

Operations Practice

# The present-focused, future-ready R&D organization

There's no one right way to organize R&D. But a set of core design principles can provide the flexibility R&D organizations need to outpace competitors.

*by Anne Hidma, Sebastian Küchler, and Vendla Sandström*



**Across engineered industries**, the explosion in software has increased product complexity by an order of magnitude. Along with rapidly evolving technologies, fast-changing consumer preferences, accelerated product cycles, and the practical realities of globalized operations and markets, R&D departments are under unprecedented strain. As product variation grows and product portfolios expand, updating existing products compounds the already heavy load R&D organizations bear.

Yet amid these 21st-century challenges, R&D units are still following 20th-century models of organization—models not designed for today’s need for speed and the expanding web of interdependencies among all of the moving parts. The traditional component-based approach to R&D is no longer sensible in an era when digital and electronic systems are so thoroughly integrated with hardware. Still many companies struggle to shift toward an approach that focuses more on the function the customer wants, rather than the components that make the desired function work.

There is no one right way to organize R&D. But there are certain fundamentals that can help R&D organizations in advanced industries act more responsively and meet the burgeoning challenges they face today. From our work with clients and our extensive research, we’ve distilled a set of core design principles for R&D organizations and identified the important ones. By following these principles, companies can help their R&D organization serve as engines of innovation for outpacing competitors. And they can foster the agility organizations need in supporting collaboration among remote, distributed teams—as has become more important than ever in response to unpredictable external events.

### **A growing mismatch between design and function**

Determining the right structure for the R&D organization has never been easy. The division of responsibility is a balancing act between the project-management organization and the R&D line

organization, with inevitable trade-offs. Today’s R&D teams don’t have the luxury of following a sequential, piece-by-piece approach in which finished, designed components are handed off to testing at the end. Moreover, the teams need to be appropriately protected from the external and internal disruptions that the broader organization experiences, which today come with greater frequency.

As they’ve grown organically, many R&D organizations continue to operate with the same structures and processes they’ve used for years. Despite (or perhaps because of) the increasing inadequacy of those structures and processes, organizations don’t follow them consistently. Pet projects are often hard to kill, even long after their diminished promise becomes apparent. And because research effectiveness is hard to measure—and companies often don’t understand R&D costs or ways of working—the black-box image persists without challenge.

Thus, adhering to an existing structure isn’t enough: the shifting demands, the sheer volume of work and the growing complexity (much of it the result of software integration) make it incumbent on R&D organizations to reappraise their design. Instead, they can create new mechanisms to provide the coordination, transparency, governance, and risk protection R&D needs in the digital era.

### **A set of winning design principles**

In the ideal R&D organization, responsibilities are clearly established, and interfaces between and among teams (internal and external) are seamless and transparent. These requirements, although not new, have become even more important of late, particularly when more teams are working remotely. R&D organizations that fulfill them can better meet further requirements—managing complexity actively and efficiently while staying focused on the future, and also maintaining the tools and capabilities for adapting to change.

## Clearly delineate responsibilities for systems and end-to-end work

Historically, the R&D function has been organized according to field of expertise, components, or location which has the effect of creating silos. Product properties are defined at the start of the development process, without being analyzed according to larger internal systems or user functions. Little attention is given to thinking in terms of the overarching goals customers want to achieve, or to the growing interdependencies as software and digital functions have pervaded almost every engineered product.

Many of the complications R&D organizations encounter today are the result of organizational interfaces that don't match the product, along with a lack of transparency between groups. Take, for example, a feature such as lane assistance for vehicles. Developing further advances in this function depends on a high level of coordination among teams developing steering systems, brake systems, and electrical systems. But too often

