



Photo credit/Getty Images

Transforming IT infrastructure organizations using agile

Santiago Comella-Dorda, Peter Dean, Vito Di Leo, Nick McNamara, and Pankaj Sachdeva

Traditional ways of managing IT infrastructure can impede the fast-paced delivery of digital solutions. Agile methods can be used to boost efficiency, speed, and quality.

Many companies have accelerated application development by adopting agile principles and modern software-engineering best practices, such as automated testing. Yet it remains uncommon to apply these methods and tools to IT infrastructure and operations, even though doing so presents opportunities to increase productivity and the pace at which digital products and services are brought to market. The typical IT infrastructure organization continues to emphasize stability over speed. Requests for infrastructure services still often go through an assembly line-style process involving many handoffs, long delays, and frequent misunderstandings.

Traditional IT infrastructure processes made sense in the past. But now that the latest technology advances have eliminated the need for manual configuration work and consumers expect to interact with companies digitally, it has become essential for companies to modernize their IT infrastructure organizations, thereby accelerating IT deployments and shortening time to market for technology projects. Four shifts can enable IT infrastructure organizations to operate in a more agile and efficient manner. The first of these shifts involves managing infrastructure much as application developers manage code, by using software to configure environments in a swift, reliable way. The other

three are organizational: forming cross-functional teams (or “squads”) of well-rounded infrastructure engineers that work using agile methods, simplifying processes for delivering infrastructure service offerings, and improving how infrastructure teams and development teams work together.

Using an agile transformation to modernize an IT infrastructure organization isn’t easy, but it is worthwhile. In our experience, agile approaches can enable IT infrastructure groups to boost their productivity by 25 to 30 percent in six to 18 months, depending on the size of the organization. The gains can increase further as automated solutions are built and fully adopted. Additional benefits often include improved infrastructure service delivery and shortened time to market for digital products and features. In this article, we explore how infrastructure organizations can modernize themselves using agile methods, starting with a glimpse of what the shift looked like at three companies. We also provide a look at the four shifts described above, along with practical recommendations for how to get the transition under way.

Three transitions to agile IT infrastructure

Three companies demonstrate how unique approaches to agile transformation, based on similar principles and tailored to their needs, can help modernize their IT infrastructure organizations while improving performance significantly. At a large provider of software and services, the infrastructure staff of several thousand people managed a global footprint capable of handling millions of active users and thousands of log-ins a second. The processes that the company had used to provide infrastructure services had grown more complex and labor intensive as the company grew, so it could take months to bring new products and features to market.

When the company’s IT infrastructure leaders modeled the effects of applying agile methods to their organization, they saw an opportunity to improve productivity by 20 to 25 percent in 12 to 18 months. Given the scale of the infrastructure group, the leadership team chose to roll out agile ways of working over that span of time iteratively, launching about 150 agile teams to bring new methods and technologies to the entire company. The leadership had teams focus first on improving the infrastructure department’s internal operations by simplifying and automating processes and then on developing self-service tools and application programming interfaces (APIs) that could be used more broadly.

A European financial-services company with a far smaller IT infrastructure organization also recognized that traditional processes for building and managing infrastructure were slowing the release of digital products and services, as well as the adoption of more efficient, sophisticated application-development practices and tools. This company too set out to introduce agile methods and to implement highly automated infrastructure service offerings within its organization. However, its approach was to roll out a new agile operating model to its entire infrastructure organization at once instead of iteratively, as the software-and-services company did. The company also chose to focus from the start on building an operating model and tools that would empower developers to manage the operations of their applications directly.

Another business—a large US-based financial-services company—also adopted an agile approach in its 250-person IT infrastructure & operations group. Like the European financial-services company, it rolled out a new agile operating model to its entire infrastructure organization at once. However, the company chose to focus initially on improving its processes. In six months, it completed

a transformation that cut IT costs by more than 35 percent and doubled overall productivity. With the new operating model in place, the company now plans to focus on automating up to 80 percent of its operations work.

Principles for agile transformation

Despite the differences between their transformation approaches, these companies followed many of the same principles. In the sections below, we'll explore those principles in four areas: technology, organization and talent, processes, and collaboration with developers (exhibit).

