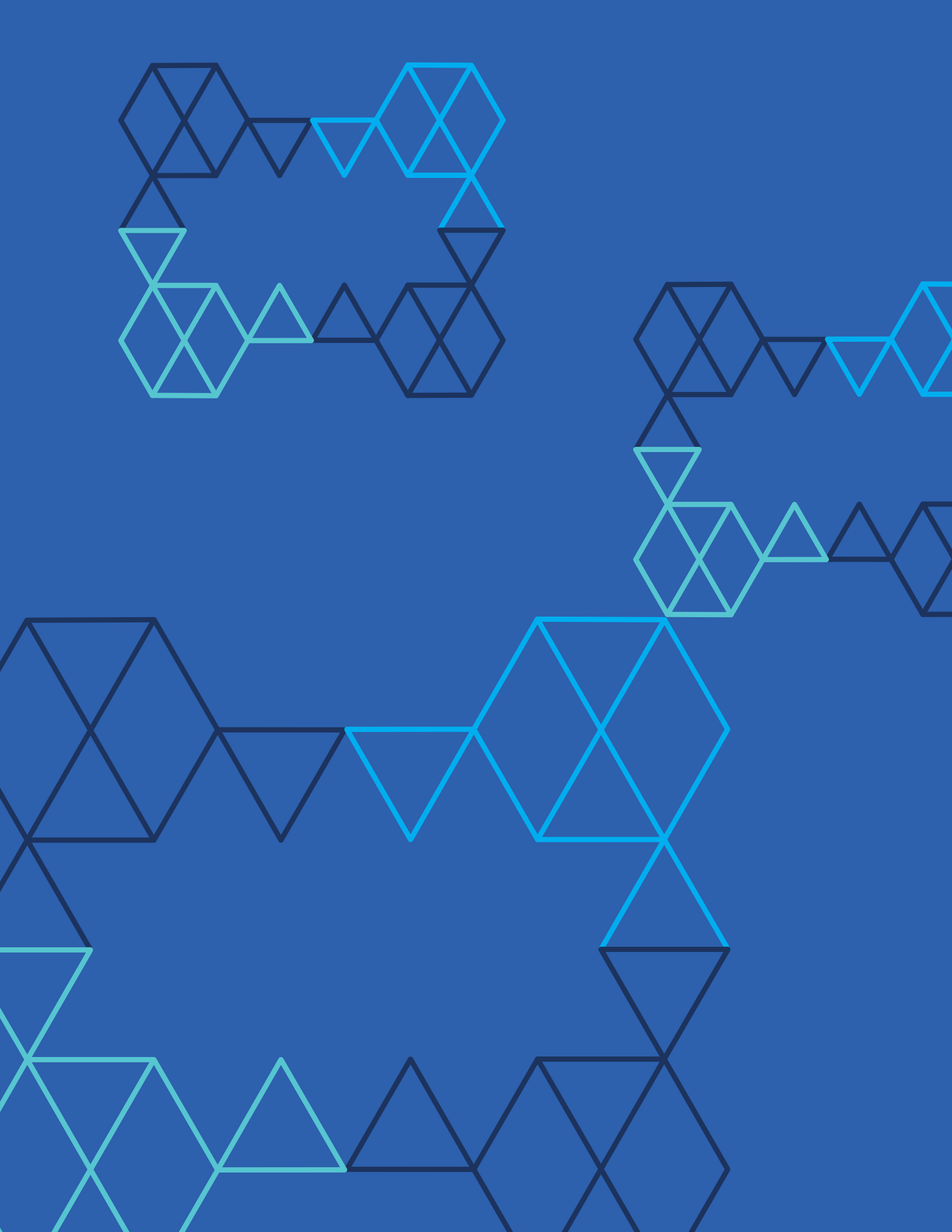
Agile Compendium

August 2016

Digital/McKinsey



Agile Compendium
Contents

Introduction

The agile approach to innovation and software development gained popularity in the technology sector around the turn of the century and has since been adopted by a wide range of industries. As the name makes clear, agile emphasizes speed and flexibility, along with a strong customer focus. Getting it right reduces time to market, increases customer satisfaction, and delivers many efficiency improvements.

Yet, too many companies still get it wrong.

At its core, the agile development approach values individual initiative rather than rigid processes and tools. Eschewing extensive documentation in favor of getting software to work, collaborating closely with customers, and responding quickly to changing market conditions are also hallmarks of the system. It aims to eliminate unnecessary work, while focusing on speed and satisfaction.

Experience in companies show that successful agile programs can increase productivity per developer by 27%, reduce launch delays by 30% and decrease defects in production by a factor of 3x when compared to non-agile projects.

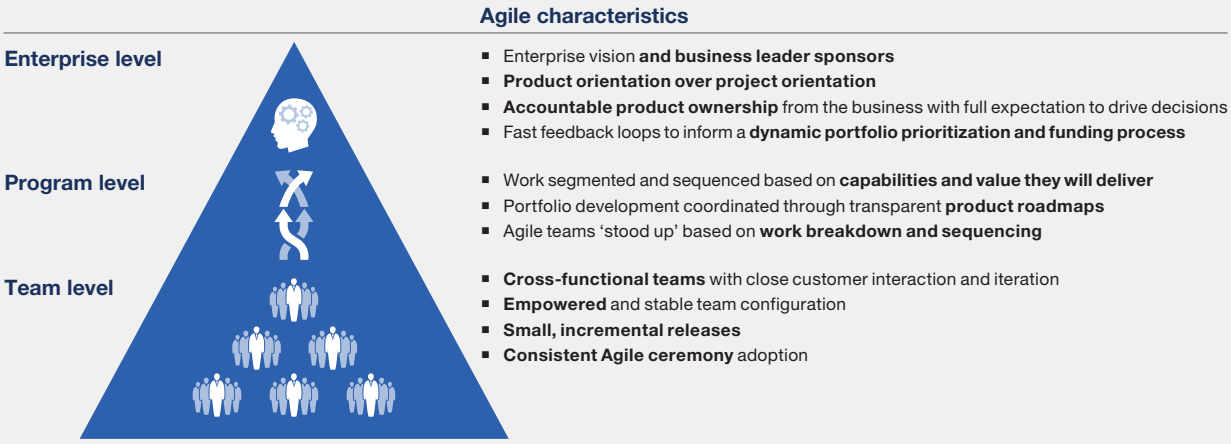
As digital technology seeps into every industry, agile has become relevant to a wider range of companies, from banks creating mobile payment systems to drug makers crunching massive amounts of data from clinical trials. Unfortunately, successful implementation also varies widely. Indeed, a recent survey of ~20 companies trying implement agile-at-scale showed that only 15 percent of the companies in the sample reached all their goals in implementing an agile system.

In this collection of articles, we bring together some of McKinsey's recent insights on how companies in all industries can successfully implement an agile approach. As part of our collection, we'll look closer at companies that have captured the promise of agile and distill best practices from these case studies and our experience helping many companies through the transformation.

While every company will face unique challenges in initiating an agile program, we have found that the most successful implement changes across three levels: the enterprise level, the program level, and the team level (Exhibit 1). Each is discussed in detail in this collection.

At the **enterprise level**, four operational themes stand out. First, successful companies generally redesign their organizational structures to give product development priority. They also create

Exhibit 1 **The three levels of successful agile implementation**



closer links between business units and IT organizations, assuring that each understands the needs and constraints of the other. They redefine roles and responsibilities to assure alignment with agile principles. And finally, they adjust their planning and budgeting systems to allow the flexibility needed to capture the full benefits of an agile approach.

Also at the enterprise level, many companies find it necessary to operate what we see as a two-speed IT architecture. Often legacy systems will run in tandem with newer systems based on digital technologies and founded on the agile approach.

Adopting an agile approach will also require adjustments at the **program level**. A crucial aspect in bringing products out faster is breaking down silos. For example, too often IT infrastructure and IT development teams work in isolation, while working more closely in what's known as a DevOps model can lead to greater automation, self-service tools, and other efficiencies. Freed of silos, developers can concentrate of improving a product rather than creating an individual module or component without context.

And finally, at the **team level** successful companies deploy available capabilities where they are needed and when they are needed, using a flexible structure and fluid assignments. Cross-functional teams can be assembled quickly as needed. In addition, they develop a clear agile playbook, implement agile ceremonies effectively, and manage talent carefully.

Beyond the examples in the individual article, this collection also includes detailed case studies from the public and financial sectors, illustrating the potential offered by the lean approach and best practices in implementing a transformation.

An agile approach to innovation and product development has helped many companies generate greater efficiencies and stay ahead of the competition in rapidly changing markets. More can reap the benefits of this approach with a clearer understanding of the challenges that await and their solutions.

Editors:
Sriram Chandrasekaran
Santiago Comella-Dorda
Niv Goldman
Krish Krishnakanthan
Gerard Speksnijder
Manu Tandon

Enterprise level

An operating model for company-wide agile development

Organizations are succeeding with agile software and product development in discrete projects and teams. To do so in multiple business units and product groups, they must rethink foundational processes, structures, and relationships.

Santiago Comella-Dorda, Swati Lohiya, and Gerard Speksnijder

Many digital companies are using agile development practices to deliver goods and services to customers more efficiently and with greater reliability. Using this software-development approach across all business units and product groups, digital giants have been able to design and build features quickly, test them with customers, and refine and refresh them in rapid iterations.

By contrast, few traditional companies—those with both online and offline presences—are using agile methodologies across the majority of their productand application-development teams. Many banks, for instance, have established digital units to develop and release mobile apps or website features quickly. But those groups typically remain physically and strategically disconnected from the rest of the IT organization and the rest of the company.

Research indicates that many traditional companies are experimenting with agile practices in discrete pilot projects and realizing modest benefits from them. But fewer than 20 percent consider themselves "mature adopters," with widespread acceptance and use of agile across business units.[1] Meanwhile, according to our own observations, the companies that *are* deploying agile at scale have accelerated their innovation by up to 80 percent.

There are many reasons traditional companies have not been able to successfully scale up their agile programs, but we believe a chief impediment is their existing operating models and organizational

Takeaways

Although traditional companies are experimenting with agile, their efforts lack the organization-wide reach necessary to capture the full potential of the approach.

We found that the companies making strides in their agile development practices have focused on four parts of their operating models: they've made organizational structures more product oriented, stepped up business–IT interactions, recast roles and responsibilities, and taken a new look at budgeting and planning.

Several methods, including big-bang redesigns and "wave and spike" models, can help companies transform to keep pace with new entrants, technologies, and customer expectations.

structures. In most of these companies, the process of software or product development remains fragmented and complex: a business request for a new website feature can kick-start a development process involving multiple teams, each tackling a series of tasks that feed into the original request. For instance, one team working on the front-end application, another updating associated servers and databases, and still another reconciling the front-end application with legacy back-end systems. What's more, the supporting business processes (among them, budgeting, planning, and outsourcing) and existing roles and responsibilities in both the IT organization and business units continue to adhere closely to the legacy waterfall approach.[2]

For most companies, it will be difficult to incorporate agile practices from small-scale pilots into all business units and functions—regardless of the success of those pilots—without making significant structural changes.

We have helped many organizations adopt agile development practices in their IT and business groups. Building on that base, we recently studied in depth 13 large traditional organizations that are implementing agile methodologies across functions and business units (see sidebar, "Launching agile at scale: The research base"). To facilitate widespread adoption, these companies have made changes in one or more parts of their operating models, targeting the following four areas: modifying their organizational structures to be more product oriented, improving interactions between the business and IT, redefining roles within business units and the IT organization, and reconsidering their budgeting and planning models (exhibit).

Launching agile at scale: The research base

To better understand the impediments to deploying agile software development at scale, we conducted an in-depth study of 13 organizations that are in the process of extending their agile capabilities. These included companies in financial services, healthcare, telecommunications, and several other industries. The bulk of the companies represented in our research are in North America (six), but we did study companies in Europe (three), Latin America (two), and Africa and Asia (two). All were at different points in their adoption of agile at scale, with some of the most advanced organizations having deployed agile across 60 percent or more of their innovation activities. Through our research, we learned that without making significant shifts in organizational structures, roles and responsibilities, and other underlying elements of the operating model, it can be quite difficult for companies to extend agile practices beyond pilot teams.

Exhibit 1 **To deploy agile development at scale, companies will need to alter their operating models and organizational structures.**

	Before	After
Organizational structure	**Application-oriented focus**, with ever-changing teams and pooled resources; siloed perspective	**Product-based focus** with stable teams and dedicated resources; end-to-end perspective
Interactions between business and IT	Development process is managed by **proxy product owner** from IT, with input as needed from business	Development process is managed by **strong product owner** from business, who works closely with IT at all stages
Roles and responsibilities	Scrum teams comprise **developers and testers**; project-manager and line-manager roles remain unchanged from waterfall approach	**All roles are integrated** within self-organizing scrum teams; project-manager role is minimized and line managers focus on capability building
Budgeting and planning	Traditional **yearly budgeting**, with fixed budget allocated to projects	**Venture-capital-style budgeting**, where minimally viable product is launched and future funding depends on product performance

Source: McKinsey analysis

The companies that have started on this path to change are realizing early benefits. One has switched from a project- to a product-oriented operating model. It has deployed talent and IT resources based on IT requirements for the entire customeronboarding experience, for instance, rather than according to individual applications used during onboarding. As a result of this change in focus, it is now launching up to four website features a month instead of the typical four a year the company was able to release previously. This successful shift to agile was made more attainable when the company carefully considered when and how to phase in various modifications to its operating model.

Scaling agile practices

The benefits of agile are by now well known. Under agile development methodologies, IT organizations and product developers cocreate products and services with the business, rather than simply collecting feature specifications and throwing them back over the wall, as would happen under the waterfall development model. Teams can experiment with minimally viable products, test and learn from those prototypes, and ultimately deliver new software features and products in days or weeks, not years. Based on our observations of leading-edge adopters, quick codevelopment of products and collaboration among highly skilled IT and business professionals can happen on a broader

scale when companies take steps to remake their operating models and organizational structures, focusing in particular on these four principles.

Adopt a product-oriented organizational structure

Traditional companies tend to organize their IT resources according to applications and projects—creating the type of fragmented development experiences described earlier. Instead, they need to organize IT resources around products, gathering business-unit leaders, developers, and other members of the organization in stable end-to-end teams that are focused on delivering designated business outcomes. Such a structure would mean the end of projects as they are traditionally defined and of coordination bodies such as the projectmanagement office.

In an agile-at-scale environment, products can't be defined solely as commercial offerings. They may actually be combinations of offerings (for instance, a payroll service), or the customer experience (say, all the features and tasks that make up the online purchasing journey), or an IT system shared by multiple product teams (such as pricing software that generates quotes on demand). So it's important for business and IT leaders to redefine the units of delivery. And once products have been recategorized, the company must designate an agile team, or clusters of agile teams, that will be responsible for the development and maintenance tasks associated with those products. These teams typically will include developers, testers, product owners, and others. They can draw additional support from a centralized group of experts—specialists in security issues, user-experience researchers, or enterprise IT architects, for instance.

A large medical-device manufacturer significantly shortened its time to market by refining its organizational structure. Under its traditional structure, there could be as many as 20 handoffs when a business unit shared its specifications and requirements with the technology organization for a new piece of software or an additional feature in existing software. Because of the interdependencies among its products, leadership knew it wouldn't be enough to deploy agile within one business unit or within certain product-management teams in the technology organization. In 2015, the company tweaked its product-ownership model so that software requirements were directly transmitted from dedicated product owners in the business units to the agile teams, rather than passing through multiple parties. With this change, the company was able to reduce the amount of time it took to release products in the market. The structural changes also facilitated the rise of several communities of practice. These role-based or topic-based groups (sometimes called guilds) are critical in agile-at-scale environments. They can encourage the transfer of knowledge among team members, promote coordination between teams and functions, and become the catalyst for continuous performance improvement.

Improve interactions between the business and IT

To create an agile-at-scale environment, companies will need to break down silos between and within the business units and the IT organization. It's a perennial issue in most companies. But closer collaboration can be achieved by designating strong product owners from the business units to work with IT—individuals who understand the company's products well and who have the technical knowledge and authority to prioritize feature changes in products.

In most traditional companies, product owners from the business side are involved in software development sporadically, providing input only as needed. To compensate for this lack of engagement, IT organizations often appoint a proxy product owner from IT. This arrangement can be useful in the near term but impede long-term product or project success. The proxy product owner typically has limited access to customers due to organizational barriers and possesses no mandate or

the authority to make decisions. Because direction, priorities, and accountability are lacking, agile development is stalled. Teams face a significant amount of rework and waste.

By contrast, a strong product owner has an in-depth understanding of the product in question, connections to and an understanding of customers, and full authority to make quick decisions. Such accelerated decision making helps to reduce bottlenecks in development and increase productivity.

A provider of software-as-a-service solutions was struggling to get products to market in a timely fashion. There were marked lags in decision making and unclear lines of communication between IT and the business. In 2014, the company implemented a three-tiered product-owner structure, with a chief product owner leading a product domain, a senior product owner leading a product line, and product owners working with the scrum teams. Under this revised structure, interactions between IT and the business units improved. The lines of communications were clearer. The company was able to make decisions much more quickly while maintaining consistency and coordination within and across productdevelopment groups. In part because of this structural change, the company was able to bring new software products to market quarterly— and in some instances monthly—rather than only once or twice a year.

Redefine managerial roles and responsibilities

About half the companies we studied have redefined managers' roles and responsibilities to account for the distinct capabilities associated with agile versus waterfall development. Consider the differences: the project manager working under a waterfall approach typically needs to coordinate a range of tasks occurring across applicationdevelopment teams, database teams, and so on. Under an agile approach, however, the number of tasks (and therefore the need for coordination) is minimized. The tasks that remain are handled by a strong product owner or the agile team itself. Similarly, the process-management tasks that were traditionally done by line managers—for instance, identifying and addressing dependencies and assigning tasks to individuals—are handled by selforganizing, product-focused agile teams.

A large bank in Africa redefined certain roles, shifting the lines of communication and responsibilities, to accommodate the bank's desire to deploy agile practices more widely. Previously, software-development teams worked with various technology leads to translate architects' requirements into technical specifications. Under an agile approach, however, this translation step was no longer needed. The bank eliminated the tech-lead role within agile teams. Developers are now empowered to talk directly to architects and product owners, so they gain a better understanding of customers' needs and can develop software to accommodate those needs. Line managers will, of course, continue to play central roles—providing career-development support and serving as subject-matter experts within agile teams and formally transferring their knowledge to others. But their responsibilities were redrawn, and this was communicated widely so that team members knew what to expect and whom to contact in particular situations.

Indeed, the companies we've seen that have effectively implemented agile at scale are resolutely transparent—they provide clear guidelines about which decisions should be made within the team and which require external input. The boundaries are clearly defined; team members are empowered enough to be accountable but not so much that they could create major risks with rogue or carte-blanche actions.

Reconsider budgeting and planning models

IT organizations typically adhere to annual budgeting and planning cycles—which can involve painful rebalancing exercises across an entire portfolio of technology initiatives, as well as a sizable

amount of rework and waste. This approach is anathema to companies that are seeking to deploy agile at scale. Some businesses in our research base are taking a different approach. Overall budgeting is still done yearly, but road maps and plans are revisited quarterly or monthly, and projects are reprioritized continually.

A large European insurance provider restructured its budgeting processes so that each product domain is assigned a share of the annual budget, to be utilized by chief product owners. (Part of the budget is also reserved for requisite maintenance costs.) Budget responsibilities have been divided into three categories: a development council consisting of business and IT managers meets monthly to make go/ no-go decisions on initiatives. Chief product owners are charged with the tactical allocation of funds—making quick decisions in the case of a new business opportunity, for instance—and they meet continually to rebalance allocations. Meanwhile, product owners are responsible for ensuring execution of software-development tasks within 40-hour work windows and for managing maintenance tasks and backlogs; these, too, are reviewed on a rolling basis. As a result of this shift in approach, the company has increased its budgeting flexibility and significantly improved market response times.

A handful of companies are even exploring a venture-capital-style budgeting model. Initial funding is provided for minimally viable products (MVPs), which can be released quickly, refined according to customer feedback, and relaunched in the marketplace—the hallmarks of agile development. And subsequent funding is based on how those MVPs perform in the market.

Under this model, companies can reduce the risk that a project will fail, since MVPs are continually monitored and development tasks reprioritized. Typically there is less waste and more transparency among portfolio and product managers, and it becomes easier for the company to scrap lowpotential projects early.