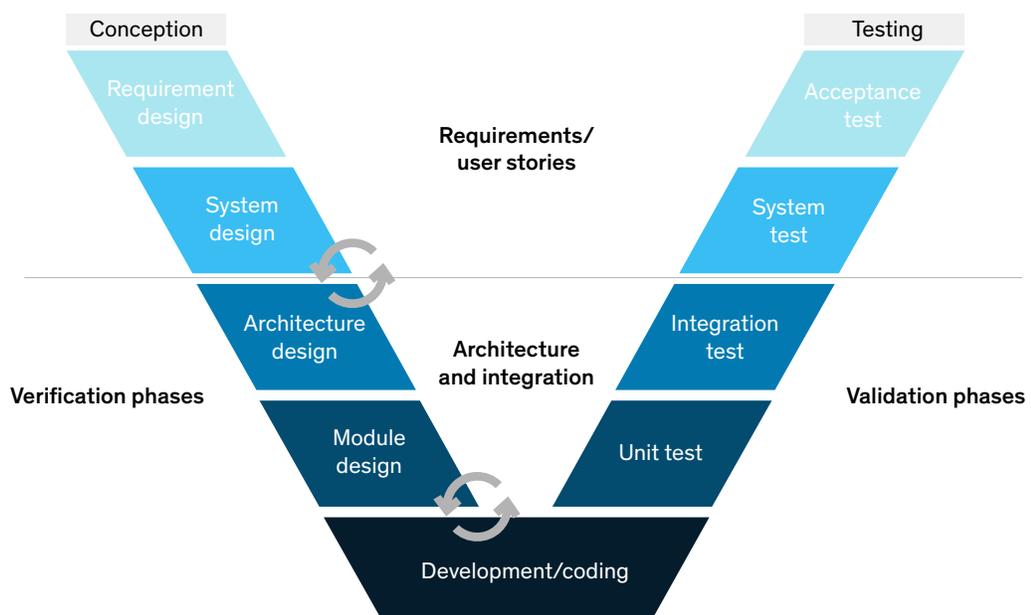
that coordination occurs only late in design, perhaps even the final testing phase, by which point addressing problems becomes expensive and time-consuming. R&D organizations are more effective when they shift their orientation from components to user function, while keeping platform development stable to ensure a core of commonly used modules.

With such a shift, assigning end-to-end responsibility for functionality has become imperative. Companies can assign responsibility for the complete product, as well as for the individual system layers, under the "V" model shown in Exhibit 1, which imposes oversight as ideas progress from concept through to market release, series development, and finally upgrades.

The process moves from left to right. Under "Conception," individual systems and their associated software are defined to fit customer demands and budget. Through early testing in the development process, issues and challenges

Exhibit 1:

### The V model divides development responsibilities by system layers, showing iterations and handoffs.



become apparent early on. The right side of the V comprises testing and integration, which are conducted along the system layers. By breaking a product's properties into requirements for systems and functions, the activities become transparent to everyone involved in development.

This approach enables iterative handshakes—more frequent interactions between concept owners and developers. Teams work together to translate properties into functional requirements. The approach also establishes dedicated responsibilities for complex functions and keeps the development process transparent. To coordinate and manage the interaction points, automotive companies tend to introduce central units that manage the whole integration process along the different development steps. These departments can be seen as a “stable backbone” within a dynamic development process, which helps improve planning for milestones and facilitates early failure detection. One machinery company, for example, set up quarterly integration meetings to align on priorities; the one to two days team members spend planning together gets them aligned for the next quarter's work.

Perhaps the biggest benefit to assigning end-to-end responsibility is that it enables R&D to manage the interfaces and different development cycles between hardware and software development. Functionality owners coordinate the development of complex and interdependent components and features, creating technical guidelines and specifications that support consistency. They effectively safeguard the implementation and validate that the solution fulfills its requirements over its entire lifecycle.

### **Keep functional interfaces across work sites to a minimum**

Companies can most effectively conceive of interfaces in terms of R&D's geographic footprint—a balance taking into account which activities are performed where, and how the locations must interact. Minimizing functional interfaces across multiple sites and avoiding

duplication of similar work are both important as well. Furthermore, to be cost-effective, location design can identify best-cost country sourcing for repetitive tasks, keeping in mind end-to-end responsibilities.

Dividing projects up among sites is usually less ideal, as people who work together in the same place tend to work more efficiently: earlier research found that with each additional development site for a given software product, productivity fell by about 14 percent (Exhibit 2.) Just the difference between one site and three sites amounts to a 37 percent decline in productivity.

Minimizing the number of handovers between sites—and making those that remain as smooth as possible—also helps. Distance between sites is not what matters; without the right management practices, a site across the city can seem as distant to employees as one clear across the country. But virtual teams can be as efficient as co-located teams, as long as the tools supporting the virtual work are utilized properly, communication is adapted accordingly, and everyone can participate on an equal basis.

