



© Todd Arena/Getty Images

## BUSINESS TECHNOLOGY OFFICE

# 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 product- and 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.<sup>1</sup> 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

---

## 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.

---

impediment is their existing operating models and organizational 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.<sup>2</sup>

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

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

	Before	After
<b>Organizational structure</b>	<b>Application-oriented focus</b> , with ever-changing teams and pooled resources; siloed perspective	<b>Product-based focus</b> with stable teams and dedicated resources; end-to-end perspective
<b>Interactions between business and IT</b>	Development process is managed by <b>proxy product owner</b> from IT, with input as needed from business	Development process is managed by <b>strong product owner</b> from business, who works closely with IT at all stages
<b>Roles and responsibilities</b>	Scrum teams comprise <b>developers and testers</b> ; project-manager and line-manager roles remain unchanged from waterfall approach	<b>All roles are integrated</b> within self-organizing scrum teams; project-manager role is minimized and line managers focus on capability building
<b>Budgeting and planning</b>	Traditional <b>yearly budgeting</b> , with fixed budget allocated to projects	<b>Venture-capital-style budgeting</b> , 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 customer-onboarding 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 project-management 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 product-development 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 application-development 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 self-organizing, 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-blanc 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 low-potential projects early.

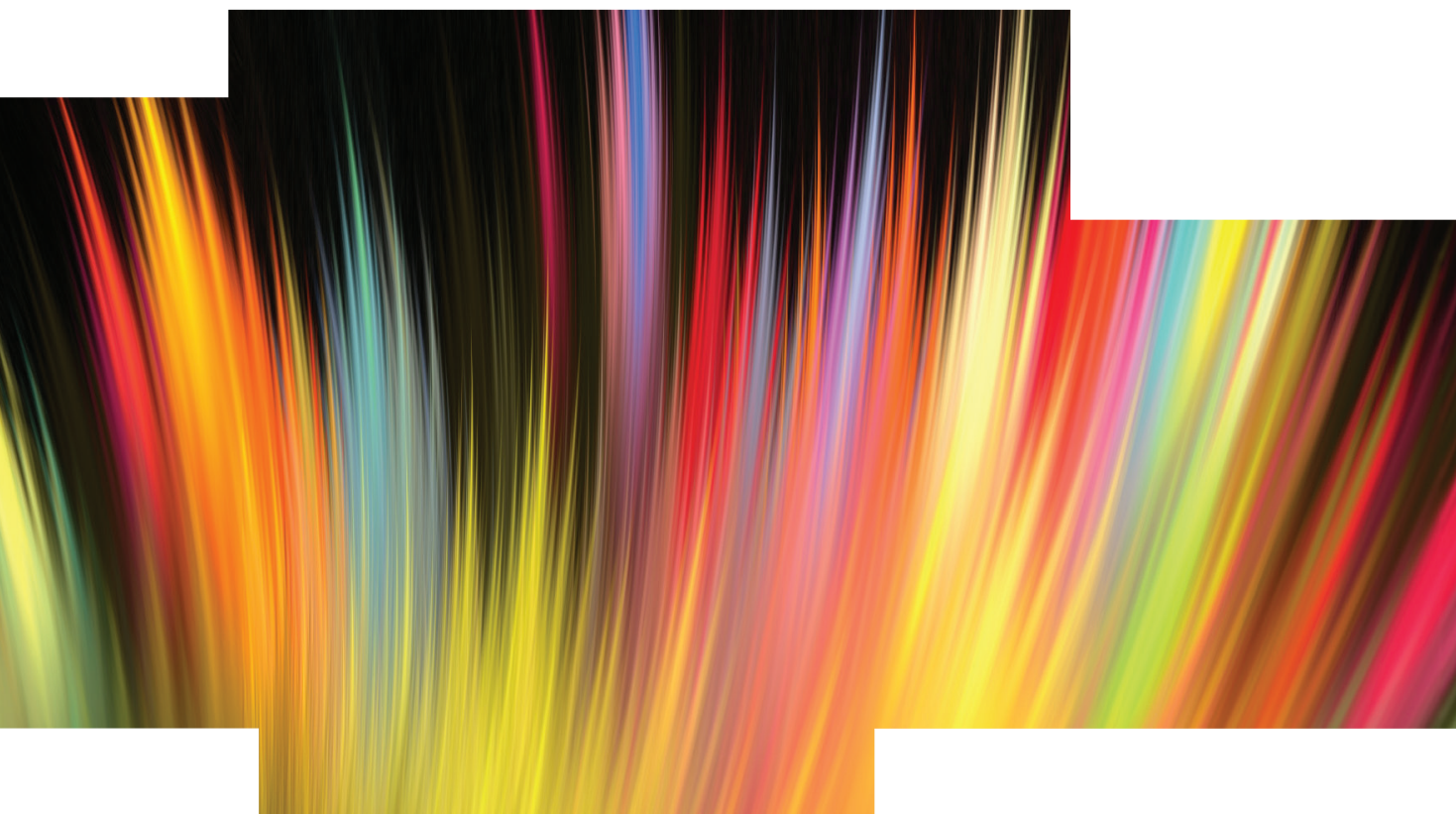
### Choosing the right approach

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 technology-solutions 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

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. ■

**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.

Copyright © 2016 McKinsey & Company.  
All rights reserved.

---

<sup>1</sup> "Tenth annual State of Agile survey," VersionOne, 2016, [versionone.com](http://versionone.com).

<sup>2</sup> 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.