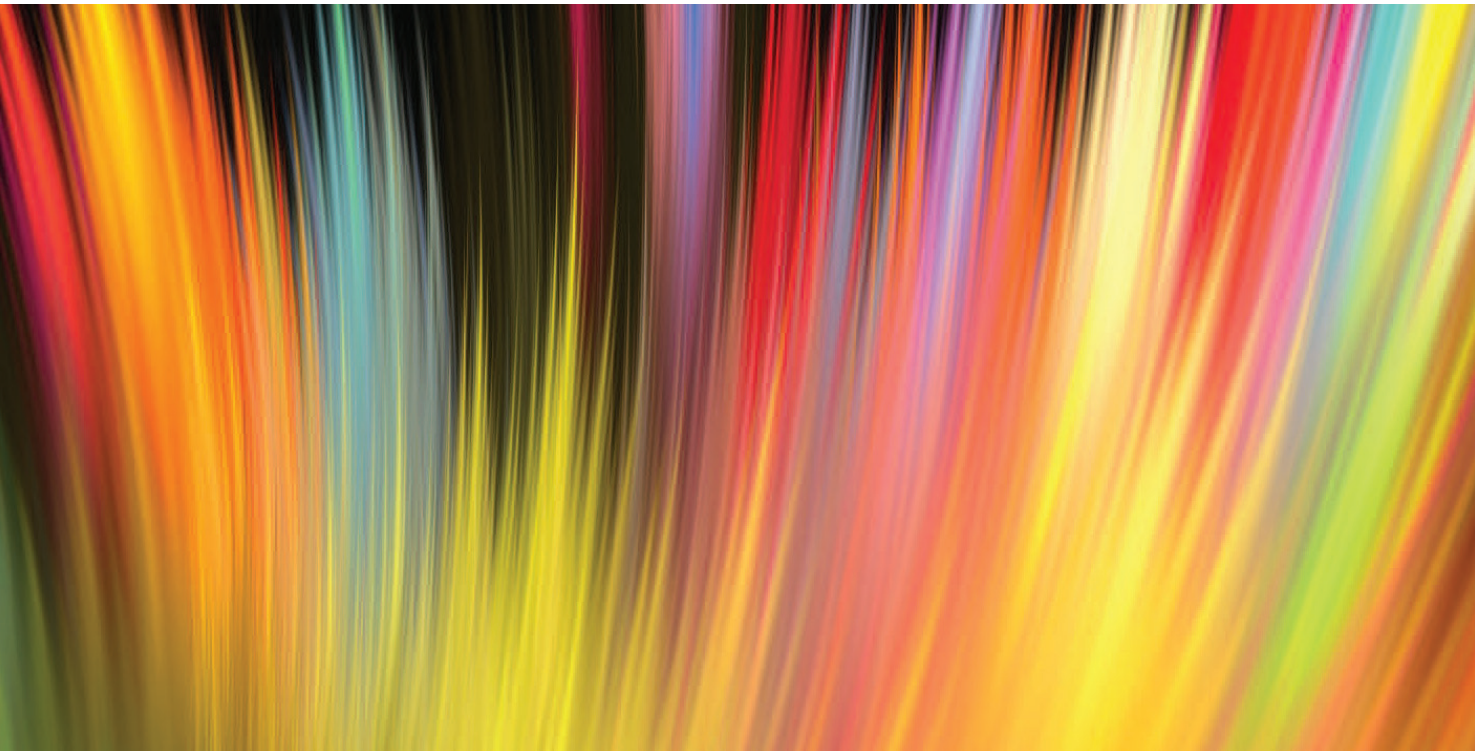
Choosing the right approach

Revamping an operating model is a large undertaking. There will be significant risks to address and short-term disruptions as new ways of working take hold. As with any large change-management initiative, such a transformation will require long-term commitments from employees at all levels, in all functions and business units. The companies we've studied have used a number of approaches to alter elements of their operating models.

At one extreme, some have used the "lab approach," in which an agile operating model is set up apart from the rest of the organization to serve as a testing ground before capabilities and processes are rolled out to the entire IT organization. This approach makes most sense when the company has only limited support from senior management for larger changes and needs to prove the business case quickly. For the most part, however, the separate organizations created under the lab approach tend to remain separate rather than influencing change across the organization.

At the other extreme, a handful of companies have embarked on a "big-bang redesign," in which they move all functions and business units toward new organizational structures and roles, self-contained agile cells, and faster processes—all in one go. For this to work, senior leadership must be all in from day one, which is likely to be the case in only a small subset of companies.

Somewhere in the middle is the "wave and spike" approach to deploying agile at scale. Under this model, individual teams are reconfigured as agile teams in waves, while elements of a new operating model are deployed in spikes. A large technologysolutions provider, for instance, needed to ramp up its digital capabilities fast. The company's IT organization was struggling to get products to market given the increasing size, complexity, and sheer number of projects. The company transitioned product-development teams



to agile practices in waves; 5 were included in the first training and deployment cycle, while close to 20 were part of the second. As each successive wave of teams was indoctrinated to agile, feedback was collected and training materials were developed or revised for the next set of teams. Agile coaches were also installed to guide teams.

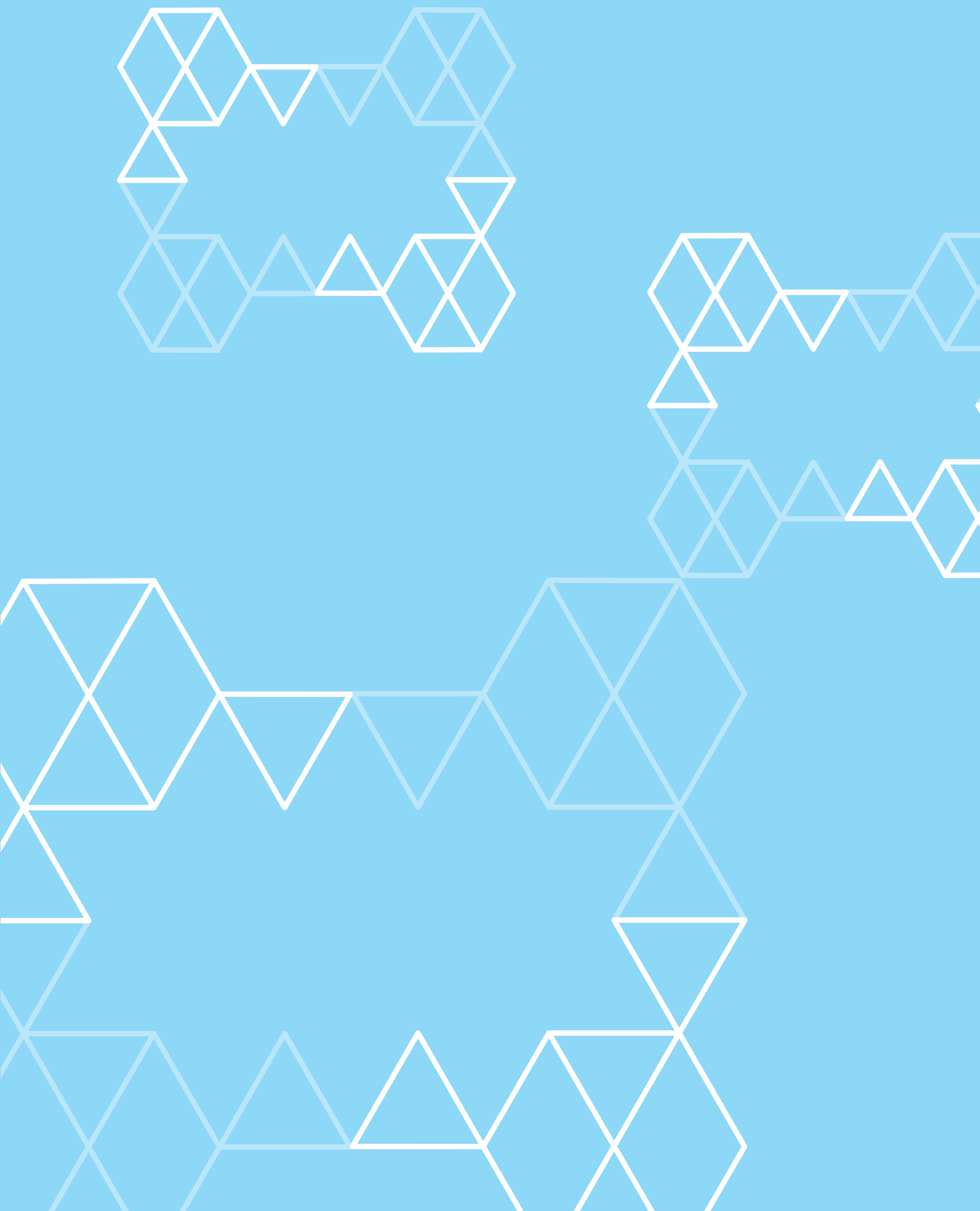
Six months into its agile transformation, the company adopted a product-oriented organizational structure, gathering business-unit leaders, developers, engineers, and members of the IT organization into "tribes." Many months after that, the company focused on a different spike—interaction between IT and the business. It adjusted its operating model so the product-development group could collaborate more closely with the IT operations group (in a true DevOps model). As a result of these changes, time to market accelerated dramatically; because teams were cocreating products, the number of defects and the rework required decreased.

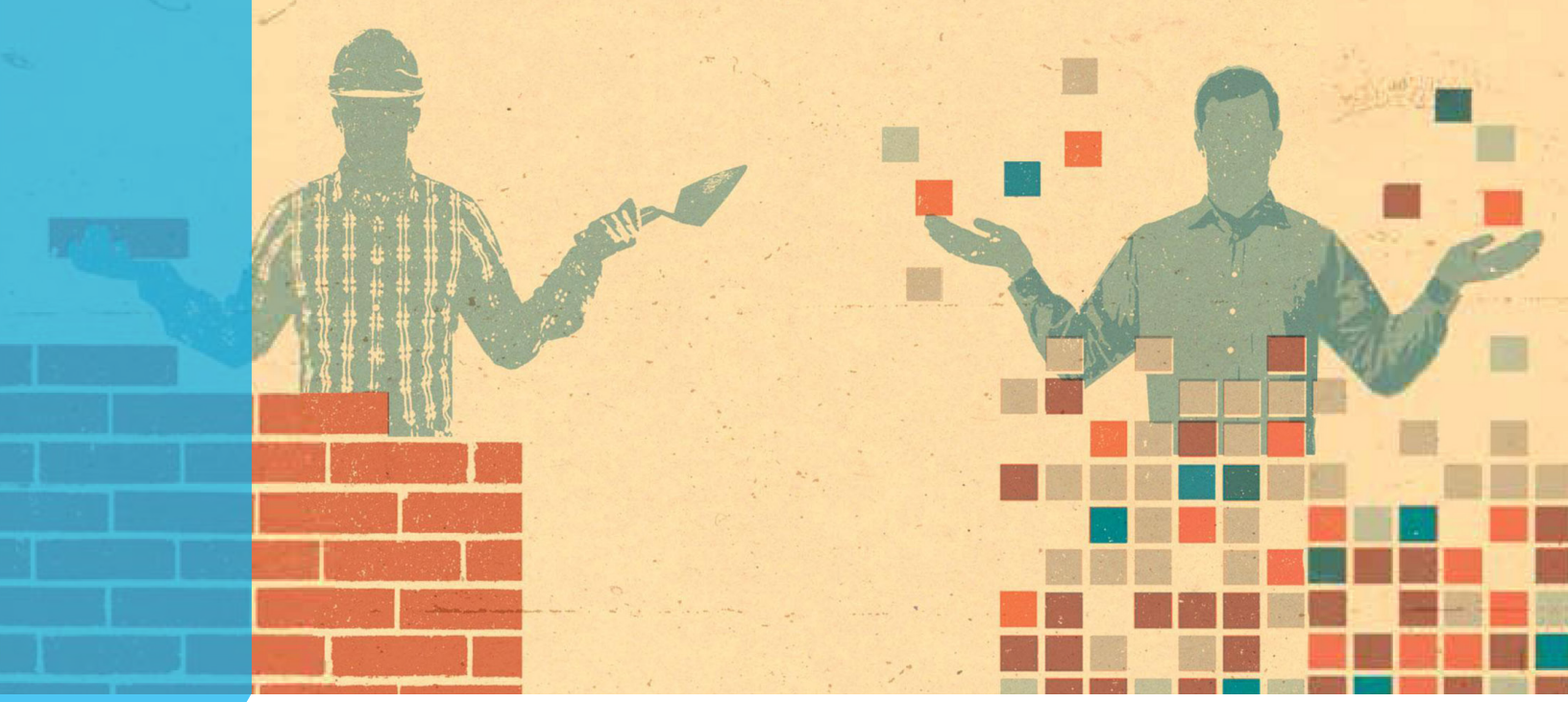
■ ■ ■

Companies that are finding small-scale success with agile development practices may be loath to mess with a good thing, figuring it best to avoid the risks that widespread adoption might present. One of the chief risks in a digital business world, however, is standing still. To keep pace with new market 8 entrants, emerging technologies, and changing customer expectations, companies will need to find ways to extend their capabilities in agile software development to all functions and business units. They must be willing to adapt the very fabric of their organizations and give agile methodologies the space and support they need to thrive. ■

[1] "Tenth annual State of Agile survey," VersionOne, 2016, versionone.com.
[2] Waterfall product development is asynchronous; teams walk through multiple process steps, requiring sign-off on each task before they can start on the next one.

Santiago Comella-Dorda is a principal in McKinsey's Boston office, **Swati Lohiya** is an expert in the London office, and **Gerard Speksnijder** is a master expert in the San Francisco office.





A two-speed IT architecture for the digital enterprise

Delivering an enriched customer experience requires a new digital architecture running alongside legacy systems.

Oliver Bossert, Chris Ip, and Jürgen Laartz

Digital business models have become essential for companies across a range of industries. With social networks and e-commerce websites setting new benchmarks for speed, agility, and user-friendliness, consumers expect similar online performance from banks, retailers, and telecommunications companies. Attackers born in the digital age give consumers what they want, but many older companies struggle to meet customer expectations. For them, going digital is now a prerequisite for surviving and thriving. Success requires strong capabilities in four areas.

First, because the digital business model allows the creation—and shorter time to market—of digital products and services, companies need to become skilled at digital product innovation that meets changing customer expectations. One such new offering for consumers is carinsurance policies enabled by geolocation tracking technology, where the price of the policy depends on how much and how aggressively a person actually drives.

Second, companies need to provide a seamless multichannel (digital *and* physical) experience so consumers can move effortlessly from one channel to another. For example, many shoppers use smartphones to reserve a product online and pick it up in a store.

Third, companies should use big data and advanced analytics to better understand customer behavior. For example, gaining insight into customers' buying habits— with their consent, of course—can lead to improved customer experience and increased sales through more effective cross-selling.

Takeaways

Attackers born in the digital age have no trouble giving consumers what they want, but many older companies struggle to meet customer expectations. For them, going digital is now necessary for survival.

To succeed, companies must ensure the digital readiness of their products, enhance their multichannel customer experience, take advantage of advanced analytics, and automate their operations.

This usually requires that they develop a two-speed IT architecture—creating a new high-speed system alongside their legacy IT system. Building a new organization and governance model in parallel will help support the program.

Fourth, companies need to improve their capabilities in automating operations and digitizing business processes. This is important because it enables quicker response times to customers while cutting operating waste and costs.

A two-speed IT architecture will help companies develop their customer-facing capabilities at high speed while decoupling legacy systems for which release cycles of new functionality stay at a slower pace.

Implications for enterprise architecture

Each of the four levers poses a substantial challenge for IT. For example, many bankingproduct lines—among them credit cards, investments, and checking and savings accounts—are managed in silos. This makes it difficult to quickly get a comprehensive view of customers, for example, to assess their loan applications. What's more, channels are often managed and tracked independently, complicating matters for customers who wish to use multiple channels as they pursue a transaction. For instance, customers starting a loan application on their smartphone may find that they have to reenter data when they switch to a desktop computer to fill in the more detailed information required. Weak systems integration and slow database-access times can prevent customers from enjoying a real-time shopping and purchasing experience. Analytics capabilities are especially difficult to integrate with operational process flows. Manual steps in these processes, such as rekeying and transferring information, present major obstacles to both analytics and automation of processes.

While a few players have overcome some of these hurdles, it is a big challenge for many IT executives to implement all four levers so customers can, for instance, purchase individually tailored products across multiple channels. One important reason is that their legacy IT architecture and organization, for example, which runs the supply-chain and operations systems responsible for executing online product orders, lacks the speed and flexibility needed in the digital marketplace.

Indeed, the ability to offer new products on a timely basis has become an important competitive factor; this might require weekly software releases for an e-commerce platform. That kind of speed can only be achieved with an inherently error-prone software-development approach of testing, failing, learning, adapting, and iterating rapidly. It's hard to imagine that experimental approach applied to legacy systems. Nor would it be appropriate, because the demand for perfection is far higher in key back-end legacy systems. Quality, measured by the number of IT system errors, and resilience, measured by the availability and stability of IT infrastructure services, comes at slow speed but is critical for risk- and regulatory-compliance management and for core transactional activities such as finance and online sales. In contrast, lower IT system quality and resilience can be acceptable in customer-facing areas, for instance, when users participate in the testing of new software. For these reasons, many companies need IT architecture that can operate at different speeds.

The building blocks of digital-enterprise architecture

In our experience, digital-enterprise architecture needs to accommodate the following elements to deliver the functionality that the digital enterprise requires

Two-speed architecture. This implies a fast-speed, customer-centric front end running alongside a slow-speed, transactionfocused legacy back end. For software-release cycles and deployment mechanisms, the customer-facing part should be modular, to enable quick deployment of new software by avoiding time-consuming integration work. In contrast, the transactional core systems of record must be designed for stability and high-quality data management, which leads to longer release cycles.

Instant cross-channel deployment of functionality. New microservices defining only a small amount of functionality, such as lookup of the next product a consumer would most likely purchase, should be deployable in an hour rather than in several weeks. Such microservices should also be available across all channels. Ideally, it should be possible to develop these services in multiple programming languages rather than being locked into a single development framework.

Zero downtime. In digital global operations, days-long maintenance windows are no longer an option. Upgrades of systems affecting the consumer's experience should be seamless, using a concept that allows the deployment of a new software or service in parallel with the old version. First, only about 1 percent of the user traffic is routed to the new version. Only when the new version fulfills a set of key performance indicators will all traffic be routed to the new version. Moreover, in daily operations, there should be fallback mechanisms in place so that issues arising in one service do not harm overall operations more than necessary. If, for instance, a retailer's personalized recommendation service is unavailable, a random recommendation in a relevant category would be displayed rather than an annoying web error page

Real-time data analytics. Customers generate data with every move they make within an app. The ability to analyze that information in real time can make analytics an integral part of operational processes and not just a stand-alone capability. For example, one retailer analyzes customers' purchases automatically when they pay with their credit card; along with the receipt, the business provides a savings coupon for a product the customer will likely be interested in buying the next time he or she shops at the store.

Easy process configuration. Business users themselves should be able to change automated processes. This would allow them, for example,

Traditional companies don't have the luxury of starting with a clean slate . . . the transformation toward digital is a continuous process of delivering new functionality.

to eliminate unnecessary process steps without requiring timeconsuming coding by an IT developer.

Product factory. Industries that provide digital products, such as banking and telecommunications, need to decouple the products from the processes. A bank, for example, would implement one sales process and reuse it for all products, such as accounts and cards

Automated scaling of IT platforms. In a digital business, workloads expand and become harder to predict. Ideally, this load would be balanced across private- and public-cloud environments, with mechanisms in place to ensure that when one provider has an outage, others can take over the workload.

Secure architecture. In a digital business model, cybersecurity must be an integral part of the overall application. Not only does the company have more valuable data to protect, making it more attractive to hackers, but the digital strategy also opens new interfaces to customers, suppliers, and partners that can be exploited by hackers.

Moving toward two-speed architecture

Unlike enterprises that are born digital, traditional companies don't have the luxury of starting with a clean slate; they must build an architecture designed for the digital enterprise on a legacy foundation. What's more, while most companies would have been comfortable in the past going through a three- to five-year transformation and not implementing new features in the meantime, today's highly competitive markets no longer allow players to alter architecture and business models sequentially. It is therefore important to realize that the transformation toward digital is a continuous process of delivering new functionality (see sidebar, "A retailer begins the two-speed journey"). Successful digital transformations focus on the following aspects

Manage a hybrid target architecture with very different platforms. Digital target architectures are heterogeneous, with transactional platforms managed for scalability and resilience coexisting alongside other systems optimized for customer experience. The transformation can only be sustained if a high-level target architecture and standards in critical areas such as cybersecurity are clearly described from the beginning. Without them, the transformation can be slowed down by the complexity of legacy and new hardware and application provisioning.

Plan for continuous software delivery with blends of methodologies. There isn't time to develop software by using a waterfall model and then separating the transformation into several long phases, as in traditional multi-year IT transformations. Nor is the solution to migrate all delivery to agile methodologies.

The answer is to do both but blend the benefits of agile (iterative development, continuous delivery) into the waterfall model. Now, the software solution for each business challenge has to be continuously developed, tested, and implemented in an integrated fashion. This requires clear segregation of platforms into domains managed for fast iterative delivery (for example, for customer-experience applications) or for transactional integrity (for back-end transactional systems).

A retailer begins the two-speed journey

One retailer learned the hard way why a new IT architecture is required in the digital world. The company tried to launch an e-commerce business as a new unit separated from the traditional brick-and-mortar operation. While the time to market for this offering was short, it was limited to the online channel and could not offer the user experience that many customers expected.

It quickly became clear that the retailer could only compete in the digital marketplace if it could provide a sophisticated multichannel experience that allowed customers functionality such as reserving goods online and picking them up in the store. To implement this broader offering, IT leaders had to overcome challenges in both the new fast-speed IT architecture and the legacy transactional systems.

While continuing to develop the fast-speed architecture of the original online offering, new development frameworks and processes were required to speed up the deployment of new software and integrate it with third parties, such as software-as-a-service providers. What's more, the entire IT organization had to adopt agile application-development methods, and the business organization had to get much more involved in the transformation, particularly regarding the budgeting process and the approval of new projects.

Adapting or replacing some of the transactional legacy systems so that they could support a multichannel offering involved its own challenges. Some of the IT systems were outdated and developed in programming languages that are no longer common among young software developers. In addition, the legacy systems could only offer inventory information via a batch interface, while customers of the online offering require the information in real time. Moreover, some of the legacy systems were costly to maintain.

Balancing the transformation of the transactional architecture with the development of the new fast-speed architecture is one of the main challenges the retailer is facing. Also, both the IT and business sides understand that, unlike most traditional IT projects, building two-speed architecture is a continuous-improvement program that has to deliver new functionality at early stages of the transformation and will continue to run on an ongoing basis.

Develop the low-speed architecture, too. It's important to establish a clear distinction between the two IT models from the beginning and not only focus on the fast-speed part but also develop the transactional back-end architecture. Those systems of record require rigorous development and testing methodologies and must be managed for resilience and scalability, with no compromises.

Build a new organization and governance model in parallel with the new technology. In the digital enterprise, business and IT work together in a new and integrated way, where boundaries between the two start to blur. This partnership has to be established during the transformation.[1]

Change mind-sets. By transforming the architecture, technology can become a key factor for a company's competitiveness. Such a development requires increased management attention and usually a place on the board agenda. While IT efficiency clearly remains important, spending

levels may well rise as companies transform IT from largely being a necessary expense to being a true business enabler. As such, expenses are managed as investments rather than just costs; this will often require a substantial mind-set shift for the organization.

