology platform.

Every company has a part of their IT architecture that they manage as a platform, and (increasingly) other parts that are organized by business capabilities (e.g., web or microservices). To shift to a perpetual evolution architecture, they must draw explicit boundaries between these two parts of the asset base. Then they must enforce those boundaries through strict oversight and other governance processes.

Because a company's digital business capabilities are what enable it to make rapid changes in digitally-based products and business processes, we believe IT professionals must shift their focus along these lines as well. To start, that would mean defining the parts of the IT platform in terms of the business capabilities they support (e.g., customer management), rather than as technologies. For example, defining an IT capability as "service integration" will help a company identify the technologies in the organization with comparable functionality. But it will do more—help the company create more meaningful roles such as service integration architect, rather than integration product XYZ architect.

### **Regard the platform as one that will continuously change**

By separating the 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 can't ignore the need to continuously improve their platforms. Companies must make sure they can update pieces of their platforms continuously as well.

Crucial advances in platforms can provide key performance improvements. Capitalizing on them requires a platform whose parts are modular, decoupled and organized for reuse.

And only when the pieces of that platform are defined as business capabilities will companies be able to quickly plug in improvements to them, or whole new capabilities.



In nearly every industry, companies have found they need continuous innovation in their digital products and services, and the business processes that support them. Those that can achieve this will stay competitive in a world in which providing a great customer experience has become paramount.

But to do so, companies must abandon their enterprise architecture practices of the past and adopt a new one that enables perpetual evolution in their digital business processes. Those that do this well will take a big step forward in keeping pace with those born-digital companies that have been attacking their businesses.

Oliver Bossert ([oliver\\_bossert@mckinsey.com](mailto:oliver_bossert@mckinsey.com)) is a senior expert in McKinsey's Frankfurt office. Jürgen Laartz ([jürgen\\_laartz@mckinsey.com](mailto:jürgen_laartz@mckinsey.com)) is a senior partner in McKinsey's Berlin office.