Technology: Defining infrastructure with software

One reason traditional infrastructure organizations operate slowly is that their technology systems require teams to configure infrastructure manually for each new application. To bring agility to the infrastructure function, companies can not only eliminate manual work by building automated systems that allow infrastructure to be defined by software but also provide “guardrails” that enable application-development teams to manage more of their own operations safely. And while it's possible to build such systems with existing infrastructure, automation becomes easier as a company moves more of its infrastructure onto modern platforms, especially cloud platforms offering a wide array of enabling tools and technologies.

At the software-and-services company, even though the infrastructure team had standardized much of the hardware and virtualization architecture, it still spent a lot of time creating custom virtual-machine and operating-system configurations for product-development teams. Solution engineers reviewed the needs of each application with its developers and then set up the necessary environments, which often involved performing many steps manually.

As part of the company's agile transformation, agile infrastructure teams implemented automated solutions to streamline the provisioning and configuration of servers. One agile team built and maintained a centralized platform that automated the provisioning of servers and could be accessed through self-service tools. Other agile infrastructure teams, each aligned with specific software-as-a-service (SaaS) products, automated the configuration of those servers for the products they supported, using a configuration-management tool to define the servers' configurations entirely in code. This change reduced build times for environments from several months to about ten minutes. After these solutions were implemented, whenever a cluster of servers had to be updated or expanded, teams could make the necessary changes rapidly, with minimal manual effort and risk of error.





The European financial-services company chose to automate its IT infrastructure offerings using similar technologies. As part of a broad push to adopt DevOps principles, it also sought to empower application developers to manage their own operations as much as possible. IT infrastructure squads built automated, self-service infrastructure solutions for application developers and taught them how to use those solutions. Developers could then, for instance, produce code to tell the system how to configure or update servers given the unique requirements of their applications.

Organization and talent: Building cross-functional teams

At traditional companies, infrastructure organizations have long been structured around teams with narrowly defined responsibilities for specific technical functions (for example, managing relational databases or operating systems) or stages of the plan–build–run IT service life cycle. Neither this structure nor the specialization it promotes is

EXHIBIT

A modern agile IT infrastructure organization relies on well-rounded engineers to work closely with developers and deliver solutions efficiently, making extensive use of automation.

	Traditional organization	Agile organization
 <p>Technology</p>	<ul style="list-style-type: none"> • Highly customized infrastructure, provisioned on request • Significant manual effort required from infrastructure teams 	<ul style="list-style-type: none"> • Standardized infrastructure service offerings with largely automated delivery • Self-service tools let application developers configure and control infrastructure on their own, with appropriate guardrails
 <p>Organization and talent</p>	<ul style="list-style-type: none"> • Technology- or function-specific teams • Staff with highly specialized skill sets focused on operations and administration 	<ul style="list-style-type: none"> • Integrated, cross-functional teams (or squads) build well-defined infrastructure service offerings • Infrastructure engineers with sophisticated development skills
 <p>Processes</p>	<ul style="list-style-type: none"> • Rigidly sequenced processes, with many handoffs among groups of specialists • Repetitive tasks (such as deployment and incident resolution) performed manually 	<ul style="list-style-type: none"> • Squads responsible for end-to-end delivery of service offerings • Processes in which repetitive work is automated and streamlined
 <p>Collaboration with application development</p>	<ul style="list-style-type: none"> • Infrastructure requests submitted as “tickets” • Relationship and service managers deal primarily with application developers on behalf of the infrastructure function • Developers not accountable for application code after it is put into production 	<ul style="list-style-type: none"> • Application-development and operations responsibilities become more integrated • Self-service tools let developers handle more operations directly

conducive to efficiency or agility, because multiple teams must typically work on each service request. To become more agile, infrastructure organizations can organize their staffs into small cross-functional teams focused on providing well-defined services. They can also develop modern workforces of well-rounded engineers who can learn new skills rapidly and work across multiple functional domains to carry out the end-to-end delivery of infrastructure services, as we describe below.

CIOs and technology leaders should bear in mind that engineers in agile infrastructure organizations typically need more diverse skill sets than application developers do. For infrastructure, that makes agile transformations more challenging. The infrastructure organization at the European financial-services company found some of the well-rounded infrastructure engineers it needed by carefully screening existing employees. The most capable ones were offered roles on infrastructure squads charged with building the highly automated self-service solutions described above.