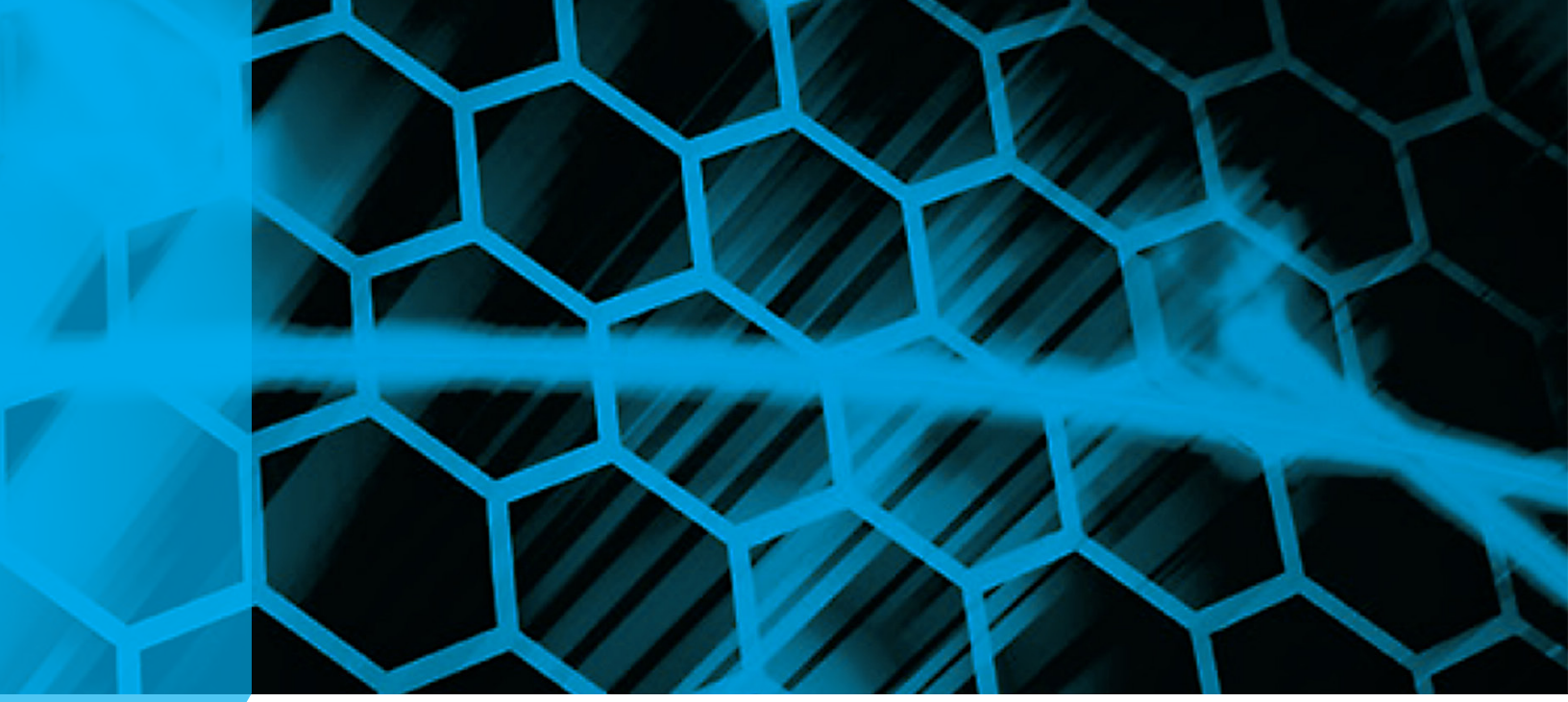
Run waves of change in three parallel streams. In a two-speed transformation, it makes sense to have an implementation plan that runs in three parallel streams. The *digital-transformation stream* builds new functionality for the business, supported by the results of a *short-term optimization stream* that develops solutions that might not always be compliant with the target architecture (for example, using noncompliant interfaces). To ease the development of short-term measures and create a sustainable IT infrastructure, an *architecture-transformation stream* is the third necessary component.

■ ■ ■

For most traditional consumer-facing companies, building a new digital IT architecture that runs alongside legacy systems makes it possible to compete with digital natives. ■

Oliver Bossert (Oliver_Bossert@McKinsey.com) is a senior expert in McKinsey's Frankfurt office, **Chris Ip** (Chris_Ip@McKinsey.com) is a director in the Singapore office, and **Jürgen Laartz** (Juergen_Laartz@McKinsey .com) is a director in the Berlin office.

Program level



Integrated expertise: Models to break siloed software development

The tradition of completely separate organizational functions is incompatible with effective software development. Understanding the options for functional integration and embedding Knowledge across units can help deliver substantial value for software organizations.

Santiago Comella – Dorda, Chandra Gnanasambandam Bhavik Shah Tobias Stralin

Software organizations have historically been divided into functional groups with developers, quality assurance (QA), user experience (OX), security, analytics, and operations sitting in distinct functions. With the exception of a growing number of cases in which QA teams are more broadly integrated, the software organization's functions are walled off from each other and operate independently. As software makes the transition from an on-premise model to the cloud, however, this siloed way of working is being challenged.

Cloud's software disruption

The shift to cloud is driving three distinct software development changes. First, vendor operating models are changing. Operational responsibility now lies with vendors. This places greater importance on reliability, uptime, and the operational effects of the software architecture and the resulting application. Efficiency, scalability, and performance of the application are also becoming more important because more resources are required to support the application. Operational responsibility means a greater security burden, making vendors more active in safeguarding customer data. The shift to cloud also gives vendors access to new capabilities —including advanced analytics, A/B testing, fine-grained customer segmentation, and continuous deployment— making it easier for them to handle the demands of their new responsibilities.

The scalability and reliability requirements of cloud applications have also forced vendors to dramatically improve how they modularize, deploy, and monitor the application. "Hot deployment," horizontal scalability, and transaction monitoring are fundamental vendor capabilities, requiring a diverse set of functions (development, release management, performance engineering, CJX, security, for example) to collaborate effectively.

Second, cloud enables iterative and agile development methodologies along with continuous deployment. These methodologies have shortened the development cycle and the expected time to market.

Third is the fact that performance differences have narrowed across emerging platforms. This convergence has led to the increased importance of design and UX as differentiators. The focus on user experience can no longer be just about placing the button at the right location on the Web page or selecting an appropriate icon. There has to be end-to-end consideration that ensures the application experience is both intuitive and innovative. Together, these cloud driven trends are making the case for sharing expertise across functions and integrating it into development teams.

The traditional siloed software development approach not only slowed down the overall development cycle, but it also produced low quality products. Integrating functional expertise into development teams can facilitate major improvements in the development of both cloud based and on-premise software. McKinsey's software productivity benchmarking data shows that a collaborative, cross-functional approach reduces schedule slip by go percent, cuts down residual defects by 70 percent, and improves overall productivity by 27 percent.

Archetypes of functional integration

The move to cloud is making old ways of working obsolete, and software organizations are responding. They are beginning to integrate the knowledge of the QA, CJX, security, analytics, and operations groups into development teams. Integrating these different skill sets and perspectives is giving software organizations the added power they need to perform in a rapidly evolving software environment. In a survey of software organizations, McKinsey identified four, distinct organizational approaches to inte grated expertise.

Fully embedded resources. With this approach, individuals with expertise in various functions become wholly a part of the development team, and those experts are part of the standard team configuration. They report up through R&D, and priorities are set entirely by the R&D teams. This model is best suited for teams or business units that operate mostly independently and when consistency across teams is not required. It has the potential to create the greatest agility within a development team. It also gives the experts the best opportunity to understand the development team's function and become familiar With the application. UX and operations tend to be the teams that most commonly fully embed their resources into development teams. The trade-off of this decentralized model is that it creates quality inconsistencies and design fragmentation across product groups. To mitigate this, companies that adopt this model create "communities of practice" across embedded teams to share best practices and align on approaches up front. Also, this model tends to lack the structure to support the growth of functional experts within their areas, limiting their professional development opportunities.

Semi-embedded resources. In this model, experts also work full-time within a development team, but they report through a separate organizational hierarchy. Experts across teams collaborate and establish and drive new standards for the organization. This model is as applicable for the LIX and operations teams as the fully embedded model, and companies actually find this one more scalable. The software organizations that participated in the McKinsey survey reported that this model was suited for the security and data analytics roles too. The benefits of knowledge absorption, in general, apply to this model as well. The central governance aspect of this model also promotes a level of consistency. Development teams with integrated operations expertise—known as DevOps—enable experts to better optimize performance or respond more effectively to an incident with the knowledge of the application. Integrating UX expertise can result in a consistent experience for users across all product groups. This is especially important if products across the organization need to work together seamlessly and be integrated. This approach also enables consistent training and skills development across the organization. Some

R&D teams, however, have reported that some applications present a rather steep learning curve for those joining the product team. For LIX and security experts, for example, the idea of "just-in- time" resource sharing may be impractical.

Centralized expertise. In this model, the functional expertise for development teams comes from a centralized external team that provides core capabilities to all development teams. Typically, the central team is comprised of senior-level functional experts who create guidelines and standards for all teams to follow. This type of model works very well for complex topics like security. The capacity for application teams to develop products informed by experts in other functions is built slowly. This model is less radical than the first two in that it doesn't break down the siloes. It is, however, a suitable option for integrating specific expertise not required at scale. It is equally suitable for small and large organizations.

Developer-owned/rotational expertise. This model requires development teams to dig in and take ownership of areas that lie outside of their expertise. Without expert guidance from the embedded models or consultation from the centralized model, engineers are expected to fulfill the cross-functional duties themselves. This model requires development team members to take on various expert roles for a predefined period of time, then rotate responsibility. Given the scaling difficulty, this model may be most appropriate for integrating operations expertise into the development units within start-up or start-up-like environments. It encourages the dissemination of knowledge across the organization, but it can create the most fragmentation. Teams also lose the benefit of having dedicated expertise, since developers are regularly out of practice between their rotations. Smaller companies reported that the pain and benefit of this model was that developers quickly understood the difficulties other functions faced and worked with them to identify appropriate solutions either by changing how they designed and coded applications or collaborating better.

Implementation and impact

The degree to which individual engineers develop deep functional knowledge varies by archetype. Regardless of the model chosen, however, there is a shift in the expectation of all software engineers. It is important that all developers possess security knowledge, for example, and apply its principles when architecting and designing products. Organizations report that implementing one of these models to integrate security expertise into the development unit has promoted a "culture of security."

Software organizations looking to integrate functional expertise into their development units don't have to select just one of the archetypes described above. Integrating one type of expertise into a development team may require one approach, while another expertise may be best suited to a different model. Each function can operate in a different archetype, depending on the function's capabilities, scale, and role within the broader organization's priorities. Organizations have the option of implementing one or a combination of these archetypes (for example, UX can be fully embedded, while analytics is semi-embedded). While organizations have implementation flexibility, McKinsey's survey found that the "fully embedded resources" and "semi-embedded resources" models are the most common for the integration of all four functional roles – UX, security, operations, and QA.

Successfully integrating functional expertise into development teams can deliver substantial value to a company. Looking at UX as an example, organizations have shown improvement in conversion and customer engagement by doing this. One bank, in particular, analyzed feedback on its mobile banking application and noted ease of use was frequently cited as an issue. In response, the bank fully embedded UX resources. It then performed a controlled rollout of the new functionality in one country as an A/B test. After monitoring performance in that country and seeing increased frequency of usage after launch, the change was rolled out to other countries. Fully integrating UX expertise into teams led to improvements in the mobile banking application and an increase in user engagement of 2.25 times (Exhibit 1). The model for integrating UX expertise was subsequently adopted by all major product groups.

Evidence also exists showing the potential of DevOps teams. Research reveals improvements in asset utilization of over 25 percent and significant decreases in provisioning time. According to a recent study by DevOps software provider Puppet Labs, teams that integrate operations and commit to continuous release practices deploy code go times more frequently, have half the number of production failures, and can restore services 12 times faster after a production issue.

■ ■ ■

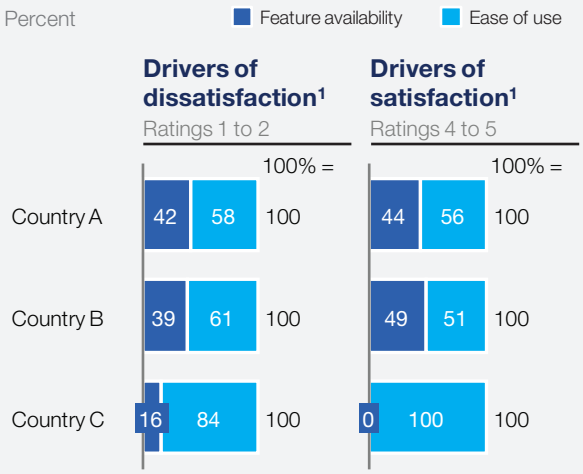
Software's rapid evolution from on-premise toward cloud-based platforms is enabling, among other advances, continuous deployment. With this opportunity, however, comes the need for greater agility within software organizations. The traditional ways of working that isolate developers from the functions of IJX, security, operations, and QA no longer suffice. Organizations must integrate these elements of functional expertise into their development teams. Their options for doing so comprise varying degrees of resource dedication and different organizational structures. They all, however, require developers to understand and apply functional principles in ways that haven't been required before. Those that successfully integrate this expertise are seeing improvements in productivity, quality, and user engagement. ■

Santiago Comella-Dorda is a principal in McKinsey's Boston office; **Chandra Gnanasambandam** is a principal in McKinsey's Silicon Valley office; **Bhavik Shah** is an expert in McKinsey's Silicon Valley office; and **Tobias Strålin** is a principal in McKinsey's Stockholm office.

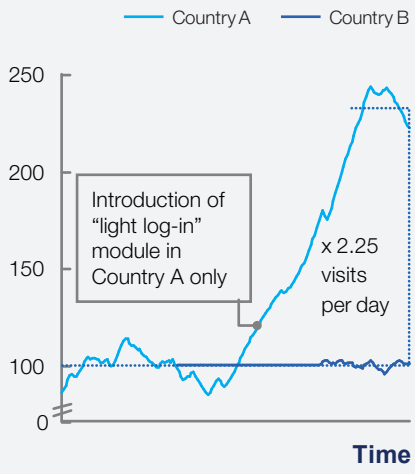
Exhibit 1 **Embedding user experience expertise helped one bank dramatically improve customer satisfaction**

Monthly log-ins per unique user (disguised example), indexed

Ease of use is the primary driver of both satisfaction and dissatisfaction among mobile banking users



Integrating user experience expertise led to the development of a more user-friendly log-in mechanism



1 Based on sample of app reviews, respective rating and theme of the comment, i.e., a review rating the app with 1 star and commenting on the hassle of log-in counts as dissatisfaction and ease of use.
Source: McKinsey analysis

Beyond agile: Reorganizing IT for faster software delivery

The integration of software development with IT operations can rev up companies' delivery of new applications. But this approach may not be right for every part of the IT portfolio

Oliver Bossert, Chris Ip, and Irina Starikova

After more than two decades of experimentation among Silicon Valley giants, "agile" has finally gone mainstream. Companies inside and outside the Valley are using some form of this softwaredevelopment methodology, which emphasizes, among other things, rapid building and frequent delivery of software and system updates, with continual user involvement. 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 (Exhibit 1).

A lot of companies are now kicking the tires on DevOps, the next wave of innovation in software development and delivery and a critical enabler of agile software development. Under this productdevelopment approach, companies seek to fully integrate their software-development functions with their IT operations so teams can jointly build, test, release, and maintain new digital applications more frequently and more efficiently.[1] Software is designed with discrete business requirements and system integration in mind, rather than in a vacuum, and developers and operations staffers are equally responsible for the delivery and stability of code.

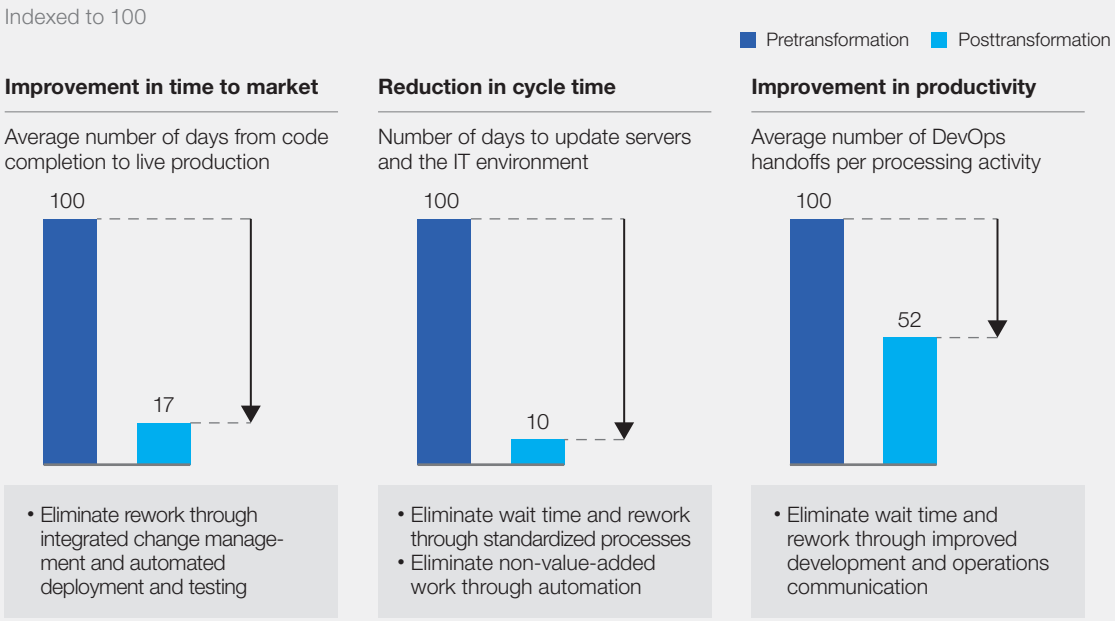
However, few companies, regardless of industry, have been able to reap the full value of DevOps. The implementation of agile has typically affected

Takeaways

Companies are now attempting to integrate their software-development functions with their IT operations. Commonly referred to as DevOps, this approach brings teams together to jointly build, test, release, and maintain new digital applications more frequently and more efficiently.

To succeed with this approach, companies need to do two things: reorient their IT operations around a two-speed IT architecture and identify those parts of the company that would benefit most from DevOps.

Exhibit 1 **The value of adopting DevOps can be significant.**



Source: McKinsey analysis

interactions only among small groups of business stakeholders and discrete application-development teams. By contrast, the move to a DevOps model requires that companies make broader, more systemic changes that could significantly alter interactions among *all* software-delivery teams, IT-operations staffers, and business stakeholders. This is a more complex undertaking.

For most established players, reorienting IT operations around a two-speed IT architecture—which features stable, transaction-oriented systems on the back end and fast-changing, customer-facing applications on the front end[2]—is a prerequisite for implementing both agile and DevOps approaches. But not every application the company develops or every update in a two-speed environment will require the joint collaboration that is central to a DevOps model. Some of the mechanisms used to support rapid development of e-commerce applications, for instance, may not be as successful in building or maintaining applications for a core transactional system developed in COBOL. In those instances, the traditional split of roles and responsibilities among IT operations, software development, and business stakeholders may actually be more acceptable.

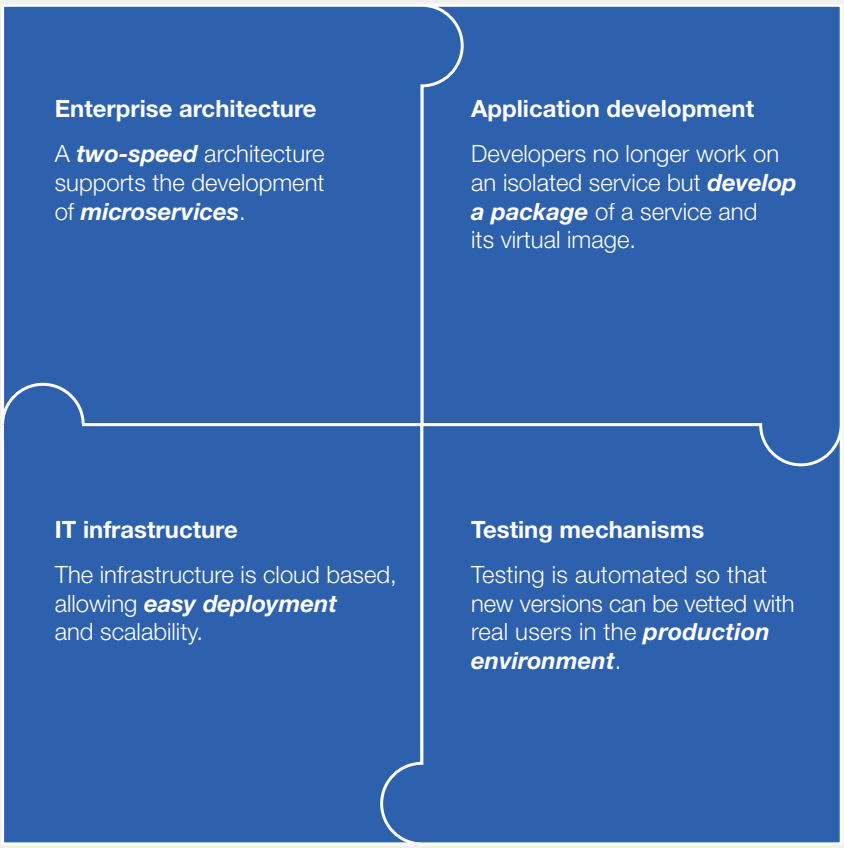
In this article, we will discuss the considerations IT executives face when trying to adopt a DevOps model within a two-speed IT environment (Exhibit 2). They will need to determine how and where to introduce new technologies, such as automation and cloud platforms, depending on which parts of the company they think would benefit most from a DevOps approach. And they will need to explore new production processes and forms of governance so that IT operations and software-development functions across the company can work together effectively, despite the fact that they may be operating at different speeds

Running at two speeds

Over the past decade or so, companies that were born online have revolutionized how technology infrastructure is built and maintained, and how software applications are developed and deployed. They have been among the first to integrate their software-development functions with their IT operations and focus on continuous delivery of small upgrades, where teams rapidly design, integrate, test, deliver, and monitor software changes.

Netflix, for instance, has created a cloud-based IT architecture that allows its developers to launch

Exhibit 2 **To deploy DevOps in a two-speed IT environment, companies need to pay attention to the following factors.**



Source: McKinsey analysis

hundreds of software changes a day. Its website comprises hundreds of microservices hosted in the cloud, and each service is maintained by a dedicated DevOps team. Developers don't need to request resources from the IT operations team; instead they can automatically build pieces of code into deployable web images. As those images are updated with new features or services, they can be integrated with Netflix's existing infrastructure using a custom-built, web-based platform on which infrastructure clusters are created. Testing is carefully done in the production environment with a subset of users. Once the web images are live, a load-balancing technology routes part of the traffic to them from older versions. Automated monitoring ensures that if something goes wrong with the deployment of new images, traffic is routed back to older versions, and the new images are rolled back. Because of this level of automation, Netflix can deploy new code into its production environment within hours, where most companies would need months.