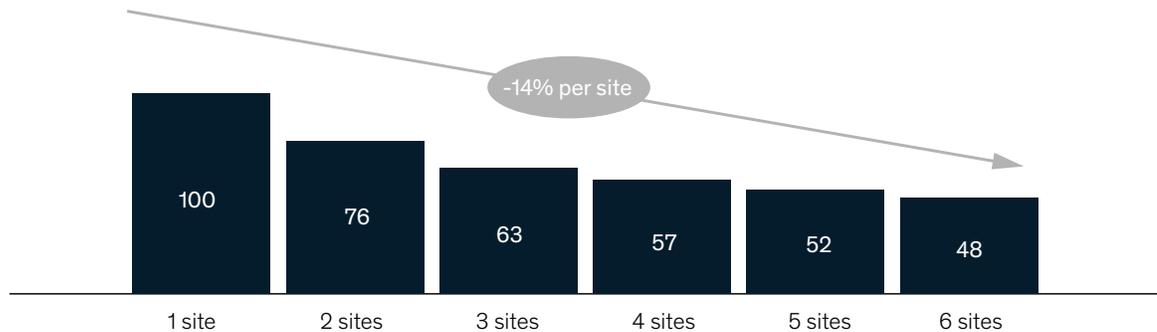
R&D leaders can consider future roles and competencies when thinking about the physical design of the department. Where should the development of next-generation products start? How could a transition to new products be built for sites currently focused on legacy products? How will cost and availability figure into the overall network structure? The answers form a long-term site strategy that can help avert a talent crunch.

The right footprint model also builds in a detailed understanding of local requirements, such as interactions with suppliers, local regulations, and internally, the interdependencies with other departments or components. The sophistication of the design work, and the degree of conceptual work that will be done in a particular location, will inform the kind of competencies and technologies that will be needed. For example, one white-goods manufacturer carried out almost all development in its home market, later building a few local

Exhibit 2:

## Fewer development sites means higher productivity.

**Productivity decreases by ~14% per extra development site added<sup>1</sup>**  
Complexity units per person-week, indexed



<sup>1</sup>Based on McKinsey Numetrics database

development centers in key markets to help adjust the products for local preferences, such as for refrigerator and freezer sizes, configurations, and color schemes.

### Synchronize software and hardware development

Complexity in all its forms has increased markedly—product variations alone have exploded over the past two to three decades, driven largely by the rise of embedded software and digital capabilities.

But R&D protocols often fail to account for the unique challenges of managing the development of integrated software and hardware. Software and hardware development follow different development cycles and require different approaches to project steering. And when digital features or components aren't explicitly considered in milestone planning,

integration problems and delays are almost inevitable.

As essential as synchronizing development may be, it isn't easy. In automotive, for example, map software generally takes about a year to develop, with frequent updates, while apps or innovative vehicle-control features (such as autopilot) may be updated monthly, with ongoing development and improvement. Contrast these cycle times with the hardware that runs navigation systems (which take two to three years to design and build), vehicle platforms (about seven years in the making) and basic vehicle components, such as heating systems and airbags—mature components that typically have a 10-year lifespan.

With such wide disparities in cycle times, transparency becomes crucial. The lack of it is a problem not only in concept development, but in

delaying product launches as well. For complex functions, such as lane assistance, R&D units may have limited ability to measure how mature the product really is. When changes are made, teams may therefore fail to assess the implications on other features currently in development. Beyond cost overruns, delays, and risks to product integrity, poorly managed complexity invariably leads to finger-pointing among system teams as well as conflict between R&D and the project-management team.

R&D organizations have two options for managing the complexities of syncing software and hardware development.

- Embedding software development within existing departments. This approach promotes integrated development—but in practice, processes are often designed from a hardware point of view, and software complexity is not managed effectively.
- Keeping development separate but coordinated. With this arrangement, individual technology components don't get short shrift. The onus is on leaders to establish synchronization points to identify potential conflicts that would require escalation to senior management.

The approach to take is generally determined by the nature of the product, as well as the organization's experience with software—bearing in mind that complexity will likely grow. Increasingly, services are developed not only within the engineering department but also within IT, creating still more interfaces and responsibilities, with implications for organization design.

### **Strike a balance between old and new technologies**

When it comes to developing new technologies, R&D managers have three choices: segregate them completely in a separate unit; include

them in the R&D organization, but keep them separate; or integrate them fully into the core R&D organization (Exhibit 3).

Segregating the current and new technologies has its advantages. Unfettered by standard processes, separated units are free to realize their full potential. The R&D organization keeps budgets separate and shields the new technology from the noise of existing projects.