At the software-and-services company, the leaders of the infrastructure group chose to organize their staff into skill-focused “chapters” to help with capability building, professional development, and standard setting. Chapter leaders determined which new skill sets their areas needed and were asked to develop training or hiring plans to meet those needs. For working purposes, the company organized everyone from those chapters into two types of cross-functional agile squads led by product owners who defined and prioritized the backlog of activities that their squads would work on. Infrastructure squads focused on developing highly automated foundational infrastructure solutions (such as server provisioning) that other teams could use to set up, manage, and decommission infrastructure. Product squads were aligned closely with specific SaaS product-development teams and worked to engineer

and automate hosting and operations for their applications, leveraging services from infrastructure squads when available.

Processes: Simplifying and integrating activities to minimize delays

The traditional IT infrastructure organization’s functionally oriented structure imposes a particular working style—specialized resources complete tasks in a prescribed order, with many handoffs between groups. This working style causes innumerable delays: every time a request is passed to a new group, it goes to the bottom of that group’s task list, where it might languish for days. Frequently, tasks are sent back to previous groups for clarification, increasing wait times even further.

Companies can eliminate many of these delays by creating small cross-functional teams as described in the previous section. Such teams can minimize or even eliminate process handoffs by managing the end-to-end delivery of specific service offerings. They should be empowered not only to deliver service offerings but also to improve their delivery by streamlining processes and engineering fully automated solutions.

The processes of the software-and-services company’s infrastructure group had become increasingly complex as the company grew and added new customer-facing products. That led to the use of project coordinators to help push service requests through the organization. After the company grouped its infrastructure engineers into agile squads, however, the waiting periods that had previously followed handoffs among functional groups vanished. That change alone halved the amount of time required to provide many core service offerings. The company’s squads also redesigned common processes to simplify workflows or eliminate unnecessary steps, such as certain approvals. The number of steps in virtual-server

provisioning, for example, was cut by more than two-thirds, and the remaining steps were then largely automated through better engineering.

By contrast, the US-based financial services company mentioned earlier took a different approach to compensate for the limited development skills of its infrastructure organization. First, it set up cross-functional squads to simplify processes without automation. The resulting productivity gains bought employees enough time to learn more advanced engineering skills. Then they began planning the development of automated capabilities to address common requests.

Collaboration with application development: Fostering understanding and accountability

Traditional infrastructure organizations have minimal interaction with application-development teams. Collaboration between the two camps is normally limited to the initial setting up of systems for new applications and the resolution of critical incidents. As a result, typical infrastructure engineers know too little about each of the applications they support to help improve the stability of those applications. Moreover, developers lack the awareness of operational issues they would need to engineer robust, easy-to-support applications. Modern agile organizations, by contrast, make a point of increasing the level of collaboration between their application-development and infrastructure functions.

The European financial-service company described earlier exemplifies one collaboration style: making developers accountable for operating their applications. Involving developers in the incident-response and postoutage follow-up processes for their applications makes them more aware of issues in their application code. Involving developers in operations also encourages them to write code easy to manage and support—they can be awakened in the middle of the night if incidents occur.

The large software-and-services company demonstrates a contrasting approach. Its infrastructure organization continued to support operations for application-development teams but found a new way of doing so: closely aligning agile product squads and application-development teams. The alignment greatly increased coordination and collaboration. Many of the product squads were co-located, at least in part, with the application-development teams they partnered with. Core members of each product squad would attend some of the agile ceremonies of the application-development teams. In addition, the close alignment helped infrastructure engineers to gain familiarity with the applications they managed, so they had a stronger attachment to the success of those applications, which could now be better monitored and supported.

An approach to transforming infrastructure using agile

In our experience, the challenges of modernizing IT infrastructure using agile can be overcome using a structured approach to designing, launching, monitoring, and enabling agile teams. (At larger organizations, applying that kind of approach in waves can help the transformation get under way quickly.) This can be effective as part of a broader effort to transform a company with agile methods, or as an effort that is solely focused on the IT infrastructure group. Either way, the key steps in structuring an agile transformation of an IT infrastructure function are as follows.

1. Create a vision for the new infrastructure organization, particularly how the organization should operate and how quickly it should evolve. Several key questions will help IT and business leaders to define their vision for the organization.

What infrastructure service offerings should the organization provide to application developers and business users? Establishing a catalog of infrastructure service offerings helps companies

to design and define the scope of agile teams and to decide which of them should own the tasks of delivering and improving those services.

How should the infrastructure organization collaborate with application developers and how should the interaction model evolve over time?

Teams that are closely aligned with application-development teams can be beneficial if the infrastructure organization has responsibilities related to operating applications (for example, deploying code).