Of course, Internet companies such as Netflix have had the advantage of being able to start from scratch with their IT architectures—with no complex legacy systems to either reconfigure or maintain. And because their main products, web applications, are 100 percent customer facing, these companies have learned how to react quickly to customer feedback and release new features and improvements on the fly.

By contrast, most non-Internet companies seeking to similarly adopt a DevOps model are often saddled with older, transaction-based systems that they must somehow reconcile with agile approaches to software development. What's more, not every function within the brick-and-mortar organization will require DevOps; this would be the case, for instance, for systems of record that are not time sensitive, such as a general ledger. These companies therefore must not only contend with developing a two-speed IT architecture but also enabling a twospeed IT organization.

Managing a two-speed IT architecture

A two-speed IT architecture allows large-scale companies to accelerate the release of innovative products and applications that could make a substantial difference to customers while maintaining legacy IT systems that are less innovative but still necessary for the stability of the business. This sort of architecture emphasizes tight integration between the software applications being developed and the hardware infrastructure supporting them. Historically the IT operations teams maintaining software and hardware have been kept entirely separate. But with the increasing prevalence of vertical enterprise-resource-planning systems, the advent of network virtualization, and the emergence of software-as-a-service models, the two sides have moved closer to one another. These technology trends have had the effect of removing complexity from hardware stacks and making them more accessible to software developers.

A two-speed environment requires that companies introduce automation tools to support continuous delivery of software—especially in the testing and production phases. Automation can allow for better management of, among other things, the release of software updates, the porting of new code, and the general processing environment. Most important, automation tools and cloud-based technologies can serve as the bridge between legacy IT systems on the back end and consumer-facing applications on the front end, allowing companies to pursue seamless testing, provisioning, deployment, governance, and security of servers and new software releases (Exhibit 3).

A two-speed IT architecture conveys a number of critical advantages, but it takes time, careful consideration, and commitment to establish. Netflix, for instance, developed most of its cloud and automation technologies in-house, but companies have any number of products and packages (some open source) to choose from that can allow them to achieve similar dual-speed performance.

A two-speed IT architecture allows large-scale companies to accelerate the release of innovative products and applications that could make a substantial difference to customers while maintaining legacy IT systems that are less innovative but still necessary for the stability of the business.

The most critical factor in establishing a two-speed architecture is for IT leaders to adopt a capabilitiesbased view of the IT architecture, rather than a system- or process-oriented view. This means identifying and clearly defining those software applications that cut across multiple business units. From a capabilities perspective, for instance, IT leaders could see that certain applications developed for the company's customer-relationshipmanagement (CRM) system may require a DevOps approach while others, such as core banking systems or transaction-processing applications, would not. The CRM system would not simply be considered a system of record, too slow to qualify for a DevOps program. Instead, IT leaders could allocate resources toward "fast" and "slow" applications as required—gaining the critical benefits of the DevOps approach where it is possible to do so.

Managing a two-speed IT organization

While addressing the technology architecture and infrastructure required to enable DevOps, companies should simultaneously consider making changes to various operations, processes, and governance structures in the IT organization and within the business overall.

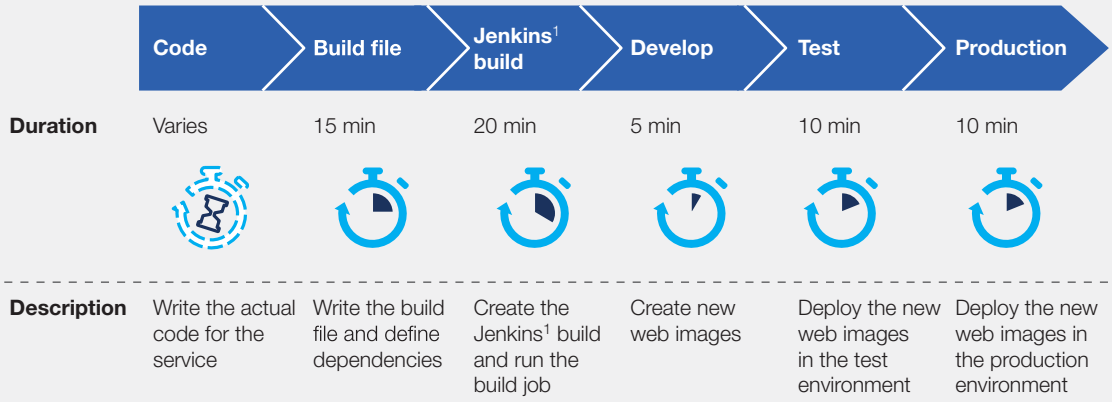
The DevOps approach challenges the established product-development norms in most IT organizations. Historically, companies have separated their infrastructure (hardware) from their application-development (software) organizations and have kept the "build" staff away from the "run" staff. A DevOps approach requires that companies tear down these organizational silos, thereby marking a significant change in IT management strategy. Additionally, IT leaders adopting DevOps organizational models may need to reconsider how technology partners are integrated into their software-delivery processes—a trend that is forcing some system vendors to consider ways to make their platforms available as a service.

The biggest task for IT leaders is to identify those parts of the company where the use of DevOps would make most sense—likely focusing on those parts of the business where speed is at a premium, and where there is a significant opportunity for the company to differentiate its customer experience from the competition. (Think of a retailer using DevOps to improve its website checkout experiences, or a bank offering new fund-tracking capabilities at its site.) For those parts of the business where DevOps might make less sense—where reliability and resilience of software is more important than speed to market—IT leaders will need to determine how to maintain the split between software development and IT operations, and which roles and processes to adapt for a culture of continuous delivery.

Redefined roles

By its very nature, integrated product development requires strong collaboration between business and IT—and in some cases new or redefined roles. Business analysts must communicate the requirements for new software features and functionality in terms that employees in all departments can understand—and they must

Exhibit 3 **It is possible to deploy new code on a site within an hour.**



[1]Jenkins is an open-source continuous-integration application that monitors execution of repeated jobs, such as building a software project.
Source: McKinsey analysis

be flexible and willing to change the business requirements slightly when doing so could speed up implementation. Engineers and product developers must work across functions and among different product teams—under a DevOps model, informal collaboration and coordination among these business and IT coworkers actually becomes more important than formal reporting and approval processes. Software testers must collaborate with developers and business analysts—first with business analysts to clarify feature requests, and then with developers after the code has been developed, giving them immediate feedback on software performance. With DevOps, end users are no longer passive recipients of "big bang" software or service releases—companies seek their input early and often as they develop and test new software features.

Cross-functional teams of application-development, infrastructure-management, and operations professionals should be convened to streamline the ownership of stacks across the application-delivery pipeline. In the case of continuous delivery, for instance, a joint team would oversee all processes (and associated tools) relating to this development activity, such as application building, testing, and deployment; performance management and monitoring; and virtualization and configuration management. Previously, some of these components would be owned by different organizations. Also, infrastructure teams should be given a seat at the table, with decision rights equal to those of softwaredevelopment teams.

Redefined culture and talent

Integrated development and continuous delivery can only happen within a corporate culture that empowers its software developers and refines its IT and R&D reporting structures. In most organizations, product development and IT operations live in separate towers, with people of different mind-sets, skills, and experiences. IT and business executives will need to break down these barriers. For instance, rather than have all developers report to the head of "build" and all operations employees report to the head of "run," some must be purposely assigned different reporting lines. Further, employees will need training opportunities, and their salary schemes may need to be reconsidered. Traditionally, product developers have focused mainly on programming

frameworks; in a DevOps environment they will be held responsible for the quality of their code. They will need to know operating-system basics and must show strong collaboration skills as they work jointly with operations engineers to determine how best to solve application-development or deployment problems. As a result, many companies are already modifying their recruiting practices to hire "full stack" engineers—professionals who understand all aspects of computing, including user interfaces, databases, and networks.

Redefined processes and governance

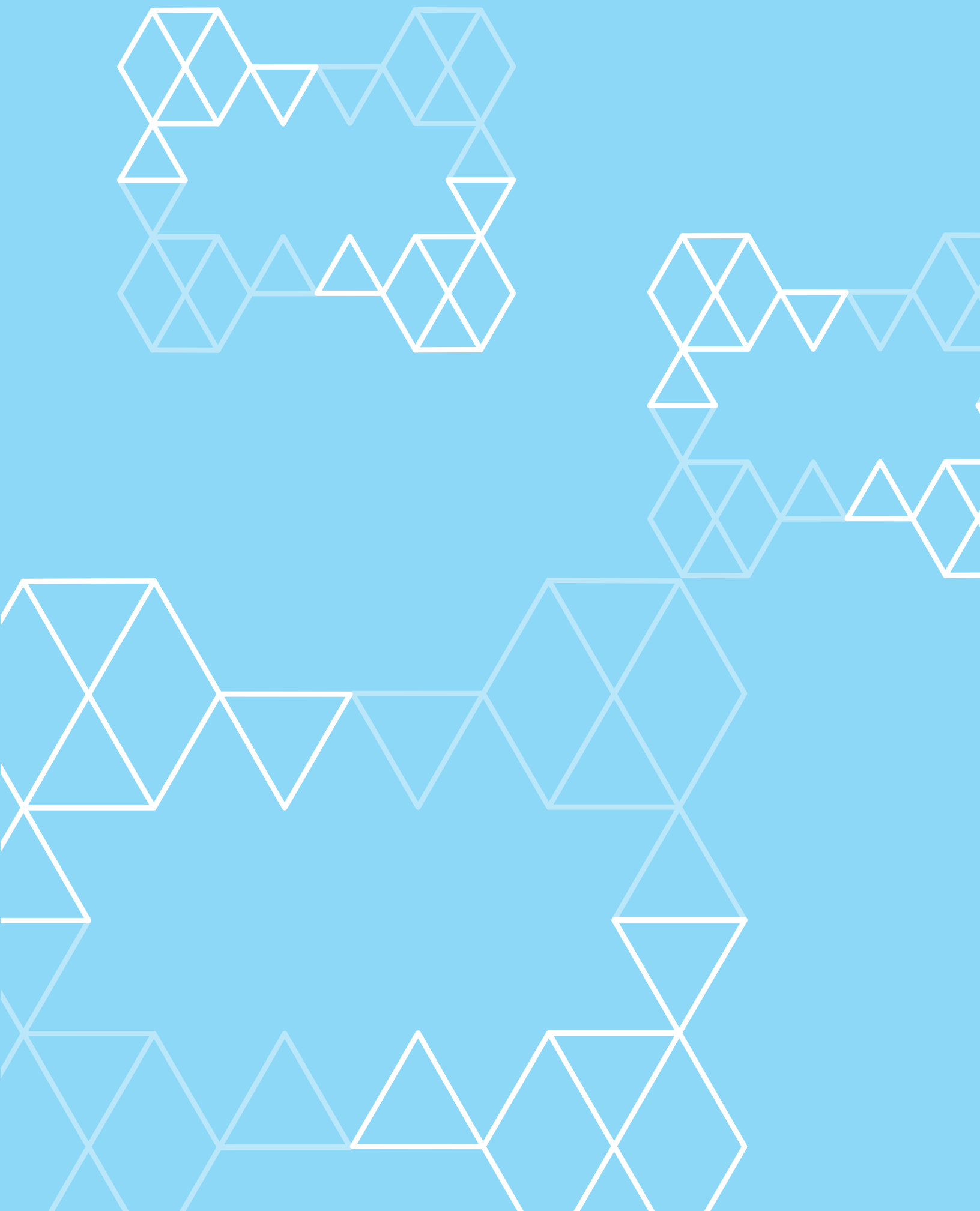
Companies may want to look across the entire spectrum of software-delivery processes to determine which will need to be redefined or fully automated so that development teams can take advantage of infrastructure as a service, as needed, and so that code can be ported into testing and production environments in a standardized way. There are a number of lessons companies can take from Internet pioneers on the types of process and governance changes to deploy in support of DevOps. For instance, Internet companies enforce "self service" for developers; teams can test, promote, and deploy code in production environments without requiring constant handson involvement from infrastructure-operations teams, although both teams share responsibility for code performance. Internet firms also impose rigorous, automated testing of new code at all stages of the application-development process; in some dot-com companies, sophisticated tests are completed automatically every 10 to 15 minutes. Additionally they take advantage of advanced analytics and other tools to preemptively scan code for exceptions and send developers automated reports about the code segments that are most likely to create errors.

■ ■ ■

The value of implementing DevOps can be significant with respect to both productivity and time to market. But the implementation of DevOps is not simply about the deployment of new IT methodologies. It must be treated as a companywide transformation—one that incorporates process and governance considerations as well as technology-related ones.■

[1] Satty Bhens, Ling Lau, and Shahar Markovitch, "Finding the speed to innovate," April 2015, mckinsey.com.
[2] Oliver Bossert, Chris Ip, and Jürgen Laartz, "A two-speed IT architecture for the digital enterprise," December 2014, mckinsey.com.

Oliver Bossert is a senior expert in McKinsey's Frankfurt office, **Chris Ip** is a director in McKinsey's Hong Kong office, and **Irina Starikova** is an associate principal in the Silicon Valley office.



Driving Quality at Scale through Pair Programming

Pair programming can help development teams produce software with fewer errors and greater reliability.

Sanjay Kaniyar, Amit Rahul, and Philip Wiltshire

In today's hyper-connected, click-speed environment, a single coding error in a critical financial system can have break-the-company repercussions. In 2011, for instance, the Securities and Exchange Commission (SEC) levied a $25M penalty on an investment firm after a software program applied the wrong decimal-to-percentage conversion in its portfolio risk estimate, a seemingly minor error that caused about $217M in losses to more than 600 client portfolios . And in June 2014, the program behind the Institute of Supply Management's monthly purchasing manager's index inadvertently set off a 35 point drop in the Dow Jones after it accidentally used the wrong month's data in its calculations . With financial institutions becoming ever more digitized and with billions of dollars at play every day, the risks of coding missteps are growing in number and severity.

CIOs of financial institutions are well aware of these risks, of course. But, despite the high profile nature of some strategic software projects, there is often very little differentiation when it comes to quality assurance practices. Big projects may get more resources and status review meetings, but the standard review protocols remain largely the same. That can work well for routine projects or those with longer lead-times, but many new, mission critical software initiatives come with tremendous complexity and accelerated time-to-market requirements that require a different approach.

In response, some IT organizations are getting closer to the source and looking to rid coding errors where they begin, at the developer level. They're embracing pair programming, a software development technique in which two developers jointly work on the same problem, each acting as a sounding board for the other. That kind of approach often allows teams to find creative solutions more quickly and produce software with fewer errors and greater reliability.

Using this joint problem solving method, pairs take turns "driving," tapping away at the keyboard to write the code and "navigating," reviewing the lines of code as they're written and keeping the end deliverable in mind. The approach allows both programmers to back each other up in designing, innovating and trouble-shooting.

Pair programming can confer many benefits, among them better design, better quality software, and increased code readability. A recent survey that we conducted found teams that employed pair-programming improved time-to-completion by 20% and reduced defects by 40%, with benefits accruing throughout the application lifecycle. In addition to lower maintenance costs, pairs avoid over-production by developing acceptance and unit tests first. And they avoid "gold-plating" by writing just enough code to successfully pass well-defined tests. Because every task on a development cycle is explicitly agreed upon with the business users at the outset, teams can focus their time on the right areas – redirecting efforts away from low-priority functionality – and ensure the software delivers as intended, steps that sharply accelerate the development curve. Cross-training is also better, since role and project switching speeds the ramp-up time for new developers and allows skills to be distributed among the team. Managers like the accountability and collegiality that pair programming brings, as opposed to the lone-crusader working style typical of traditional development practices.

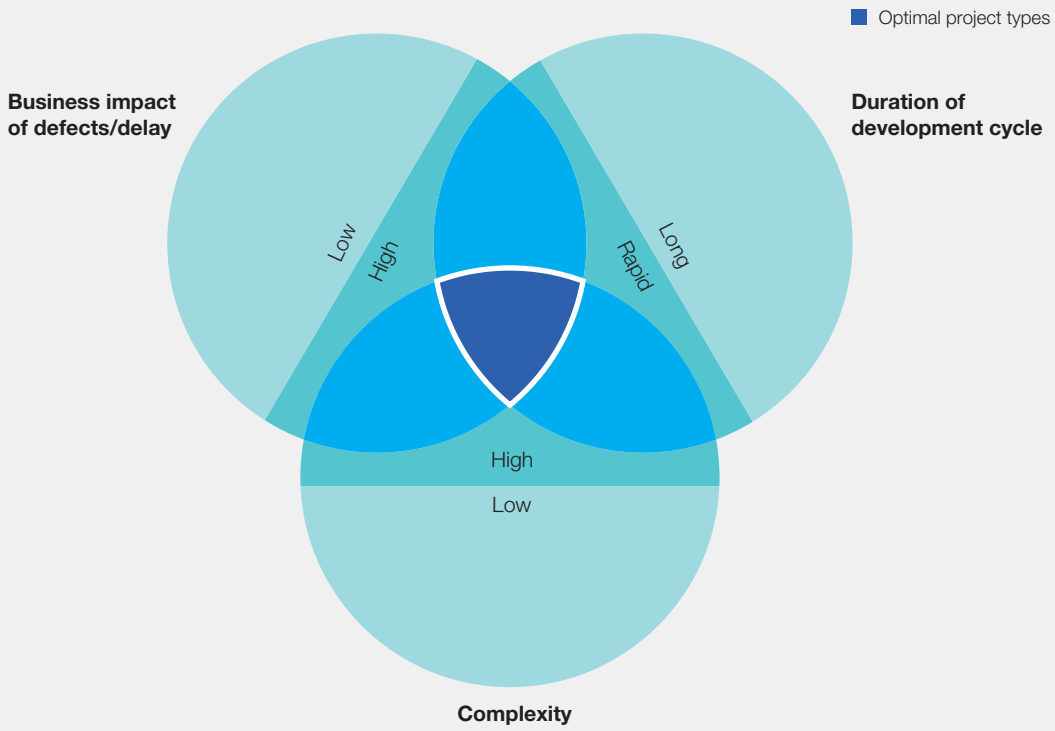
The financial services industry stands to gain particularly. Because their industry is characterized by significant risk exposure, pair programming can be selectively deployed to help with complex, rapidly evolving, business critical applications where time-to-market and quality are essential.

To capture these benefits, pair programming should be used in a deliberate way and scaled appropriately. Organizations have found the following practices to be helpful:

1. Launch a pilot program with a team that is already doing agile development. The most appropriate projects for pair programming are complex tasks that require significant problem solving where quality and time-to-market are primary goals.

2. Cultivate the right mindsets. This requires a top-down commitment to capability-building at all levels. Not everyone is suited to pair-programming. It's essential to attract people who thrive in collaborative environments and are comfortable with having their work reviewed at all times to drive adoption and roll out pair programming in multiple waves across the organization.

Exhibit 1 **Optimal type of projects for pair-programming deployment**

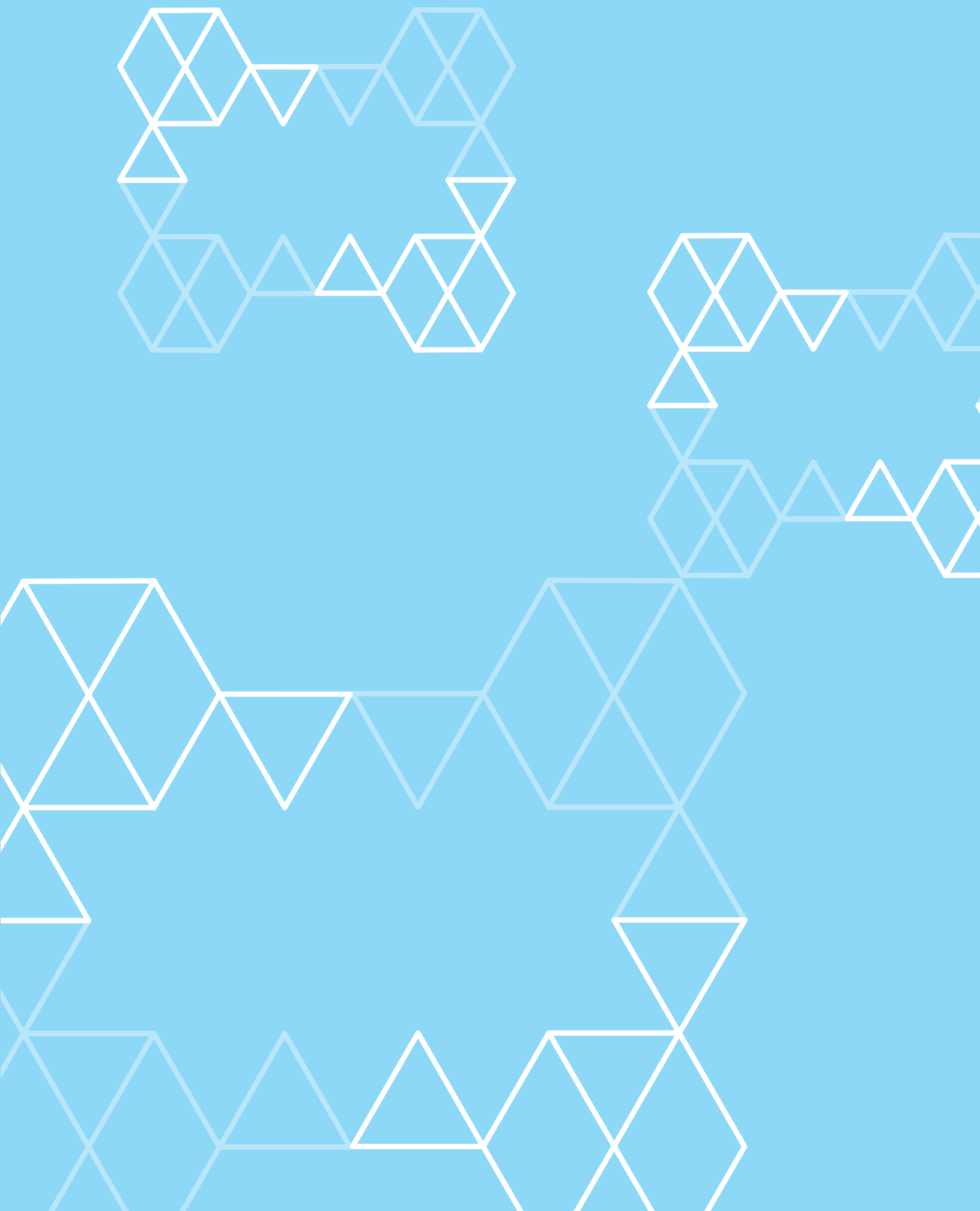


3. Design a transparent work intake and prioritization process. Invite business partners to "show-and-tell" and planning meetings for each development cycle to build support. The frequent participation of the business in regular progress reviews helps accelerate both progress and buy-in. The use of charts and other visuals, in particular, often makes reviews more accessible. To keep the programming process on track, pairs should be encouraged to take on tasks from others if they have time left during the cycle, but rewards must be tied to quality and not quantity.