But this option can be a hard sell to management, as creating a new unit can be costly, labor-intensive, and harder to absorb into the existing structure. Beyond the break-in time to adapt to existing products and processes, segregation also limits the broader organization's ability to transfer capabilities and knowledge, particularly given that cutting-edge technologies call for special (at times rare) expertise and training time for employees.

Short of total separation, there are essentially two ways to include new technology development within the R&D organization. Integrating new technologies fully into the existing organization helps transfer knowledge, and lets the new part of the organization tap into existing capabilities and processes, all of which helps in reaching scale faster. However, in this arrangement, there are risks—new technologies could be prematurely quashed by senior management, or if developed according to current methodologies, could yield less-than-optimal results.

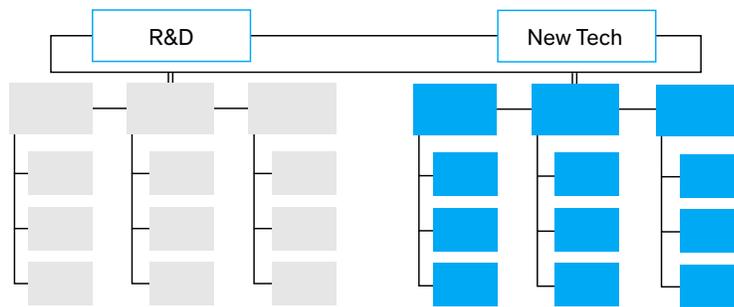
Most often, the best bet is a happy medium, in which new technologies are assigned to a separate team but explored within the current R&D organization (see sidebar, "An R&D makeover to sustain market leadership").

The right approach is also a function of the situation and the culture. Consider the electric powertrain in the automotive industry—the different manufacturers offer a sample of all of the archetypes.

Exhibit 3:

**R&D departments can choose among three organizational archetypes for new-technology integration.**

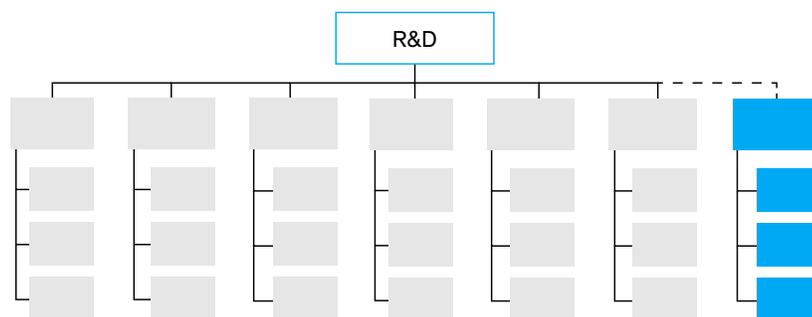
Classic R&D organization



Ring-fence new technologies in separate unit (with separate budgets)

- Separation allows for full focus on new-technology development
- Easier to segregate because the new technologies are nearly independent of the current ones

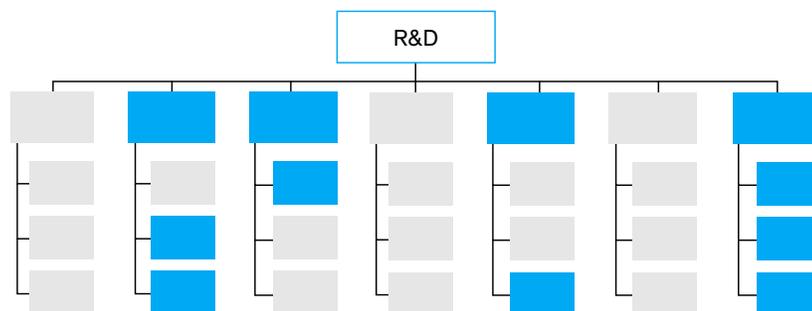
Segregate the new



Nucleus for new technologies is located within existing R&D organization

- New technologies are already linked to existing processes and systems because they're in the same organization
- Facilitates early knowledge transfer

Integrated new and old



Development of existing and new technologies is fully integrated

- Existing processes and tools can be fully leveraged
- The new technologies must be important or have high potential in order to be relevant in existing organization

## An R&D makeover to sustain market leadership

A global production-equipment manufacturer had long viewed its R&D organization as a crucial source of competitive advantage. But the company's rapid growth and increasingly complex product portfolio meant that more products were being developed in parallel. That led to even greater specialization among engineers and more technical interdependencies across modules. As the number of engineers and management layers grew, so did the number and complexity of interfaces, threatening the company's rapid growth.