How quickly should the organization push to engineer automated solutions and adopt cloud technologies?

The structures, processes, and skills of agile teams that focus on operations can be very different from those that focus on engineering infrastructure offerings.

How will infrastructure leaders and business executives gauge the efficacy of the transformation?

Going into an agile transformation of the infrastructure organization, business and IT leaders should set clear objectives for improving performance and value creation, so that they can track progress and results with well-defined measurements.

2. Segment and prioritize opportunities with respect to the potential to create value for the organization. It is important to assess demand for infrastructure by developing a data-driven understanding of past consumption patterns and projected future needs. Knowing how much work is involved in delivering specific infrastructure offerings helps with organizing the work into scopes appropriate for an agile team. If, for instance, demand for storage-related work calls for a workforce of 24 people—too many for a single team—the effort might be divided among two teams: one focused on block storage and another on file storage services.

Analyzing demand can also help with identifying the greatest opportunities for improving efficiency and with prioritizing the rollout of teams accordingly. For example, a company can realize a great deal of value in a transformation by assigning the first agile infrastructure teams to handle and improve frequently performed labor-intensive services.

3. Design each agile infrastructure team to match the focus of each team with the working methods it will use. Teams focused on developing automated infrastructure service offerings tend to be relatively small—typically with eight to 12 people. They usually find that they work best using the scrum methodology, developing solutions in two- to three-week development sprints. Teams focused mainly on operations (such as level-one support teams) might benefit from longer rosters of up to a couple of dozen people. These teams often use the kanban or scrumban methodologies, which are more appropriate for managing a continuous flow of unplanned or event-driven work.

Over the long term, it is often preferable to have the same infrastructure team own both the planned development work and the unplanned operational work for a specific offering. This approach encourages teams to identify operational issues and fix them. However, at the beginning of an agile transformation, separating out unplanned operational work can help newly established infrastructure teams to focus on engineering highly automated solutions.

4. Create a structured process for rolling out agile infrastructure teams. The process should give all the people involved enough time to prepare for the launch of their teams. Our experience shows that it is critical to provide time and guidance to train team members, develop a strong team charter, align key stakeholders, and build out an initial backlog.

At the software-and-services company, for example, before each agile squad launched, its product owner and scrum master received two days of role training on how to perform their new roles. They then completed a six-week self-organized program, facilitated by agile coaches, in which they designed their teams' vision, scope, objectives, performance metrics, minimum viable product for improving delivery, and composition. Product owners also had to identify their key stakeholders up front and to review their plans with them and with the sponsors of the transformation so that everyone was aligned. Once the product owner and scrum master had finished these steps, the agile coach would lead the full team through a one-week "sprint zero," when it received training on agile and built out an initial backlog of work. After the sprint zero, the agile coach attended key ceremonies during the first several sprints of the team to make sure it was stable.

5. Focus on the sustainability of the transformation. Soon after agile infrastructure teams have been launched, governance bodies (such as a committee composed of senior IT leaders) will probably be needed to ensure that the teams are advancing toward their goals, refreshing their objectives as the organization's priorities change, and improving their use of agile practices. In addition, many infrastructure organizations quickly discover a range of opportunities to build on the agile transformation's initial improvements. These

include revising career models to support new agile roles, adopting more flexible budgeting processes, and making strategic planning more agile.

Addressing these improvement opportunities will take time, but senior IT infrastructure leaders can handle the work by using the same methods their newly launched teams do. They can organize themselves as a team, create a backlog of opportunities, determine priorities, assign owners, and carry out the work in sprints.



Legacy IT infrastructure processes common at companies that weren't "born digital" can impede the rapid delivery of new digital products and features. Agile methods can speed up the process significantly, and the benefits often start to materialize within the first six months of an agile transformation. A modern IT infrastructure organization that collaborates closely with developers and uses automation to accelerate configuration and maintenance can greatly boost its own performance, along with that of the wider company. For incumbents facing the threat of disruption from digital challengers, this can help make the difference between success and obsolescence. ♦

Santiago Comella-Dorda is a partner in McKinsey's Boston office, **Peter Dean** is a senior expert in the San Francisco office, **Vito Di Leo** is an associate partner in the Zurich office, **Nick McNamara** is an associate partner in the Chicago office, and **Pankaj Sachdeva** is a partner in the Philadelphia office.

Copyright © 2018 McKinsey & Company. All rights reserved.