Organizations that have used pair programming successfully not only reduce the risk of coding errors but help instill a work culture that promotes ownership and continual learning, characteristics that go a long way towards improving software quality and reliability.■

[1] http://www.sec.gov/news/press/2011/2011-37.htm
[2] http://www.forbes.com/sites/leoking/2014/06/03/oops-bad-software-code-knocks-down-us-manufacturing-dow-jones-and-sp-500/2/.

Sanjay Kaniyar is Associate Principal with McKinsey & Company; **Amit Rahul** is an Associate with McKinsey; and **Philip Wiltshire** is a Consultant with Nationwide Insurance. The authors wish to thank Jason Patterson, Mark Vair, James Grafmeyer, Michael Jones and Tom Paider from Nationwide Insurance for their contributions to this article.



Combinatorial Testing

A **scientific** approach to **improving** test **coverage** with **less effort.**

What is it?

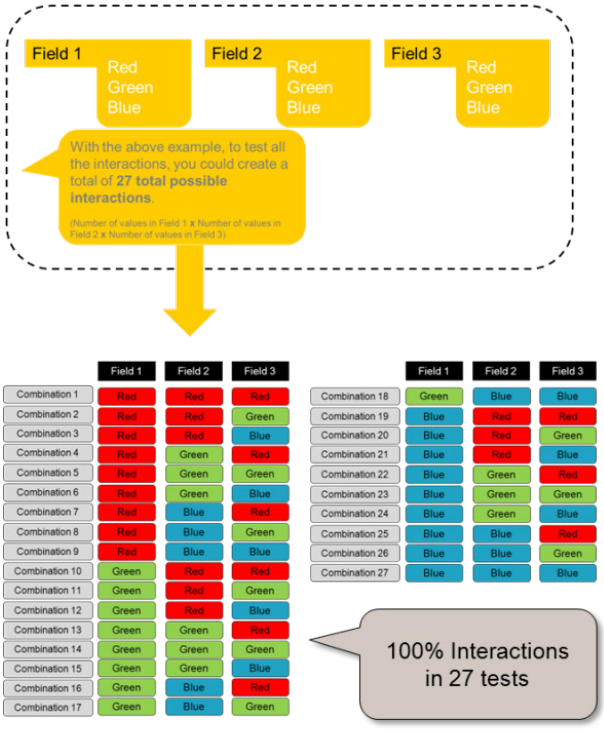
- Reduce exhaustive testing, over production of test assets, mitigated through a mathematical approach to testing
- Removes guess work
- Increase collaboration between teams
- Move quality assurance to the left
- Overall improvement in coverage with reduction in risk

What is it NOT?

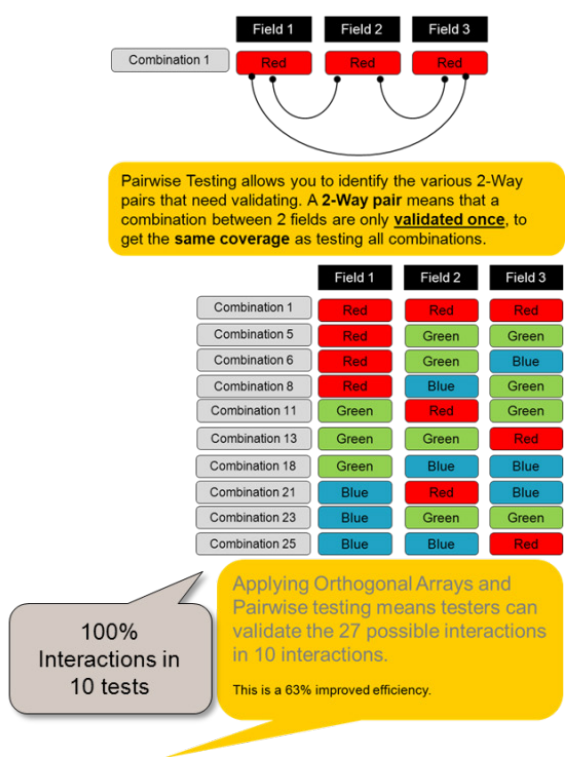
- Silver bullet for testing
- Replace ALL testing activities

Example – Ensures that Field 1 Red and Field 2 Red is only validated once.Thus, eliminating non-value add tests.

Current Method of Testing



Pairwise Testing Method



Benefits

Improved Collaboration
All communities working together to finalise requirements and resolve clarifications

Transparency
All teams will understand what testing is covering and how conditions are derived

Defects Identified Earlier
Design gaps and requirement gaps are identified earlier – before coding where its cheaper and easier to fix

Reduced Walkthroughs
Save the cost and effort for having multiple revisions and walkthrough of artifacts

Consistent Understanding
All teams will have the same understanding of what the business wants

Reduced Test
Able to achieve an improved test coverage with a reduced but optimal set of tests.

Potential Savings
Save potential costs from analysis, preparation and execution

Steps To Success

Step 1
Apply

- Understand that combinatorial doesn't apply to everything.

Step 2
Prep

- Identify and understand the process being validated
- Create preliminary model with SME's

Step 3
Attend

- Ensure you have **ALL** the required attendees to the workshop including Business, BA, Dev, Test & Architect (if required)

Step 4
WS #1

Workshop #1
Process Overview

- **Run by B.A**
- Define E2E Logic Flow
- Identify Alternate and Exceptional Flows
- Confirm Agreed Changes
- Identify Parameter attributes

Step 5
WS #2

Workshop #2
Detailed Analysis

- **Run by Test**
- Retrofit Model as required
- Attribute finalisation
- Prioritisation
- Gap Analysis

Step 6
WS #3

Workshop #3
Playback & QA

- **Run by Test**
- Walk through model
- Further Gap Analysis
- Pre-Conditions and Post Conditions
- Finalise E2E Coverage

Step 7
Final Review & Signoff

- Ensure you have **ALL** the required attendees to the workshop including Business, BA, Dev, Test & Architect (if required)

Team level

Acquiring the capabilities you need to go digital

Some of the talent and tools won't necessarily be found in-house. Here's how to create a sustainable strategy for sourcing the right people and products.

Matthias Daub and Anna Wiesinger

Digital capabilities are now a prerequisite to compete in the long term.[1] Yet many companies seeking to go digital are still unclear about the best way to set up their IT organizations and develop the tools and talent required to manage digital information and establish and maintain online services and automated processes.[2] What most don't properly acknowledge at the outset is that many of the critical resources required to facilitate this transition will not be available in-house.

The right talent, for instance, may be in short supply. The distinctive experts required to develop successful digital offerings and channels might include product managers who are literate in cutting-edge technologies that can be used to reshape the consumer decision journey,[3] experienced business and data analysts who can extract useful insights from customer data, and user experience experts and design-oriented content managers who can ensure that the offerings will appeal to target audiences. But technology-services companies are often better positioned to win the battle for professionals with these skills because they can offer more diverse career paths and personal development in the field.

And while in many sectors the time to go digital has come, developing the required capabilities in-house can take years. The IT department in one major travel company, for instance, recently embarked on a digitization initiative—a move prompted in part by increased competition from online players and eroding margins. The goal was to scale up quickly, but in assessing their needs, leaders realized they lacked necessary expertise in a number of critical

Takeaways

Companies need to move quickly to go digital, but building in-house capabilities can take years. A dynamic approach to accessing expertise from outside the organization can help them compete. To take full advantage, players need to reconsider their digitization targets and redefine their operating models and capability-building efforts.

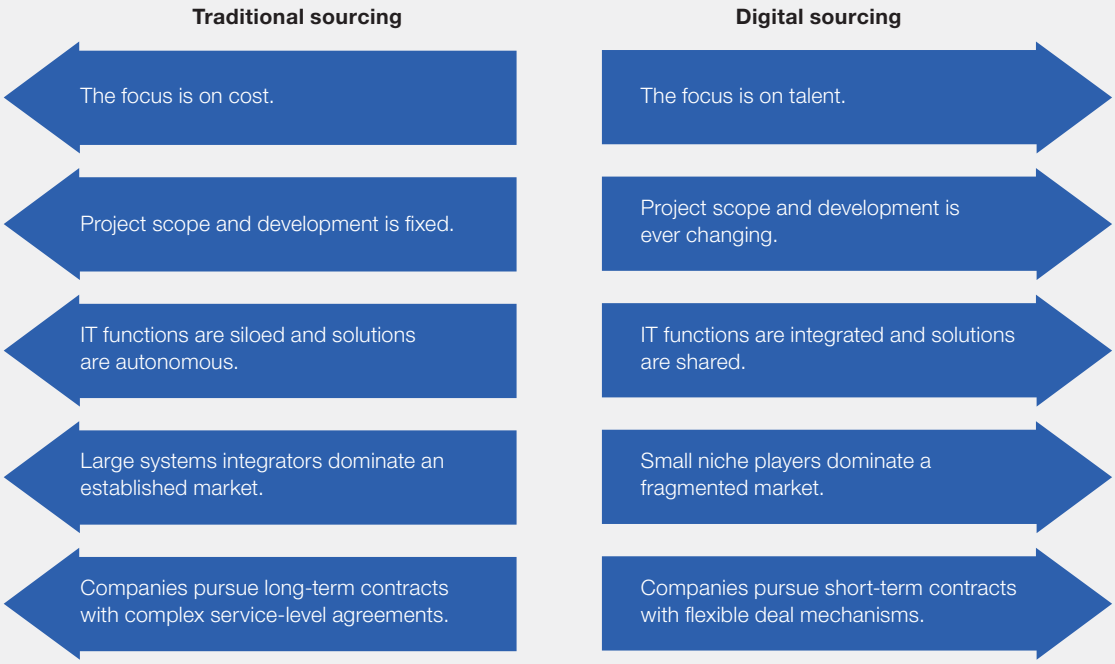
technology areas, including user-interface design and agile software development. The company's existing IT organization was equipped enough to manage smaller-scale digital projects but not a full-on digitization initiative. Hiring, developing, and retaining the appropriate talent would require a sustained effort.

To better compete, the travel company and others like it need to adopt a dynamic approach to accessing digital capabilities from outside the organization. In large part, this will require learning to balance the two speeds at which IT organizations must operate—integrating slowly changing legacy transactional back ends with more dynamic customer-facing front-end systems and applications.[4] Specifically, we believe companies must take a closer look at their digitization targets, operating models, and capability-building practices in the context of this two-speed architecture. In this way, they will be able to scale up nascent digital initiatives quickly and sustainably: accelerating the use of emerging technologies, aligning the fragmented activities being pursued by individual business groups, and developing vendor relationships that can evolve with customers' changing needs.

Sourcing for digital is different

In the fast-moving world of digital, finding the necessary tools and talent can be challenging for several reasons (Exhibit 1).

Exhibit 1 **Sourcing for digital differs in a few key ways.**



Talent has become more important than cost

Traditionally, companies have realized competitive advantage from technologies that support a specific business requirement at the lowest cost possible. The travel company, for instance, might rely on simple Excel spreadsheets to assess internal data sets in a limited way rather than rolling out more-expensive systems to capture and analyze data. In a digital environment, however, having cheaper technologies is much less valuable than having the right talent behind them. Companies must factor into their digital transformations greater investments in talent despite the potential increase in overall IT costs. For those that do, the payoff can be significant. For example, the travel company might be able to predict customer demand with greater accuracy—and get a leg up on competitors—by investing in an advanced data-capture system that incorporates multiple external and internal data sources and by hiring the analysts required to make sense of the collected information. Our work with global companies suggests that in creating and rolling out cutting-edge technologies, "top talent" developer teams are between five and ten times more efficient than "average" developer teams of a similar size.

The scope of projects is ever changing—and must be managed that way

Companies undergoing traditional business transformations have tended to adopt a sequential approach to acquiring the skills and tools they need. Talent and technologies are brought in based on software-release initiatives that are rigidly scheduled; nothing is rolled out to customers before it is fully complete. This approach is inadequate for digital transformations; by nature, these projects are iterative and call for continual clarification of targets and hence updates to internal requirements. One of the important advantages of digitization, after all, is that companies can capitalize on opportunities for end-to-end customer-centric innovation, where targets are constantly refined and experimentation is encouraged. Using this approach, for instance, a company might want to update its online offerings with new functionality more frequently (say, every week) and introduce them one by one—gathering feedback from customers and revising its website in a test-and-learn fashion—rather than launching all front-end and back-end changes at once.

Integration and accountability are more critical than ever

A large financial-services organization had empowered a number of teams to launch their own digital businesses. However, instead of working with the bank's IT organization to find the required talent and resources, each of the project teams went out on its own to negotiate deals and forge relationships and alliances. The result was a confusing web of suppliers operating under different contracts. Because there was no adherence to a central sourcing strategy, there were inevitable compatibility, security, and productivity issues when new code was introduced in production environments. For successful digital transformations, companies must ensure that the IT department is looped in and accountable for the delivery of sustainable, integrated solutions that meet the business's needs regarding time to market and flexibility.

The supplier market is fragmented

Digital transformations rely heavily on the use of emerging technologies and specialized tools, many of which are being developed by innovative niche players and start-up companies rather than the established systems integrators that currently dominate the market. Consider the number of small niche players who have already built extensive experience in big data analytics, for instance, versus the number of large systems integrators who are only just beginning to traffic in big data. The provider market, therefore, has become increasingly fragmented, less transparent, and, in many ways, is still quite immature—all of which complicates sourcing decisions for companies looking to go digital quickly. (See sidebar, "Picking the right digital vendor.")

New deal mechanisms are required

The processes by which companies have traditionally struck deals with external providers may not be fully applicable in the case of digital

Picking the right digital vendor

The market for digital suppliers is dynamic and, to this point, has been characterized by a large number of smaller and younger players. These boutique players provide innovative, specialist services—for instance, user-interface design; analytics, business-specific platforms, services, and application programming interfaces; and cloud services. They are widely perceived as better able (and willing) to embrace digital trends. However, some incumbent players are entering the market, either because they've developed their own innovative digital offerings or acquired platforms, technologies, and talent from the start-ups. Because of this volatility in the marketplace, it is critical for companies to carefully assess vendors and products, looking beyond price. Specifically, when selecting a partner for their digital transformations, companies should consider the following:

Long-term prospects. It is important for companies to provide long-term perspectives and incentives as part of partnership agreements, especially when the contracts involve smaller industry players that can disappear from the market quickly or be targeted and taken over by larger companies. Agreements should contain financial incentives that reward positive relationships and that provide predictability for smaller players.

Ability to cocreate. Suppliers must be able to collaborate with companies' IT and business stakeholders and cocreate products that will work with legacy systems. To assess whether potential suppliers are a good fit, companies can launch experimental pilots to get to know different providers.

Supplier flexibility. Ambiguity is part and parcel of most digital transformations. Vendors must therefore be able to adapt to ever-evolving requirements but also adhere to agreed-upon architecture-design principles. In this way, companies can create sustainable products and systems that can factor in contributions from multiple suppliers.

Lock-in requirements. Technologies and platforms are rapidly evolving, and it is not clear yet which ones will emerge as future standards. When selecting a product or platform today, there is always the risk that it is based on a niche technology that will become obsolete. And if a company depends on a supplier's proprietary technology, this can become a major business risk. Using open-source solutions can mitigate this risk; many suppliers offer corporate editions of their open-source products and services, and some companies may decide to open up their proprietary solutions for input from the open-source community. However, lock-in risk cannot be avoided entirely: some service providers tend to customize open-source solutions to an extent that creates de facto lock-in. It is therefore important to consciously select the products that suppliers prefer to use as part of the sourcing process.

transformations. Such vendor deals are often focused on specific solutions and predefined deliverables, and contracts can last for years—longer than the entire life cycle of some new and emerging technologies. Instead, companies will need to explore new deal mechanisms, such as risk-sharing agreements and innovative pricing schemes that reward experimentation and collaboration.

Finding the capabilities you need

The companies that recognize and strategize for these digital differences can scale up their online capabilities at short notice. To successfully pull in new talent and tools from the market and better manage the varying speeds at which digital transformations can occur, companies must deploy a more dynamic sourcing strategy. Specifically, they must reconsider their digitization targets and redefine their operating models and capability-building practices accordingly.

Digitization targets

Companies need to identify the business domains and activities that would most benefit from rapid digitization, and they should manage those projects separately from their conventional IT projects.[5] These will typically include customer-facing applications and internal systems that involve a high degree of employee interaction (for example, intranets and employee self-service tools). Companies may also want to reassess their attitude toward sharing critical knowledge (for example, sales data and advanced analytics expertise) with vendors to accelerate their development of digital capabilities—such as, being able to price products based on customers' purchase histories.

Operating models

Once target areas for digitization have been identified, companies will want to determine the best-fit suppliers and the appropriate engagement model required for each. Taking a close look at their operating models can help companies account for the ever-shifting scope as well as the integration and accountability issues that are unique to digitization projects. The usual staffing and service models, which emphasize rigid communication channels and objectives, may still apply, but for each supplier, the project-management model may need rethinking. Consider the financial-services company we described earlier: its various digitization projects require an agile software-development approach, so the company may need to build formal and informal meetings with vendors and the central IT group into the development schedule to allow for frequent feedback loops. The supplier, the financial-services company, and the central IT organization may not be able to agree on a fixed set of deliverables up front, due to the iterative nature of lots of digital projects blooming at the company. But all parties may be able to standardize some basic metrics to judge performance. Agile teams working on each of the initiatives could rely on "story points," for instance—the development team's estimation of the difficulty of meeting a requested business need—or the amount of work expected to be delivered over certain periods of time.[6]

Another important success factor is making clear the activities that external vendors will provide versus the activities that will be performed in-house. Traditionally, the tasks required from suppliers are usually specified in a service catalog, which would include descriptions of critical activities and clear distinctions about which party owns which tasks—for instance, the supplier might be responsible for testing certain new system components, while the client might be responsible for running tests to ensure that its customers actually like the new features.

This philosophy also holds true in digitization efforts: vendor relationships will be much more successful, and there will be less friction, if there is explicit collaboration between companies and suppliers. Both sides must follow rapid and flexible delivery practices, and within the customer organization, IT must play a central role in bridging the cultural differences between providers of new technologies and providers of established legacy and back-end solutions.

Best practices in supplier management

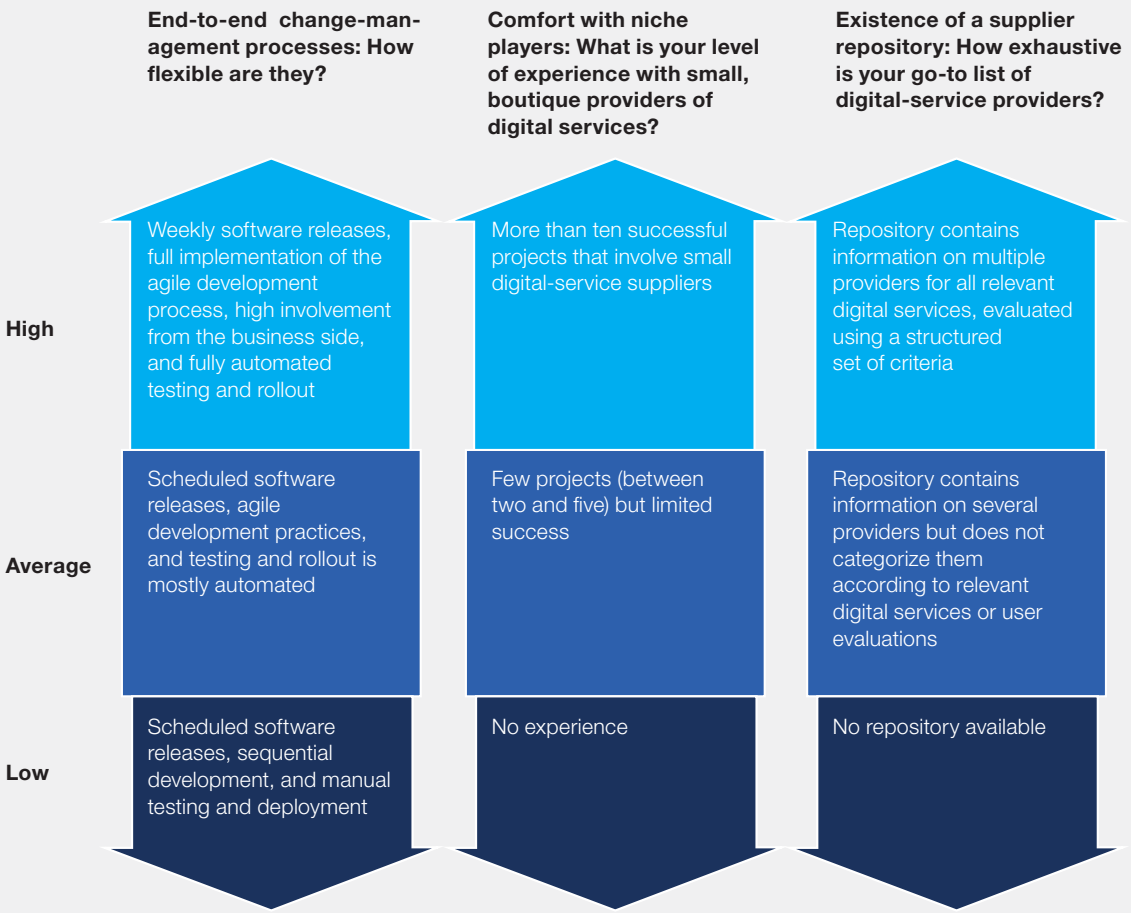
When dealing with new suppliers, companies must acknowledge the market-fragmentation and deal-mechanism challenges described earlier. They must be able to articulate the IT demands from various business units, specify requirements but be willing to revise them periodically, manage integration efforts, assess (and occasionally reassess) the suitability of vendors, evaluate the stability and future support of solutions from small vendors (including the use of open source), make frequent decisions to stop or continue with relationships, and dynamically manage the capacity and capabilities required from external vendors. Small players providing new products, platforms, and programming interfaces tend to have narrow, specialist functions; few can provide end-to-end integration or complete solutions as a service. Therefore, experimental deals are becoming the standard—that is, contracts with short time frames and a narrow performance scope but that rely on both quantitative measures (such as the speed of development) and qualitative measures (such as collaboration style).