Historically, R&D groups had been organized in two types of departments.

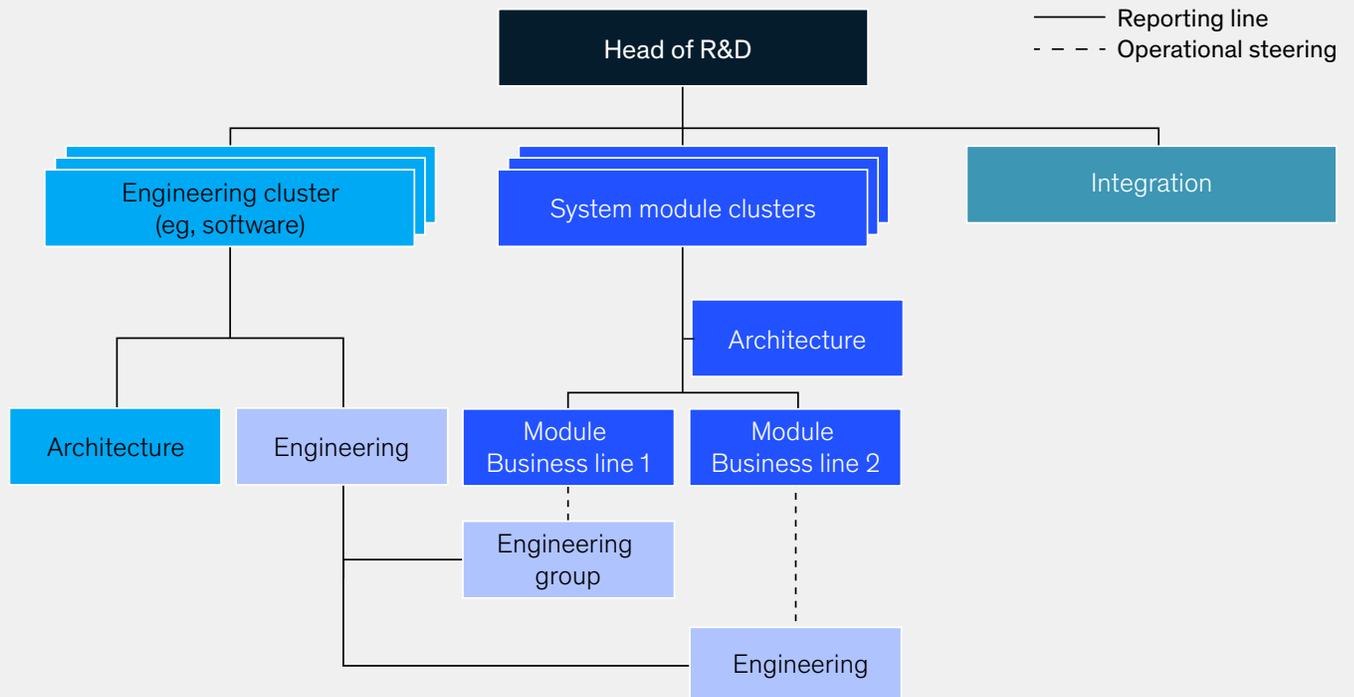
- System functions, which handled the functionalities that met system specs and customer requirements, such as for productivity and machine precision.
- Engineering functions, encompassing specialties such as electronics, mechanics, software, and environmental controls. These functions were required for developing the system modules and the system architecture needed for the integrity of the assembled machine.

### Cutting complexity

As a first step in redesigning the R&D organization, R&D leaders made system function leaders responsible for tangible and testable machine modules. System leaders' reports were given responsibility for the respective submodules. In that way, every production module and submodule would have a clear owner with end-to-end responsibility, from new-product introduction to third-line field-service support. Whereas before, each engineer worked on multiple products, under the new system each now works on only one business line and handles only one submodule at a time (Exhibit).

Exhibit:

**Redesigning R&D clarifies ownership, enhances transparency, and reduces interfaces.**



System-function departments are now primarily business-line dedicated. Each system function has a central architecture team that promotes commonality in the system modules' roadmaps and the maximum reuse of module elements among business lines.

Engineering-function teams (such as software teams) are largely dedicated to modules or submodules. Leaders have the authority to deliver their technical roadmap with more stable, focused, and experienced people. Within each engineering function is a central architecture department that's responsible for system

design and standards (the left side of the V in Exhibit 1 in the main text) and for setting guardrails for module design and development. This structure also ensures integrity in the final product.