When dealing with established suppliers, companies must adjust their existing contracts to ensure that these often larger vendors are working in sync with smaller, more nimble players—maintaining and enhancing legacy systems while simultaneously allowing for rapid front-end changes. Existing contracts may need to be rewritten to allow for seamless and timely interactions between the two.

Companies will want to build and continually update repositories of potential digital-resource providers, noting the suppliers' experiences, locations, and differentiating capabilities (Exhibit 2). Before committing to digital-service providers fully, companies can engage them in small, low-risk pilot projects to assess fit. And to build relationships and expand their contact lists, companies can get involved in open-source and other technology communities by participating in conferences, roundtables, and other industry-convening events.

Exhibit 2 **How ready are you to source digital capabilities?**

Companies can look at a number of organizational elements to determine how prepared they are to undertake sourcing for digital, but these three are particularly critical.

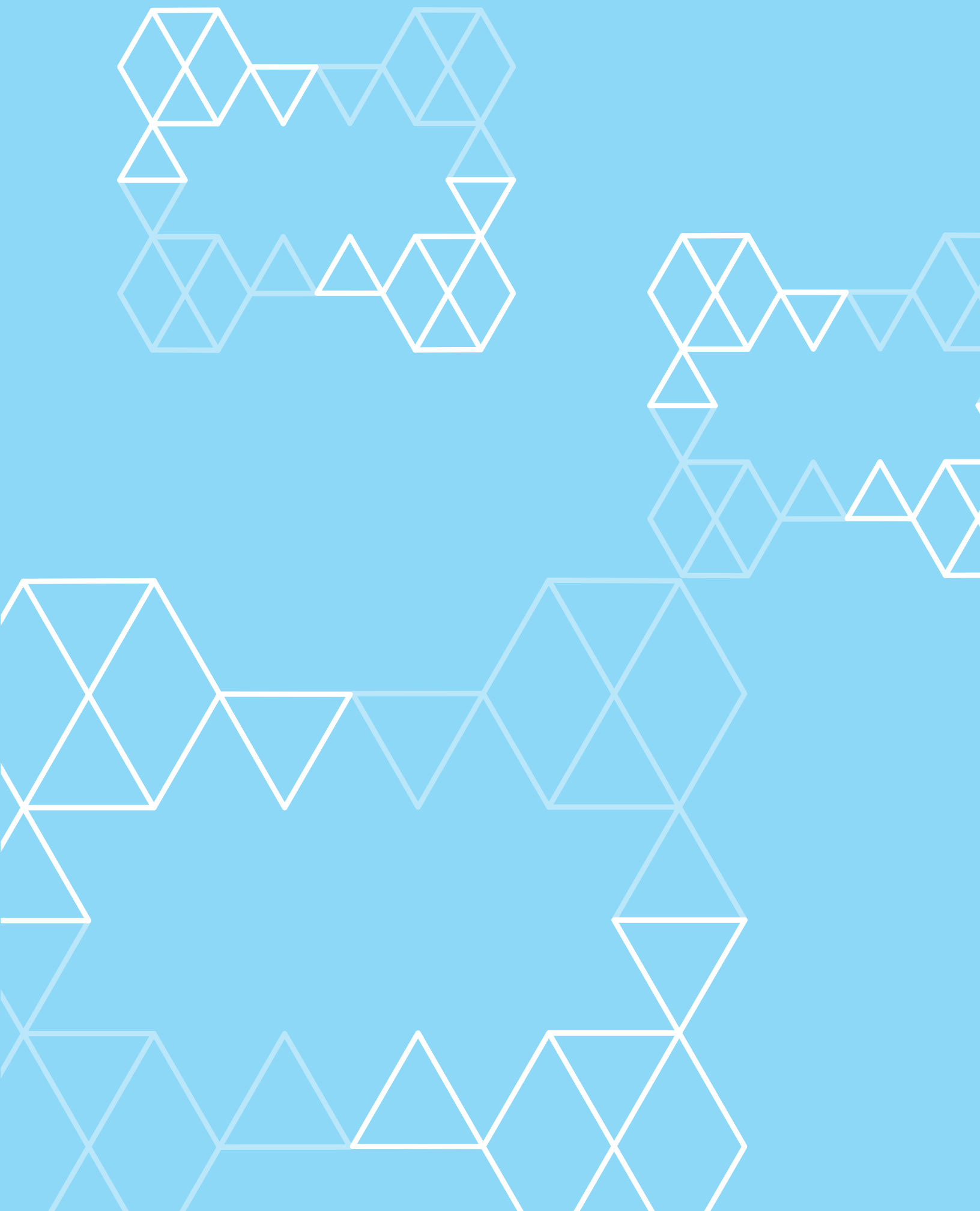


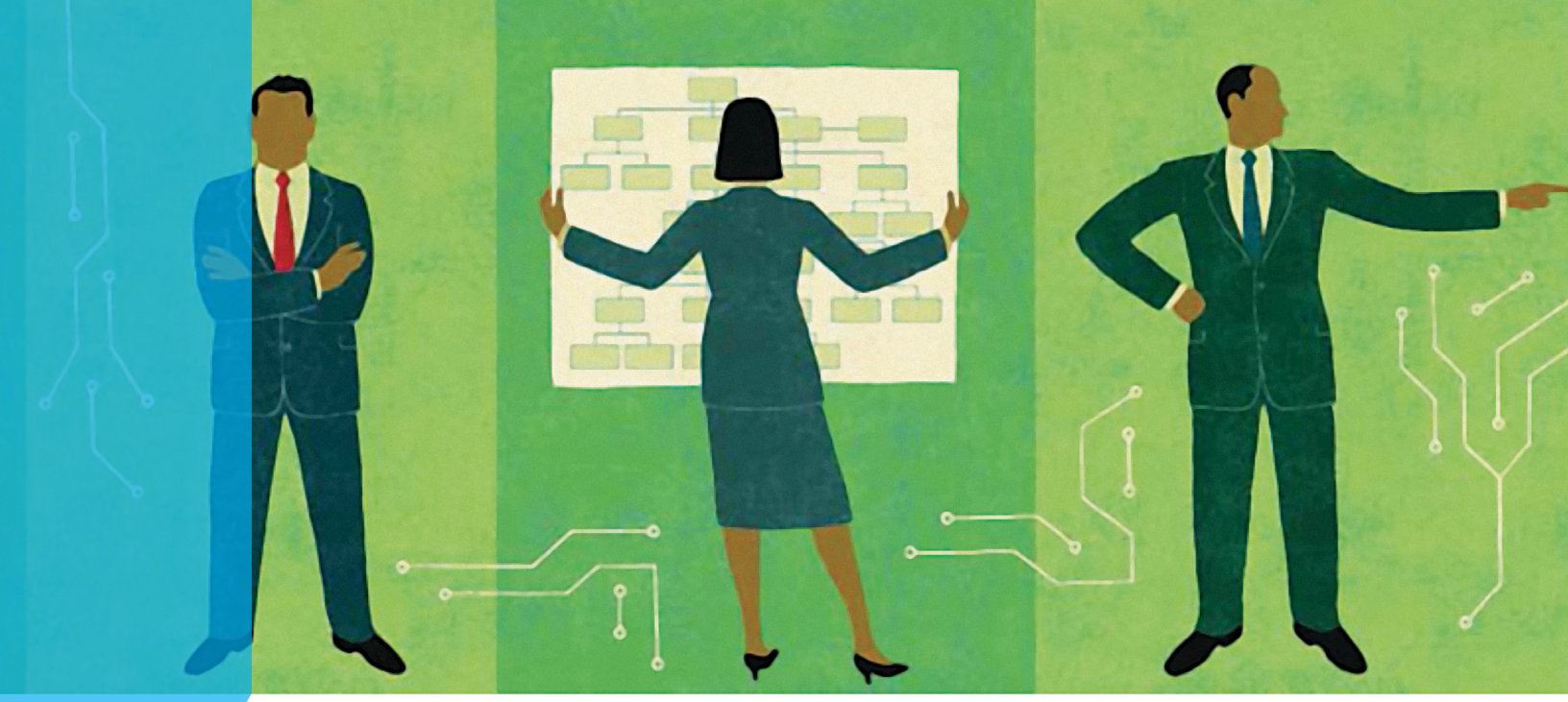
Sourcing for digital is merely one building block for going digital—albeit a critical one that can fill the talent and tool gaps often faced by those that are not digital natives. To start a successful digital journey, companies need to fully embrace the idea of a transformation that extends across all parts of the organization. This means establishing rapid decision-making and escalation processes to match the digital way of working and exploring the use of new standards for contract elements and terms and conditions.

The sourcing journey won't be easy; looking outside for digital capabilities likely will involve a number of starts and stops. Over the long term, however, this approach can help companies meet the challenges of innovating and competing more effectively online.

[1] Martin Hirt and Paul Willmott, "Strategic principles for competing in the digital age," McKinsey Quarterly, May 2014, mckinsey.com.
[2] Henrik Andersson and Philip Tuddenham, "Reinventing IT to support digitization," May 2014, mckinsey.com.
[3] The consumer decision journey refers to the process by which consumers research and buy products. It is a circular journey encompassing four phases: initial consideration, active evaluation, closure, and postpurchase.
[4] Oliver Bossert, Chris Ip, and Jürgen Laartz, "A two-speed IT architecture for the digital enterprise," McKinsey on Business Technology, November 2014, mckinsey.com.
[5] Ibid.
[6] Michael Huskins, James Kaplan, and Krish Krishnakanthan, "Enhancing the efficiency and effectiveness of application development," August 2013, mckinsey.com.

Matthias Daub is a principal in McKinsey's Berlin office, and **Anna Wiesinger** is a consultant in McKinsey's Düsseldorf office.





Developing talent for large IT projects

Good program managers are hard to find. Here's how organizations can attract and develop their own senior IT program-management talent.

Francine Debane, Katya Defossez, and Mark McMillan

Large IT-driven transformation programs are important for creating business value in all industries. However, our research, conducted in collaboration with the University of Oxford, shows that 71 percent of large IT projects face cost overruns, and 33 percent of projects are around 50 percent over budget. On average, large IT projects deliver 56 percent less value than predicted.[1]

Boosting the capabilities of IT leaders is critical for improving the outcome of these efforts. In a 2013 McKinsey global survey, executives were asked about levers for improving IT performance. Thirty-five percent of respondents said improving the overall level of talent and capabilities of IT staff was one of the most important initiatives, and 20 percent identified replacing IT management as another key lever.[2]

The responses reflect the challenge of attracting, developing, and retaining the right IT talent at a time when building a digital enterprise has become a priority for most companies. To succeed, organiztions need to cultivate in-house talent for roles that require intimate knowledge of the business and the organization. Enterprises must recognize the value and scarcity of employees who combine IT savvy with business acumen and must build and support a staff of such people. In this article, we explore three steps that organizations can take to develop the right talent to manage large IT programs.

Focus on the roles that really matter

Companies can burn significant energy trying to cultivate and develop skills that could be outsourced. Based on our research and experience, however, there are three roles that are vital to the success

Takeaways

Large IT-driven transformations typically run over budget and over time and underdeliver on their predicted value. Hiring, developing, and retaining the right talent for the roles of IT program manager, business change leader, and IT architect is critical for improving the outcome of these efforts.

To attract and retain the talent needed in these roles, companies should offer an inspiring and disciplined culture, compelling compensation and rewards, and improved career paths.

These moves should be complemented with systematically built IT projectmanagement skills, offering a mix of formal training and action-based learning and potentially creating a center of excellence.

of any IT program and are most critical to retain in-house: IT program manager, business change leader, and lead IT architect.

The IT program manager needs to oversee the project, understand both the business context and the technology involved, and have strong management capabilities. He or she must also be able to talk about technology and business decisions in a language that business managers understand.

The business change leader is responsible for ensuring that the organization adopts the new solution. This role requires strong communication skills and an understanding of the full amplitude of the transformation and its implications for the business side, including required organizational, process, and mind-set changes.

The lead IT architect is responsible for reviewing and challenging technical proposals and deliverables such as solution design and IT architecture. He or she needs to understand the current IT architecture and also have a good view of the transformation journey to ensure that decisions fit with the architecture road map. Focusing on hiring, developing, and retaining people in these three roles offers organizations the greatest return on their talent investment.

Attract talent by improving culture, benefits, and career paths

After defining which roles and skills are needed internally, it becomes paramount to attract, develop, and retain talent. In the McKinsey global survey mentioned earlier, 500 IT and non-IT executives worldwide indicated that the three most important talent levers are culture, benefits, and clear career paths (exhibit).

Improve culture, energy, and morale

Large IT programs are sometimes highly stressful; they can entail considerable overtime, they're met with little appreciation from the broader organization because of the disruption the programs might cause, and they depend on the work quality of others. Having the right culture to overcome these challenges is essential. According to a McKinsey study of organizational archetypes and characteristics of winning organizations, the culture of a large IT program should be built on three pillars.

One is clear direction that inspires employees. Large technology investments have the ability to

Enterprises must recognize the value and scarcity of employees who combine IT savvy with business acumen and must build and support a staff of such people.

Exhibit 1 **To address talent challenges, companies should focus on culture and compensation.**

dramatically improve business performance, but too often the focus on business value is lost in the day-to-day efforts of the project. Frequent town-hall meetings can help to remind teams of the impact of their work.

The second pillar is strong operational discipline; for instance, teams need strict meeting agendas and operational rules. To effectively steer large projects, managers must have access to the right tools. Defining a clear governance structure— for example, using a project "facebook" that makes clear who is responsible for what—helps create transparency and support operational discipline. Objective metrics are important because they allow fact-based communication and problem solving within teams and make it easier for executives to make informed decisions. Program managers must be equipped with tools that cover planning as well as reporting to allow program progress, business impact, and measurement of capabilities. One such tool is a capability scorecard that measures impact and development of skills.

The third pillar is an open and trusting environment that allows risk taking and encourages managers and other employees to admit mistakes. Some organizations deploy a simple web survey to regularly gauge the project team's morale and performance, followed by an open team discussion about the results.

Raise compensation and rewards

Compensation of program managers is often capped by IT salary pay scales and usually doesn't take into account the responsibility that rests on leaders of large IT projects. Companies could put in place flexible measures that would allow them to compensate program managers appropriately, for example, by introducing a new job category. It is also important to link part of the program manager's and his or her team's compensation to the success of the program with respect to its speed, final cost, and the value delivered, while taking into account that large programs bear an inherent risk of significant delays and cost increases. Beyond compensation, other benefits and rewards—such as recognition, time off, flexible schedules, and increased mentorship— are also good ways to motivate employees.

Define career paths

Career paths for leaders of large IT-driven projects are rarely clear or compelling, and they're often nonexistent, which is one reason these leaders are in short supply. Why would a program manager assume the risk of leading a large project without a clear view of career advancement or even an indication of what to do once the project is over? Given the risk inherent in large IT projects, an ambitious manager absorbs significant career risk if the project struggles to deliver. One answer is to build bridges within the organization to allow program managers to go from one entity to another and to progress as a manager. A second is to open career paths between the IT and business sides. Either way, the role of a senior program manager should be seen as a stepping stone to greater responsibility in the broader organization.

Build IT projectmanagement capabilities

Attracting and retaining the right talent is usually not enough; to complement these moves, companies should systematically build IT project-management skills. Focusing on people development not only helps build the team's capabilities but also helps drive attraction and retention. That's why organizations should offer a mix of formal training and action-based learning and consider creating a center of excellence (COE) for program management.

Learning off and on the job

Organizations can provide formal training through either internal curricula or external certifications. The training content should cover leadership skills such as team and change management as well as functional skills such as planning and management of vendors and user requirements.

To get the most out of their large IT-driven business transformations, many organizations should boost their in-house programmanagement talent and IT expertise.

While formal project-management training lays the foundation of required skills, it cannot replace real-world experience. In large-scale multiyear transformations, the early stages of the project are a great time to build the project team's capabilities.

One organization that was struggling to deliver an important IT-enabled business transformation opted to restart the program with a heavy emphasis on building capabilities right from the start. The focus was on the skills that were most important to the success of the large and complex program:

stakeholder management, building a high-performing team, vendor management, and agile software development. Through a combination of formal training sessions, on-the-job practice, and peer coaching, the program built a team that delivered its first major release in less than nine months—quite an accomplishment, given that the program had been running for more than two years without any release before the restart. In addition, approaches pioneered by the program are being rolled out to the broader organization, demonstrating the secondary benefits of capability building for the full organization.

Create a center of excellence

Another way to improve project-management capabilities is to establish a COE— an entity that, for example, can provide leadership, expertise, best practices, support, and training. That was the course taken by one large public organization that had suffered major IT project failures and needed to address important businesssponsorship and talent issues. The organization had been unable to find a single business owner for multiunit projects, and it was short on the right IT talent. As a result, the program-management team relied heavily on external vendors.

The COE centralized program-management talent and senior IT experts. It provided advancement possibilities within the COE for program managers when previously these paths had been flat. In addition, making the COE a separate organization released it from some of the constraints of the public organization's IT salary pay scales, allowing the COE to pay higher salaries. The COE also put in

place tools and processes to manage large projects. For example, all projects went through a review committee, and none could go forward without the right sponsorship

Several factors made this COE a success. Most important were strong senior business sponsors committed to allow the COE to run large strategic projects centrally; the expertise to develop an independent perspective on the project was also critical. COE members should not only be process managers inspecting work but also knowledgeable colleagues offering deep expertise.

■ ■ ■

To get the most out of their large IT-driven business transformations, many organizations should boost their in-house programmanagement talent and IT expertise. The levers range from improved compensation and career paths to formal training and the creation of a COE. Organizations need to understand their own talent issues and be creative about finding the most effective solutions.■

---

[1] For more, including different cuts of the data, see Michael Bloch, Sven Blumberg, and Jürgen Laartz, "Delivering large-scale IT projects on time, on budget, and on value," McKinsey on Business Technology, October 2012, mckinsey.com.
[2] Naufal Khan and Johnson Sikes, "IT under pressure: McKinsey Global Survey results," March 2014, mckinsey.com.

The authors wish to thank Nina Bohlken and Thomas Elsner for their contributions to this article.

Francine Debane (Francine_Debane@McKinsey.com) is an associate principal in McKinsey's Paris office, **Katya Defossez** (Katya_Defossez@McKinsey.com) is a principal in the Houston office, and **Mark McMillan** (Mark_McMillan@McKinsey.com) is a principal in the Washington, DC, office.

Success Stories

Putting digital process innovation at the center of organizational change

When a merger is announced, the IT organization has a unique opportunity to help the company reimagine its technology landscape and clarify its digitization strategy.

Janaki Akella, Neha Gargi, and Tushar Mehrotra

When it comes to mergers, acquisitions, or divestitures, IT professionals are necessarily focused on the immediate tasks of streamlining unwieldy IT systems; given the complexity and sheer size of the integration task, they must be selective about what projects and changes they can tackle within the first 100 days. That shouldn't preclude them, however, from looking a bit further ahead in the integration schedule and identifying how the business could eventually revamp its processes in ways that can create significant long-term value.

IT organizations' initial reaction to mergers and acquisitions has typically been to reduce the scope of change where possible, weather the storm, and resolve as many technical issues as possible by announced "go live" dates. They have tended to focus on maintaining what is there—primarily, costly legacy systems that add a complexity tax to every business activity undertaken. The organization can become mired in a long, frustrating integration process that yields only some (or even none) of the intended benefits of the acquisition or merger.

Instead, IT leaders should take the time to reconsider their IT architectures in the broader context, considering not just the requirements for the first 100 days of the merger but also the potential efficiencies the company may gain over time from digitization as well as the technical solutions the business needs to meet its efficiency objectives. Indeed, IT leaders should systematically

Takeaways

There are three steps organizations can take to encourage digital process innovation after an M&A: streamline end-to-end processes, define how information and data are managed, and reexamine the organizational structure and operating model.

Acquiring companies can increase their revenue from R&D efforts by up to 25 percent when they and their targets take advantage of digital processes.

The companies that can seize the unfrozen moment just before or just after an announced merger or acquisition have an opportunity to realize significant long-term change.

review their options for building a more agile frame around existing and acquired systems. A focus on the implementation of new or enhanced technology-enabled ways of working—or digital process innovation—can help companies simplify the technology landscape, reduce overall IT costs, and bring products and services to market quicker, thereby realizing greater earnings potential.

In this article, we present a three-point framework for encouraging the pursuit of digitization during M&A transitions—namely, by performing end-toend mapping of business activities, assessing and improving data management and analytics expertise, and exploring new organizational roles and operating models. To help illustrate how these activities may be carried out, we consider the potential effects of introducing digital processes in a consolidating pharmaceutical and medical-device market. In our experience, however, the framework discussed here can be applied in companies in any industry facing M&A opportunities and integration decisions.

Potential impact of digital process innovation

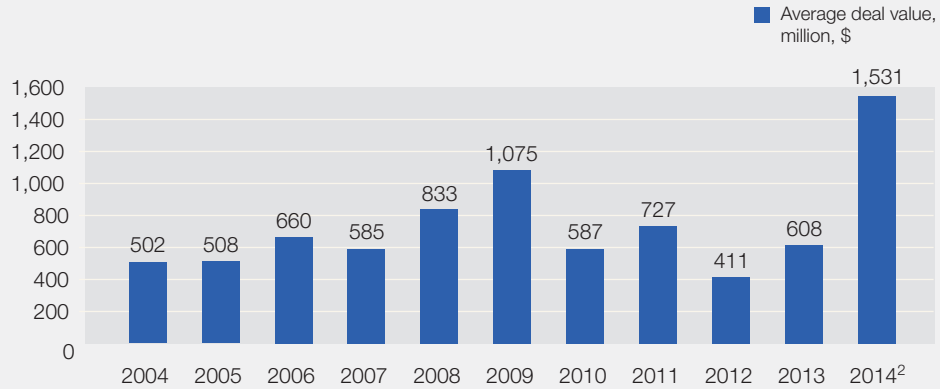
Our research indicates that introducing digital processes and capabilities throughout a company can facilitate acquisitions, mergers, or divestitures. Consider the market for pharmaceuticals and medical products; the average value of M&A deals in this industry is rising as more companies seek to streamline their R&D, supply-chain, sales, and marketing operations while still increasing market share1 (Exhibit 1). However, in this industry and others, successful deals have been hindered by poor cultural fit, poor or slow execution, lack of adequate planning, and a limited understanding of the technologies and digital capabilities each side brings to the table.

One medical-device manufacturer, for instance, acquired a start-up firm, seeking to capitalize on the smaller firm's expertise in technologies for creating digital services for customers. The device manufacturer announced the acquisition and quickly entered into a multiyear rebuilding project. While the integration period was under way, dynamics in the healthcare market continued to change, and by the end of what turned out to be an extended transformation period, the IT goals the company had set at the beginning now seemed outdated given emerging tools and technologies. Rather than capture the full value intended from the deal, the company was facing significant cost overruns and was lagging competitors.

Our research and industry analyses indicate that acquiring companies can increase their revenue from R&D efforts by up to 25 percent when they and their targets can jointly take advantage of digital processes that are enabled by big data and advanced analytics to unearth innovative ideas, make product-development decisions, and collaborate more effectively. They could also increase their revenue from sales and marketing initiatives by launching new or shared digital sales channels. And they could realize a 5 percent increase in revenue attributed

Exhibit 1 **The average value of M&A deals in the pharmaceutical and medical product industry is rising.**

Average deal value and total deal volume[1]



	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014[2]
Sum of total deal value, billion, $	105	96	140	118	160	163	100	132	88	103	191
Number of deals >$1 billion in value	15	16	23	27	16	15	19	14	20	18	23

[1] Includes deals involving biotechnology, medical-device, and pharmaceutical companies.
[2] As of September 17, 2014.
Source: Strategic Transactions Database

to operations by using virtualization, automation, and other digital technologies to create end-to-end visibility and management of supply-chain functions—from warehouse to shipping to delivery.

Enabling digital process innovation

There are three critical actions organizations can take to encourage digital process innovation: map and then streamline select end-to-end business activities, get a clear view of how information and data are managed across the parent and target companies, and reconsider the organizational structure and operating model. Underlying all of these actions is the need for a dual-speed IT architecture—one that preserves existing legacy systems on the back end but also enables quick development of innovative customer-facing applications on the front end2 (see sidebar, "Digitization requires a two-speed architecture").

Streamline select end-to-end processes

In the wake of an announced merger, IT executives should make system optimization decisions that go beyond planning for the go-live date. They should begin to consider which end-to-end processes in both the parent and target companies—for instance, procurement to payment, order to cash, and record to report—would benefit most from streamlining through digitization. They should categorize the applications associated with these processes and determine which technology capabilities and areas of expertise they still need to develop.

Assess business processes. The first step is to identify the company's highest-level business processes to determine where there are standardization opportunities across business units, geographies, and product categories. The medical-device manufacturer mentioned earlier

Digitization requires a two-speed architecture

A dual-speed IT architecture is a prerequisite for digital process innovation. It decouples the management of slower legacy systems on the back end from the development of faster customer-facing applications on the front end. And it allows IT organizations to release innovative digital products and services to customers more frequently without compromising the maintenance and stability of back-end systems.[1]

To implement a two-speed approach, companies first need to consider how their front-end software is developed and deployed— emphasizing cocreation by IT staffers and the business units. Rather than perpetuate a system where business units throw their software requirements over the wall to IT, companies can establish new work flows whereby representatives from each group collaborate in small cross-functional teams to develop new prototypes as well as frequent updates to existing software. Companies may also want to explore cloud-based infrastructures that can speed up this development process and facilitate sharing.

A move toward two-speed architecture can take time but even a gradual shift to this model can help companies digitize business processes at a relatively quick pace. The period just before or just after an announced merger or acquisition provides an ideal opportunity for companies to look at their existing IT architectures and consider whether there is a significant gap between backend and front-end processes, and how best to close that gap.

[1] Oliver Bossert, Chris Ip, and Jürgen Laartz, "A two-speed IT architecture for the digital enterprise," December 2014, mckinsey.com.

had targeted its R&D processes for reinvention; with the acquisition of the small start-up, the company wanted to launch products more quickly and incorporate customer feedback into the development process. IT executives sought to engage multidisciplinary teams that could quickly integrate new technologies (Internet of Things, 3-D printing, and so on) into the organization. This meant creating new, more agile ways of working.

A mapping exercise can help companies see process flows across business functions and spot any gaps in their technology capabilities. Organizations that prioritize supply-chain operations, for instance, may clearly see a need to introduce automation technologies to facilitate their sales and distribution planning as well as their transportation and warehousing activities—for instance, automating the collection and analysis of data to identify which size trucks are required for certain deliveries, where particular items are stored within a warehouse, and what the best storage options are. Or, they may seek to build up their expertise in advanced (and sometimes proprietary) analytics and algorithms to aid demand management—for example, generating sales forecasts and identifying purchase patterns. In those business functions in which a company's technology capabilities are more advanced, there may be opportunities to cut costs through standardization—for instance, there may be redundant processes being used within supply-chain and finance functions.

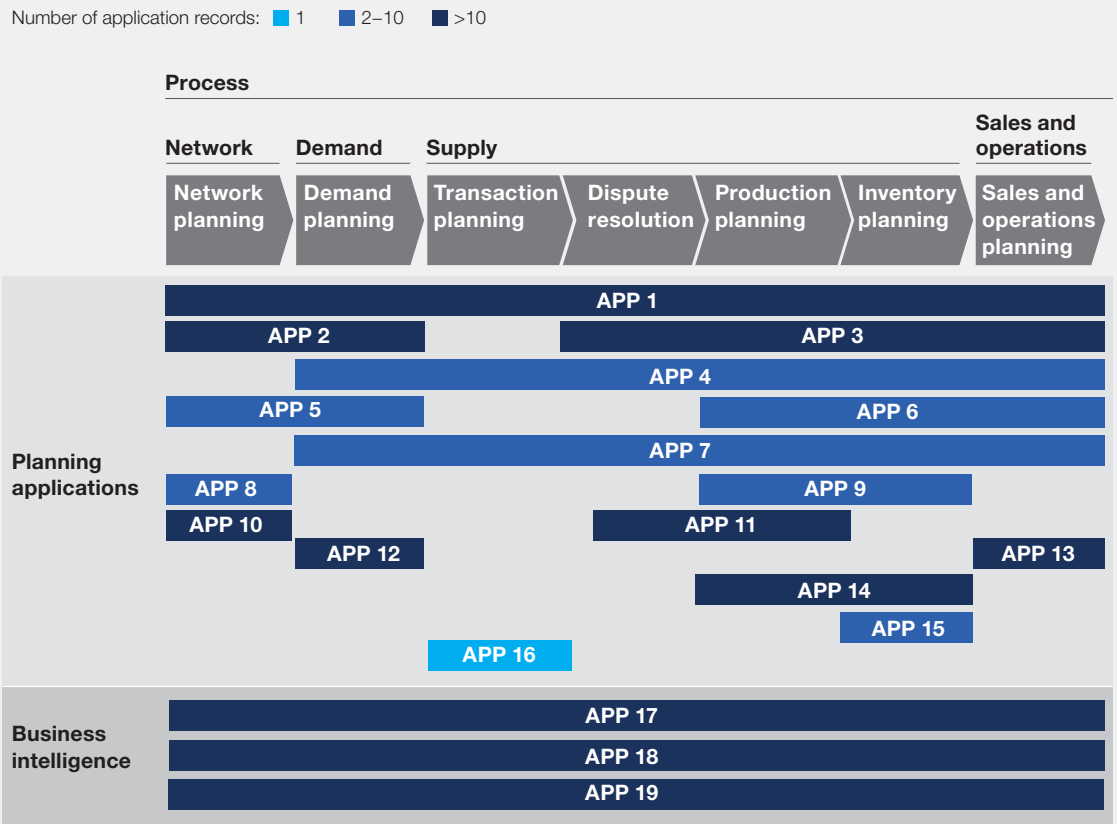
Assess business applications. The second step is for companies to evaluate their software applications—those of the target company as well as those of the acquiring firm. Mapping these applications to individual activities, within different business functions and divisions, allows IT executives to easily spot overlaps and redundancies in their application portfolios. One planning application, for instance, might be relevant across a number of functional areas while another may be relevant only for transportation planning or only for inventory planning (Exhibit 2). The period just before or just after the announcement of a merger is a good

time to take this inventory; both companies have an opportunity to determine how to reconcile and simplify their individual IT landscapes before figuring out how to actually make them work together. Indeed, applications should not be modified, retired, or rationalized without a full understanding of why the companies' IT architectures are configured the way they are. To that end, IT organizations on both sides of the deal will need to capture relevant information about their applications, such as total number of apps, total cost of ownership, usage metrics, and application ownership (which divisions in which locations). Most critical is maintaining a single, consolidated view of this information.

Get a clear view of how information and data are managed

To ensure that everyone has access to the information required to pursue digital process innovations, companies must establish a single system of record across all functions, divisions, and geographies. The system must comprise all relevant data—for instance, sales, inventory, and procurement records—plus the latest analytics tools and a reliable global delivery service, all maintained by a dedicated team of knowledge professionals. The medical-devices company, for example, created a center of excellence for data management, which included representatives from the business units and the IT organization. The company also

Exhibit 2 **Mapping business applications to business processes provides a clear accounting of merging firms' digital assets.**



hired analytics experts to provide an empirical perspective on which data should be collected, how they should be collected and disseminated, and how databases should be curated and managed over time. From a hardware standpoint, organizations will likely need to repurpose their existing data-management systems, pursue standardization options where appropriate, and develop a rigorous strategy for creating, using, and storing data as well as related governance policies. Again, a steering committee or center of excellence can play a central role in setting these policies.

Reconsider the organizational structure and operating model

It will be critical for companies seeking to digitize select processes to create a new management structure and operating model with several new roles. The most critical role is the global process manager or global process owner. This individual will be charged with overseeing process standardization across divisions, regions, countries, and product lines. He or she will be well positioned to provide an end-to-end view of digital processes and a framework for the implementation of any new step changes. He or she will also work closely with business and IT leaders, and division-specific program managers to understand "local" needs as well as "global" needs. Indeed, the medicalproducts manufacturer we described earlier ended up creating two other management positions to collaborate with its newly appointed global process manager. The business-relationship manager acts as a liaison between the IT organization and the business side and is charged primarily with defining potential digital process innovations at the division level. The IT-delivery leader is focused on execution; he is responsible for ensuring that any process innovations are properly rolled out across the organization.

■ ■ ■

To incorporate digital process innovation into their operations, IT executives will need to change some of the prevailing behaviors and mind-sets in their companies. Apart from creating a snapshot of current technology capabilities, IT executives will need to get buy-in for digital process innovations from the business side; they will need to demonstrate how the pursuit of new, technology-enabled ways of working is in direct service to desired business outcomes. As mentioned earlier, they may need to configure teams differently—not just by creating new roles, such as the global process manager, but also by reimagining existing roles in business, IT, and product-development functions. Above all else, transparency is required—among team members and in all the data captured and reported.

The companies that can look beyond the first 100 days of an announced merger or acquisition, when everything in the IT architecture is up for discussion and refinement, have an opportunity to realize significant, lasting value for the organization. ■

---

[1] Martin Hirt and Paul Willmott, "Strategic principles for competing in the digital age," McKinsey Quarterly, May 2014, mckinsey.com.
[2] Henrik Andersson and Philip Tuddenham, "Reinventing IT to support digitization," May 2014, mckinsey.com.
[3] The consumer decision journey refers to the process by which consumers research and buy products. It is a circular journey encompassing four phases: initial consideration, active evaluation, closure, and postpurchase.
[4] Oliver Bossert, Chris Ip, and Jürgen Laartz, "A two-speed IT architecture for the digital enterprise," McKinsey on Business Technology, November 2014, mckinsey.com.
[5] Ibid.
[6] Michael Huskins, James Kaplan, and Krish Krishnakanthan, "Enhancing the efficiency and effectiveness of application development," August 2013, mckinsey.com.

The authors wish to thank Adesh Ramchandran and Roger Roberts for their contributions to this article.

Janaki Akella is a principal in McKinsey's Silicon Valley office; **Neha Gargi** is an associate principal in the Chicago office; and **Tushar Mehrotra** is a consultant in the Washington, DC, office.

From waterfall to agile: How a public-sector agency successfully changed its system development approach to become digital

By switching its development methods midstream, the Danish Business Authority was able to launch an online company-registration system faster and achieve better results.

Martin Lundqvist Peter Braad Olesen

Government agencies around the world are under internal and external pressure to become more efficient by incorporating digital technologies and processes into their day-to-day operations.

For a lot of public-sector organizations, however, the digital transformation has been bumpy. In many cases, agencies are trying to streamline and automate workflow and processes using antiquated systems-development approaches. Such methods make direct connections between citizens and governments over the Internet more difficult. They also prevent IT organizations from quickly adapting to ever-changing systems requirements or easily combining information from disparate systems. Despite the emergence, over the past decade, of a number of productivity-enhancing technologies, many government institutions continue to cling to old, familiar ways of developing new processes and systems.

Nonetheless, a few have been able to change mind-sets internally, shed outdated approaches to improving processes and developing systems, and build new ones. Critically, they have embraced newer techniques, such as agile development, and succeeded in accelerating the digital transformation in core areas of their operations.

The Danish Business Authority is one of those organizations. This agency is charged with registering corporations that do business in Denmark. With the world economy teetering in 2009, it decided it could no longer maintain a largely manual registration process. It believed that replacing paper forms sent by mail with a simple online process was crucial to keeping the country economically vibrant. Specifically, a new digitalregistration process would show both domestic and foreign companies that it was easy to do business in Denmark, help track money laundering, and better identify companies that didn't report their income or pay taxes on it.

The Danish Business Authority set a goal of completing the specifications of a digital-registration system by 2011 so software developers could begin their programming efforts and formally roll out the streamlined process by 2014. In the first two years of the initiative, the agency used the traditional "waterfall" approach to design and development. But the effort stalled for a number of reasons, including ever-changing systems requirements and slow decision making.

In 2011, the IT organization revised its launch date and decided to trade the waterfall approach for an agile approach to systems development; a change in leadership was the catalyst. The agile approach has several hallmarks. New systems requirements can be accommodated late in the development process. It's possible to deliver the software for parts of a system early, even before all the requirements are completely understood, to break design logjams. And decisions can be made more quickly if companies have only one team of businesspeople and software developers, rather than throwing requirements "over the wall" between functions and thereby facilitating divisiveness.

By 2014, the system was nearly completed. By 2015, the number of registrations requiring agency support for completion had dropped from 70 percent to 30 percent. More broadly, the registration system has helped Denmark to rank high on the World Bank's annual index of digital government services and on another index that rates the efforts of European countries in helping new companies launch their businesses.

In this article, we consider the digital transformation of the agency: the challenges it faced in moving away from the waterfall methodology; the change-management principles it followed as it incorporated agile technologies, processes, and

Helping to make Denmark's economy click

The Danish Business Authority employs 600 people, who institute policies set by the country's Ministry of Business & Growth. Known in Denmark as ERST (short for its Danish name of Erhvervsstyrelsen), the agency keeps track of companies operating in Denmark, regulates accounting and auditing rules, and monitors fund transfers in the financial system (in particular, to spot criminal conduct associated with money laundering). It finances economic development in Denmark, promotes the formation of new businesses, and keeps data on business activity.

In addition, the authority regulates the telecommunications industry—administering the allocation of wireless spectrum, maintaining competition among telecommunications companies and Internet service providers, and working with other Danish government agencies on policies. In fact, the Danish Business Authority is an independent regulatory body on telecommunications, and its rulings cannot be appealed. Just as important, the authority builds digital systems that automate communications between businesses and Danish government authorities.

mind-sets into its traditional ways of working; and the results it has been able to achieve thus far. The agency's story provides important lessons for government agencies everywhere that need to build critical digital systems.

Initiating the digital transformation

In 2009, the Danish Business Authority (see sidebar, "Helping to make Denmark's economy click") decided that its paper-based process for registering new companies had to be replaced by a digital one. New businesses often required weeks or even months to complete registration: they had to request the right forms, fill them out, and submit them through land mail. When forms arrived at the authority, workers had to enter data from each one into as many as 14 different computer systems. The agency recognized that this manual process was slow and error prone.

When the agency received the go-ahead to build a new corporate-registration system, it started with the waterfall method, a systems-development approach it had used for years. Under this approach, software developers go through discrete phases, starting with gathering the business requirements. Then they proceed to process analysis, including establishing the business rules that will inform the design of the system—for example, "treat this type of company and that type of company in different ways." The next step is software design and programming. A core principle of the waterfall development approach is that a team using it can't move to any phase without ironing out the details of the preceding one. There is good reason for that: fuzzy up-front requirements can introduce big software-design and -coding problems later.

In the case of the Danish Business Authority, this limitation meant that coding on the new registration system couldn't begin until the process analysis was finished—and the development team couldn't agree on the minutia of a standard process for registering a new company. Additionally, team members were involved only part-time on the project; each had other systems priorities. That contributed to long decision-making cycles, even for minor issues. Big decisions could take weeks to make, largely because the team gathering the requirements couldn't get the attention of senior managers quickly. Plus, numerous system-design decisions had to be made collaboratively, and assembling the relevant business managers to make them was trying.

The result was analysis paralysis.

Exploring agile tools and methodologies

Around the time the system project stalled, the Danish Business Authority had brought in a new director general. One of her first priorities was getting the development of the digital-registration system back on track. After a comprehensive review of the program, the director general and the leadership team concluded that what it needed most was a design and development method that emphasized building and testing new systems in weeks rather than months and incorporating input from "customers," external and internal alike.

An agile approach seemed more appropriate than the waterfall model, the team decided. For one thing, the agile methodology would allow developers to incorporate new system requirements into their design and planning late in the process. It would prompt them to deliver software code early, even before all of the requirements were understood. And it would bring businesspeople and software developers together on a single team, rather than having one try to decipher the other's input after it had been tossed over the wall. Discrete project components could be designed and coded in weeks (or in "sprints," as they were called) rather than months or years.

Rebooting the systems-development process

At the time of its reboot, the Danish Business Authority had found few government institutions, in or outside Denmark, that had used agile methods to develop core digital systems. As a result, the team created its own flavor of agile, based on seven critical elements:

1. a focus on the customer;
2. strong governance and swift decision making;
3. an IT architecture that enables gradual changes in the system;
4. a clear systems-development road map comprising a number of small, manageable projects;
5. an organization that embraces agile and the processes supporting it;
6. outsourcing to multiple partners rather than just one or two;
7. and a culture of trust.

Here's how each element played out in the Danish Business Authority's digital transformation.

A customer-centric focus. Emphasizing customer needs gave the project's team members clearer priorities and a common vocabulary. Initially, the program had focused largely on the agency's own registration requirements and less on those of the From waterfall to agile 5 businesses that would be using the system. That all changed under the agency's agile development approach. For example, rather than forcing businesses to enter registration information into 14 different systems, the agency designed just one. That move alone saves businesses considerable time in registering companies—and it saves the Danish Business Authority a significant number of personhours, as well.

Strong governance and swift decision making. The digital-registration initiative was mission critical but wasn't treated that way until 2011. The CIO, for example, had been responsible not only for this project but also for a number of other IT priorities. In 2011, some of them were moved to a different department, so that the CIO could give more time and attention to the digital initiative. The CEO also became part of the daily projectgovernance team. This change caught the attention of other senior executives, who then recognized that they, too, had to make the system part of their agenda. Meetings to discuss the project's progress became weekly events, which enabled the multitude of subprojects to stay on track because issues were monitored continually. What's more, these weekly sessions—chaired by the CEO—let the project team bring outstanding issues to the table and have them resolved much more quickly.

The stronger governance extended beyond the CEO's weekly meetings. After the big project was divided up into more than 30 smaller subprojects, the teams in charge of each of them received the authority to make decisions affecting their own areas. That meant they didn't have to wait for answers, which had caused big delays in the first round of development. Further, each team included IT and "product owners"—people from the Danish Business Authority's business operations—as well as representatives from the agency's vendors. Project teams had the authority to make decisions on their system components quickly because someone from the business operations was there to finalize the choices.

A flexible IT architecture. The architecture of an information system stipulates where data will reside (how many databases), which software components will be shared across applications and which will not (such as user interfaces), and other technical details. A good architecture saves programmers time writing new code (by reusing common components), reduces errors, and defines system components so that they can be updated or replaced without any need to scrap the whole system. A poor architecture does the opposite: it piles on rework, introduces errors, and makes systems difficult to change.

The IT architecture for the Danish Business Authority's new registration system divided it into more than 30 components. New features could therefore be implemented and launched piecemeal rather than all at once, which reduced the risk and complexity of implementing system changes. The architecture also called for only a single database, which eliminated the requirement that the agency's people manually reenter information from one database to the next. This would be the

sole "source of truth," with all the information on a company stored in one place. What's more, in the new architecture, all users—both internal and external (Danish businesses)—would see the same interface. Additionally, the architecture required the development teams to share software components, so that they didn't have to reinvent the wheel. For example, the software for looking up a registrant's business address was a shared component.

A clear systems-development road map. Traditional approaches to systems development emphasize getting all the requirements up front before any coding begins. By contrast, the agile approach emphasizes breaking up big systems into smaller components, which can be built and implemented one piece at a time and brought to market quickly. The key is building components in ways that minimize interdependencies among them. As mentioned, the Danish Business Authority broke its registration system into many smaller 6 From waterfall to agile system projects. To track them, the management team created a clear road map showing when each would go live and its relationships with the others. That helped teams to know what their colleagues were doing, and when. Elements of the registration process were improved and launched sooner than they would have under the waterfall process. Additionally, the project teams that hadn't delivered their modules could learn from the early releases and modify their plans accordingly.