### **Responsive and future-ready**

To maintain system integrity, shared platforms, and innovation- and knowledge-sharing across business lines, the company established several central teams. To manage competence (and continue building needed skills), the organization developed a taxonomy of critical competencies, assigned to VPs and managers

and governed through an annual planning cycle.

The stable, multidisciplinary teams that characterize the new design have created a solid foundation for piloting and scaling agile ways of working in the product development teams. Since the launch of the new organization, more than 2,000 engineers have migrated to agile methods. But engineers aren't the only ones working in new ways. By forging and executing the redesign as a team, R&D leaders have developed adaptive muscle, with the ability to adjust their organization to fast-changing requirements and environments.

### **To be future-ready, adopt new ways of working**

The traditional waterfall development model that some organizations still follow is so protracted that products can be obsolete by the time they are released. Long development times become impracticable when businesses factor in the out-of-sync cycle times of software and hardware components. In addition, a siloed and fragmented organizational structure makes it hard to respond nimbly to new process requirements.

Fast-changing customer demands and rapidly evolving technologies have increased the premium for enterprises and their R&D organizations to be adaptable, flexible, and future-oriented. And the coordination, integration, and speed needed in R&D today call for new ways of working. These include agile methods that enable fast iterations and cross-functional, flexible teams that ensure that the concerns of all relevant stakeholders—people from different functional units, as well as the different engineering teams, project managers, and customer representatives—are addressed. For example, a team working

on autonomous driving would include not only software engineers but also hardware engineers from the steering, brake-system, and overall car-design teams, as well as those working on user interface design.

To foster a future orientation within the R&D function, companies can adopt certain design features and practices, in particular those structures that promote agility:

- A flat organization in which teams are granted full responsibility to design solutions. This creates a strong sense of ownership among individuals
- End-to-end, cross-functional teams whose talent is drawn from all the relevant and traditional R&D functions. Often, teams are supported by individuals outside of R&D, such as marketing managers or customer representatives. Team membership is stable and changes only when projects are finished or strategic priorities change

- Pools of experts (both internal and external) that support projects with the talent they need
- Resource allocation that is flexible, shifting as needs change
- More co-location time for teams, wherever possible
- Role descriptions and rewards that align with the new organizational structure and targets

These practices usually suggest that the company might consider changing certain roles in the organization—particularly in light of the widespread need for more architects, as leaders are charged with empowering teams to foster innovation more than ever before. In fact, an automotive manufacturer saw its leadership transformation as a driving force for putting in place its new R&D organization.

A further question we are hearing is: how does all of this work in a remote working environment? The bulk of these practices can be implemented in a digitally enabled organization if co-location is not an option, with priority for practical matters such as ensuring teams have sufficient bandwidth to connect as often as needed. Clear roles and targets will be especially important as well, as will an emphasis on empowering teams and individuals.

---

With ever-expanding product portfolios—from more product variation to additional software embedded in engineered products—R&D organizations tell us they are struggling to keep up pace. That makes the shift from a traditional, component-based approach to a functional all the more essential.

Change isn't easy for this traditionally black-box area of the organization. Engineers themselves

struggle with how to reengineer their own work processes, often not knowing where to start. To determine the right blueprint, it helps to step back and reflect on current performance and future needs by asking a few central questions:

- Do we have a clear way of addressing the complexity that comes from interfaces?
- How are we handling interdependencies between systems? Is complexity increasing, and if so, are we well set up for the future demands?
- Do we have what it takes to adapt to a larger proportion of software development in our R&D?
- Are we sufficiently agile and flexible to adjust our focus based on changing demand? Could we handle more frequent changes in demand?
- How prepared are we for future technologies? Do we have the right structure in place to acquire and scale them?
- Do we have sufficiently clear roles, interfaces, and end-to-end responsibilities within R&D between teams and sites and to other departments?

There is no master formula for making this shift—nor could there be, given the differences across industries and from organization to organization—but certain principles prevail. Abiding by the principles outlined here can provide a blueprint needed for integration at the right points, and the much-needed transparency across R&D. If R&D is the company's engine of innovation, its own transformation is more than a matter of securing market share, it's about being built for a fast-changing present in order to secure the future.

**Anne Hidma** is an associate partner in McKinsey's Amsterdam office, where **Vendla Sandström** is a consultant, and **Sebastian Küchler** is a partner in the Munich office.