An agile organization and processes. The Danish Business Authority followed standard practices in structuring each project team, which included business and IT professionals, as well as vendor staff. The teams were located in the same place, which dramatically reduced the chances for miscommunication, since it allowed each side to speak up early and often. The teams followed agile practices by asking the software-development vendors not to create specific requirements for their pieces of the system, the traditional approach. Instead, the vendors were asked to create user stories—denoting, for instance, what the system should allow the user to do at a general level. The programmers had more room to interpret the design and determine how to turn it into code.

Outsourcing development to multiple partners. Unlike many organizations, which outsource their software development to one IT-services company, the Danish Business Authority used four. The idea was to provide appropriate incentives and to promote more competition among vendors, which understood that future components would go to companies that had completed earlier ones on time, on budget, and on target. The setup even encouraged some vendors to overinvest so they could secure the next module. It also gave them an incentive to make their components easy to maintain and expand—if they did good work, there was a good chance they'd be the ones maintaining and expanding what they had built.

A culture of trust. Such incentives and ways of working—not asking vendors to collect detailed specifications before coding, creating teams of both business and IT people, and so on—created an atmosphere of trust and collaboration. The trust increased rapidly because the authority asked vendors to deliver their pieces of the system in 14 days. Mistrust didn't have time to grow; each vendor could prove its competence quickly. By breaking the project into smaller pieces that could be turned around quickly, the authority enforced better performance from its vendors. It replaced underperforming teams or team members rapidly, in this way setting a tone that, just as high performance would be rewarded with more work, underperformance would be punished with less.

This culture of trust extended beyond the development of the digital-registration program; it was a critical element in the rollout of agile ways of working in other areas of the authority, as well. The digital-registration project team helped to provide support for the widespread adoption of agile techniques—for instance, serving as agile-implementation coaches and offering process demos and training on agile principles. That

promoted trust in the methodology and, therefore, change across the organization.

Launching a new corporate-registration system

The agile approach to systems development put the digital business-registration project back on track. Many pieces of the system are live; the remaining modules are slated for release. The Danish Business Authority is already seeing some significant benefits:

- Less customer hand-holding. The average time needed to resolve a customer's problems over the phone has dropped from 16 minutes to 5 minutes. The number of customer-support From waterfall to agile 7 calls after registration has dropped from 70 percent of applications needing phone support in 2009 to only 30 percent today.
- Less need for rework by customers. The number of registrations completed accurately the first time around by businesses has risen to 92 percent, from 80 percent.
- Less time needed to get new employees ramped up. Because of the improved usability of the new digital system and because it automates much of the work employees had to do, the authority has reduced the time it takes to train new employees by 80 percent, from five months to one month.

This system has helped Denmark attract new companies. The World Bank compiles an index on the ease of doing business in 189 economies. In the 2016 list, Denmark ranked third overall, trailing only Singapore and New Zealand.1 A frequent user of the registration system, who works at one of Denmark's largest law firms, estimates that she spends 50 percent less time on registration tasks and on updating corporate information, such as the names of new board members.

More broadly, the Danish Business Authority has significantly improved its ability to provide digital solutions in other areas. It has increased its IT productivity by about 60 percent, for instance, and significantly improved its time to market for new digital services (exhibit).

Exhibit **It is possible to deploy new code on a site within an hour.**

Tasks	Before	After
Number of process flows	Multiple combinations of 100+ screens	13 to 15 screens
Creation	11 screens	1 process
Journaling	4 or 5 screens	1 process
Case types	More than 100, manually chosen	Automated creation of cases
Working with registrations	Entry, validation of information from paper documents	Validation of information entered by end user
Follow-up on documentation	Obtain missing documents	Control attached documents
Support of digital reporting	Multiple systems	1 centralized system
Training in journaling	2 to 3 months	1 or 2 days
Training in case work (per area)	4 to 6 months	1 or 2 weeks

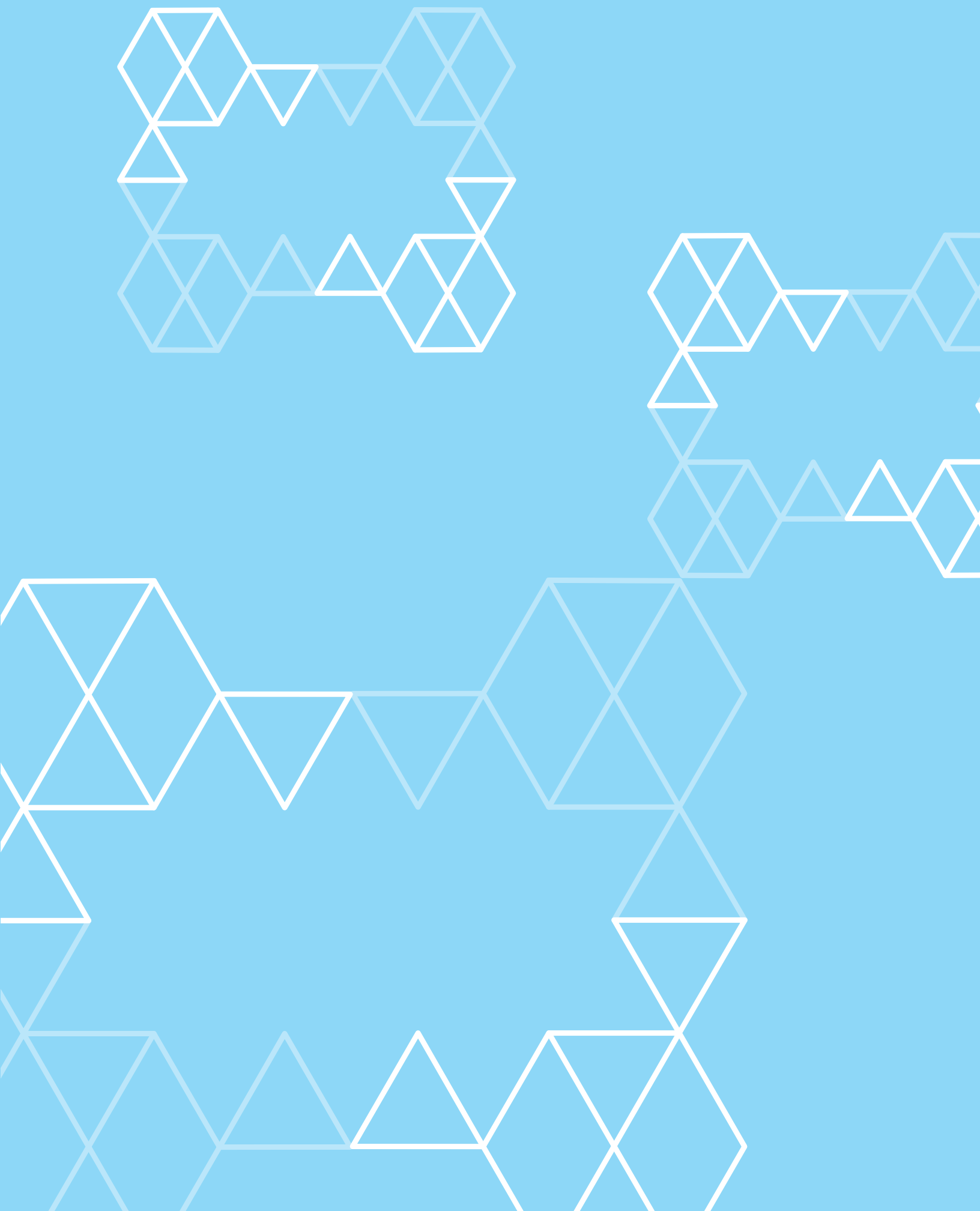
Source: McKinsey analysis

Denmark's digital business-registration system is burnishing the country's reputation as a place to launch new businesses. For example, it is helping the Danish government with the Start-up Denmark initiative—a campaign to attract new high-tech firms that can keep the country economically vibrant and serve as a magnet for jobs and talent.

By using an agile approach to building a digitalregistration system, the agency has been able not only to streamline a critical service relatively quickly but also to provide proof of Denmark's commitment to digital technologies and approaches. Indeed, government executives believe that doing business digitally has become critical to attracting the next wave of digital and nondigital businesses.

1 World Bank Group, doingbusiness.org.

Martin Lundqvist is a principal in McKinsey's Stockholm office, and **Peter Braad Olesen** is an associate principal in the Copenhagen office



Becoming a digital bank

Chief technology officer Mike Murphy discusses Standard Bank's shift to agile software development and how it has helped improve service delivery and internal processes.

Sven Blumberg and Christian Stüer

Several years ago, Standard Bank, one of South Africa's largest and oldest financial-services groups, found itself facing significant challenges from digital competitors. These banks were operating at a much lower cost while still offering customers innovative products and engaging experiences. "We felt like we were investing in all the right technologies, but we didn't have the right processes in place to get the most from those technologies," recalls Mike Murphy, chief technology officer and head of Group Technology Build for Standard Bank.

As a result, the 154-year-old bank embarked on a multiyear digital transformation. The centerpiece of this plan was a shift to agile software development, an approach that emphasizes quick product iteration, test-andlearn approaches, and frequent collaboration among teams. In this interview with McKinsey's Sven Blumberg and Christian Stüer, Murphy talks about financial institutions' increasing need for and reliance on digital channels, as well as the new technologies and strategies Standard Bank has deployed to position itself for long-term business development and revenue growth.

McKinsey on Business Technology: *Most global banks are pursuing some form of digital or mobile strategy—why is it so critical?*

Mike Murphy: We're seeing a new, younger generation of consumers who are technologically savvy and highly adapted to the online world. This is particularly true in emerging markets, where 90 percent of the population is under age 30. In Africa, for instance, nearly half the population is under 19, and most young adults have mobile phones. It used to be that banks were always chasing "credible" digital-banking solutions. Now those solutions are here. We're already seeing banks differentiate themselves through digital innovation. They are

Takeaways

Under the guidance of chief technology officer Mike Murphy, Standard Bank is shifting to agile software development as an important part of a multiyear digital transformation.

Agile software development is an approach that emphasizes quick product iteration, test-and-learn approaches, and frequent collaboration among teams.

"To get started, we narrowed down our agile initiative to focus on two things: how to streamline our software-development processes and how to gain buy-in among staff members," Murphy says. Now the bank is releasing products and services faster, with less rework and fewer errors—and with more cross-functional communication.

offering customers single-click access to loan applications and account information. They are tailoring their products to individual customers' needs. In particular, mobile adoption of these kinds of services will be breathtakingly fast, and no one wants to be left behind.

McKinsey on Business Technology: *What role can the IT organization play in facilitating digital and mobile innovation?*

Mike Murphy: Most banks just can't match the customer experiences provided by pure-play Internet companies—but it's not for lack of trying. It's more about arriving late to the game. IT has a huge role to play in helping companies get up to speed, but only if it can move beyond serving as the executor of business specifications. Look at the IT function in pure-play Internet companies. You see that IT is more of a partner with the business and with the user community, codeveloping software applications with these stakeholders. You see more automation across software development and delivery in those companies. And you see empowered IT developers who can respond quickly to changing customer needs and desires, instead of requiring consensus every time there is a feature-deployment decision. Overall, you see more of an agile approach to software development.

McKinsey on Business Technology: *How did Standard Bank decide to switch to agile software development?*

Mike Murphy: Our ADM [application development and maintenance] group had been using traditional approaches to software development, including waterfall. But these weren't fast enough. We saw what was happening in Silicon Valley and elsewhere, and we talked with executives in other banks and in other industries, and we wanted to capture the same performance and cost advantages those companies were. So we examined how digital-native companies were doing agile software development. We weighed the trade-offs between adapting some of those companies' best practices and the specific requirements in our organization. Our initial focus was on applying agile approaches to mobile applications and Internet-banking software, two areas that the business side was particularly excited about. Now we're using agile across a broad section of the ADM organization, covering 15 product-management teams and comprising a total of 150 full-time employees. By the end of 2016, we're aiming to have the remainder of the ADM organization using agile development.

McKinsey on Business Technology: *How have your software-development processes changed?*

Mike Murphy: Overall, there is much more emphasis on collaboration and co-location. Every product-management team has a product owner, developers, and testers, and they all sit together. Any number of individual product teams might work together depending on product requirements—so if a new application needs to be made compatible with an existing one, team members may come

Mike Murphy

Vital statistics
Born in 1968, in Johannesburg, South Africa
Married, with 2 children

Education
Holds a bachelor of commerce degree from University of the Witwatersrand

Career highlights
Standard Bank
(October 2015–present) Chief technology officer

(April 2015–present) Head, Group Technology Build

(2014–April 2015) Head, Group Technology Operations

2007–14) Head, Group Technology Infrastructure

(1992–2007 Various technical and management roles in IT (corporate- and investmentbanking units)

Fast facts
Certified associate of Institute of Bankers in South Africa

together to create a plan for synchronizing the two. We emphasize with business leaders and with IT leaders this idea of codevelopment and joint problem solving. That way, we can accurately capture the business requirements for the software application or service feature and build in accountability from all parties involved. We schedule lots of discussions about prototypes so both sides can refine their ideas—constantly clarifying the requirements while testing the software. We invite customers into the process, often before a single line of code has been written. And we build informal relationships with customers through online forums and social-media interactions rather than formal focus groups. If a customer proposes a new feature, we will share early mock-ups with the person to get input. Our testers are fully involved at all stages of product development, not just at the end, so they have a much better understanding of what the software is trying to do. They're not only flagging errors but also finding new solutions to problems. And we now use automated testing to speed up what used to be a very labor-intensive process.

McKinsey on Business Technology: *How did you gain buy-in for this approach?*

Mike Murphy: This was one of the toughest challenges. A lot of staffers at the bank were comfortable with the ways things were. They didn't want to change their daily routines. They were focused on simply getting the job done. We broke people out of those comfort zones by making agile a top priority for the whole ADM unit. We convened town halls for product-management teams, explaining the logic behind the change and setting explicit targets for improvement. We reinforced these messages with formal mechanisms—for instance, by co-locating workteam members, so they would collaborate more. We gave teams autonomy to make decisions on how to go about their day-to-day functions, but we did ask that they schedule regular team meetings to provide everyone with status updates, set a maximum length of time for sprint activities, and hold retrospectives regularly to discuss what teams might want to do differently during the next product sprint. People have responded favorably to this sort of loose-tight structure.

McKinsey on Business Technology: *How have your talent-management practices changed?*

Mike Murphy: Cost pressures definitely put a limit on the type and number of new hires we could bring on board. So we've focused on training existing employees in agile techniques and principles. We rely on the "train the trainer" concept—colleagues who attend trainings and workshops, or who visit with Silicon Valley companies, are expected to formally pass their knowledge and observations to their own teams. Each team member goes through multiday training on agile software development; we also have a dedicated group of agile coaches who are always reaching out to product groups. They might sit in on retrospectives and suggest ways the team could improve its practices. Most important, we have instituted a number of "interest groups" or guilds within the application-development unit. These groups meet every two weeks or so to encourage knowledge exchange—there is an iOS guild, a cloud guild, and so on. This is a terrific way to ensure cross-team and cross-functional communication.

McKinsey on Business Technology: *What does all of this change look like on the ground?*

Mike Murphy: The way that business groups interact with IT has changed significantly. I'll give you an example: a tablet application we developed before we moved to agile took 2,500 pages of documentation to explain. Many of those pages contained duplicate information about requirements and, after all that, *still* left the developers struggling to understand what the outcome should be. Now when we develop tablet applications, we rely on a few refined use cases that were cocreated by the IT group, the business, userinterface experts, and end customers. The business feels more in control, and the IT group is no longer operating in the dark. In fact, the IT group feels more empowered under this model. The team can release new online features every month, and because it is incorporating customer feedback into products early on, the amount of rework required has dropped significantly. Before agile, our developers might log thousands of defects relating to a new app and post a 38 percent testing-failure rate. After agile, those numbers are more like 100 defects and a 3 percent failure rate. The overall development process has become much more efficient. And there is more trust among colleagues and functions. In the past, some basic tests would be run twice, once by developers and once by testers who did not trust the developers' results. Today, testers immediately accept developers' results and can move on to more complex tests.

McKinsey on Business Technology: *What challenges have you faced in making all these changes, and how have you addressed them*

Mike Murphy: I can't overstate the importance of breaking down silos and breaking people out of their comfort zones, particularly when you're talking about established companies in established industries such as banking. It's one thing to state a desire to adopt agile development; it's another thing completely to get buy-in from the business units. We learned this early on. We had to put a halt to one of our first pilot projects, because the leaders in affected business units were uncomfortable with the amount of time and resources that were being taken away from day-to-day IT operations and support.

In theory, the business leaders understood the potential benefits of agile. But when we took initial steps in that direction, they realized just what was involved and how reluctant they were to compromise stability for the sake of innovation.

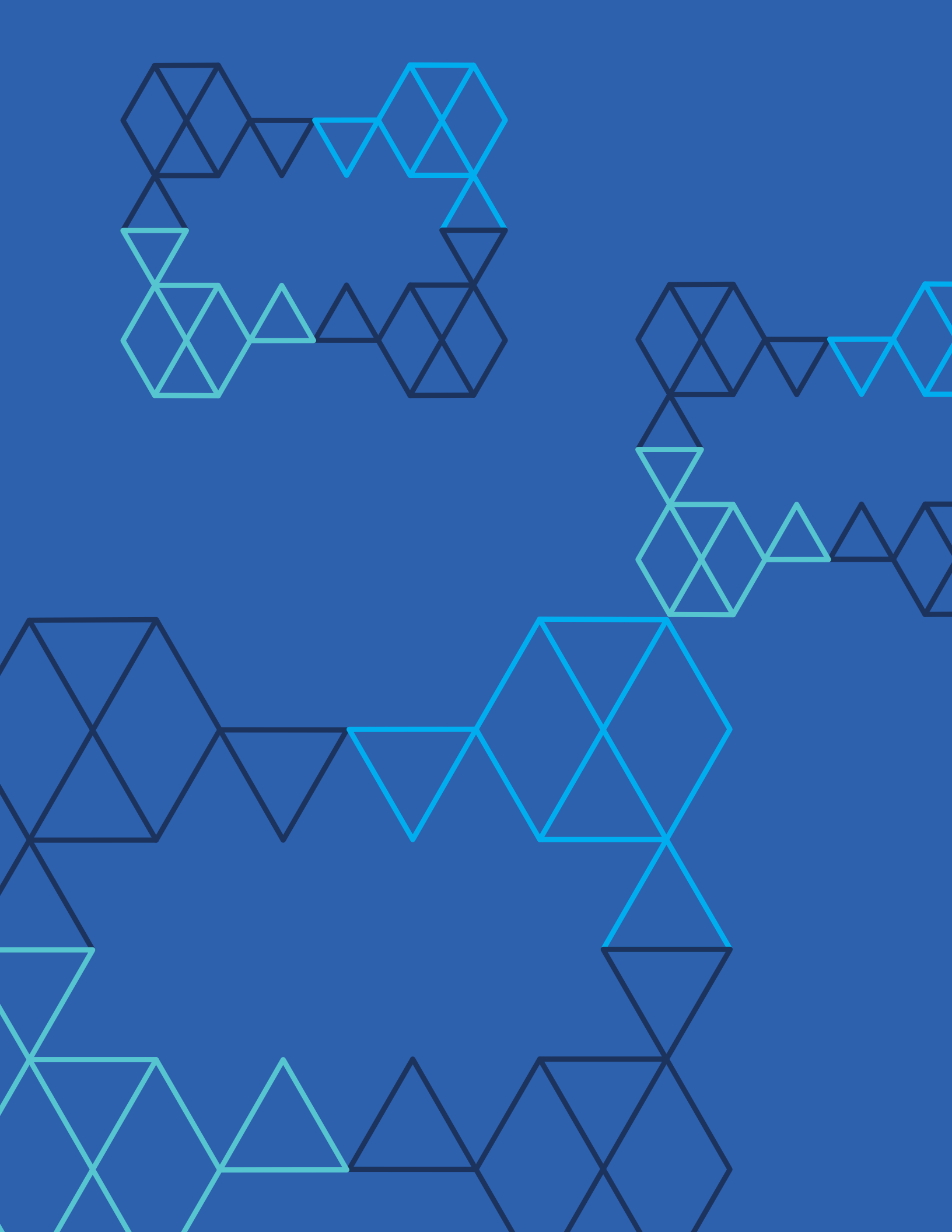
We knew they would need to see positive proof, quickly. To get everyone on the same page, we created new communication channels—a newsletter, a web page, and regular town-hall meetings—for presenting our successes and challenges in agile product development and delivery. These forums have helped build team motivation and sustain the momentum for change. We now have a number of "agile evangelists" among business-unit leaders.

Likewise, our participation in industry events has helped our team members understand where we are with our digital journey compared with our peers—in many cases, we are quite far ahead, and that piece of knowledge can be very motivating.

McKinsey on Business Technology: *What can other companies take away from your experience?*

Mike Murphy: Our first pilots have demonstrated tangible benefits—productivity increases of up to 50 percent and unit-cost reductions of up to 70 percent per function point. But, you know, we're still early in this transformation. We face a number of challenges—how to scale agile to teams outside ADM, for instance, and how to ensure that our IT architecture, infrastructure, and operating model evolve as the digital opportunities do. This requires a huge change in mind-set and organizational planning—for instance, figuring out how to break up teams of 80-plus people into smaller groups focused on specific application features or customer experiences and determining who the dedicated product owners from the business should be. We're learning to reconcile the trade-offs between respecting individual teams' autonomy and accepting varying quality levels across product teams, given the different ways they choose to work. We also have the long-running challenge of any change effort—how do we make it stick? The most important thing for us, or for any company moving to agile, is to remember that this isn't just about reducing costs. It's about streamlining the way we work and delivering the best services to our customers through whatever channel they choose to interact with us.■

Sven Blumberg (Sven_Blumberg@McKinsey.com) is a principal in McKinsey's Düsseldorf office, and **Christian Stüer** (Christian_Stueer@McKinsey.com) is an associate principal in the Cologne office.



Digital/McKinsey

August 2016

Design contact
www.mckinsey.com