MITSloan
Management Review

# Why Learning Is Central to Sustained Innovation

SPRING 2016

Many managers think they can create better products just by improving the development process or adding new tools. But it's skilled people, not processes, that create great products.

Michael Ballé
James Morgan
Durward K. Sobek II

Michael Ballé, PhD, is an associate researcher at Télécom ParisTech's Project Lean Enterprise in Paris, France, and an author with the Lean Enterprise Institute in Cambridge, MA. James Morgan, PhD, is a senior adviser on lean product and process development to the Lean Enterprise Institute. Durward Sobek II, PhD, is a professor of mechanical and industrial engineering at Montana State University in Bozeman, MT.

Provided courtesy of the Lean Enterprise Institute, Inc., www.lean.org

Vol. 57 No. 3 | Reprint #57309 | http://mitsmr.com/1SPzyWr

MITSloan
Management Review

# Why Learning Is Central to Sustained Innovation

Many managers think they can create better products just by improving the development process or adding new tools. But it's skilled people, not processes, that create great products.

BY MICHAEL BALLÉ, JAMES MORGAN, AND DURWARD K. SOBEK II

**THE LEADING QUESTION**
How do lean organizations create better products?

**FINDINGS**

- **They invest heavily and continuously in the skills of product developers.**
- **Rather than developing single products, they think in terms of streams of products.**
- **They enhance learning through standards, model testing, and creative problem-solving.**

WE FREQUENTLY VISIT companies where managers say they want to improve their product development capability. They want to learn how lean principles and practices can improve their ability to innovate while reducing costs and improving quality. When we inquire about their approach to human resource development, we often hear, as one vice president of product development recently told us, that "of course, people are our most important asset. So we recruit and hire the top people from the best universities and get out of their way."

However, the only things many companies actually do under the heading of people development is to have an annual training-hours target and a travel budget for sending employees to conferences. If managers *really* thought that people were their greatest asset and that it's the energy and creativity of employees that drives innovation, why do companies do so little? Why doesn't growing and developing people excite them just as much as installing new additive manufacturing equipment or the latest cloud-based collaboration tool?

In studying manufacturing over the past two decades, we have learned that operational excellence is not achieved by just applying so-called "lean" practices to every process. More than anything, it requires cultivating an aptitude and an expectation for continuous improvement within every employee.[1] Similarly, we learned from studying lean product development that *people, not processes, make great products.* (See "About the Research," p. 65.) We frequently encounter managers who think improvements in the development process will pay off in better products. But better products don't just appear out of thin

KEITH NEGLEY/THEISPOT.COM

air: They are created by developers working with better knowledge and supported by good design processes.

The final design, including the product, manufacturing, and supply chain specifications, is the product of a complex network of interrelated technical decisions. How developers interact in the decision-making process — everything from framing problems, choosing ideas, and negotiating constraints to testing prototypes — is what shapes the product. In more transactional systems such as manufacturing or accounting, good processes usually produce a good outcome. What's important in lean product development isn't just whether you follow the right steps but how the work is done. Indeed, there are plenty of cases where companies followed "good" processes but had terrible results. The natural response is for managers to blame the process and then to add more best practices, increase the number and rigor of checkpoints, and change their flowcharts. Yet more often than not, the results continue to fall short. Until organizations view people as central (and leaders act accordingly), the risk that development process improvement efforts will not improve anything is frighteningly high.

The term "lean product development" is relatively new,[2] but the underlying concepts have been around for more than three decades. In the 1980s, an MIT study found that Japanese automotive companies followed practices that were profoundly different from those of other auto manufacturers, from the factory floor to new product development to supply chain management. The researchers began referring to the efficiency of the Japanese approaches as "lean."[3] Subsequent research focused specifically on Toyota Motor Corp.'s development practices as compared with those of its North American competitors.[4] As companies have applied the research to other areas, knowledge about lean development has deepened.[5] However, the critical role of people seems to have been overlooked.

As John Shook, CEO of the Lean Enterprise Institute, explained it, "The most important accomplishment of [Toyota] is simply that it has *learned to learn*."[6] Rather than being a state, lean is really a process by which companies can simultaneously improve product design, manufacturing capability, and supply chain efficiency. In new product development, lean is about advancing developer skills[7] through technical training and methods of collaboration so that each developer is able to design, develop, and deliver better products and services. Lean product development hinges upon having developers achieve *individual mastery* as the essential building block for better products — not delegating the overall responsibility to the human resources department. Companies can promote individual mastery by repeatedly asking three fundamental questions: (1) What do we need to learn about our customers, products, and production processes to design better products? (2) How do we learn this? (3) And finally, what kinds of organizational structures and routines will best support learning?

**QUESTION 1: What do we need to learn about our customers, products, and production processes to design better products?** With lean product development, the development process is geared toward introducing a constant *stream* of products at a steady rhythm (or, in lean terms, "takt"), as opposed to executing separate projects. From this perspective, a product such as Apple Inc.'s iPhone is a *value stream* made up of iPhone 1, iPhone 2, and so on, released at a steady pace. The idea is to be a little bit faster than the industry's natural rhythm of innovation.

Thinking in terms of a stream of products has significant impacts on the design process. Most new products aren't designed from scratch but evolve within the value chain. In some cases, the

“Until organizations view people as central (and leaders act accordingly), the risk that development process improvement efforts will not improve anything is frighteningly high.

initial product provides a way to test an idea, which can be refined based on the response.[8] Toyota's first Prius, for instance, wasn't designed to gain a large share of the auto market. Engineers were actually surprised by the car's commercial success after it became popular in Hollywood. The aim of the first Prius was to prove that hybrid engines made sense. With the second Prius, Toyota wanted to make the technology acceptable to mainstream users. Now there is a stream of Prius models with significant market share, with hybrid technology being used in other car segments, including sport utility vehicles and minivans.

In an existing value stream of products, some product features will change, but many will not. The development challenge within a value stream, then, is not about creating the killer product that will overtake the market but improving the value proposition of existing solutions so that current customers are motivated to upgrade and potential customers are compelled to buy the product for the first time. Increasing the value involves learning about three main aspects of the product design upfront: (1) fit-to-market — how to reduce annoying features and offer new ones that fit today's customers' preferences; (2) fit-to-manufacturing — how to design a product that is less costly to build with better quality; and (3) fit-to-industry — how to exploit the opportunities in the supply chain to get more from the network of suppliers and the technical advances they offer.

A good place to start is fixing problems with the existing product. Just as a company doing lean manufacturing will attempt to solve quality issues in the current product, development teams applying lean fix whatever quality issues exist now and make improvements for future products. Take, for example, a company that manufactures gasoline dispensers, which succeeded in doubling unit sales over just a few years. Previously, engineers accepted recurrent customer complaints about rust on metal panels as a fact of life when machines are exposed to moisture. But they decided to tackle each issue one by one, improving product quality. Then they focused on the product's central functions and redesigned important components to improve the machine's performance and the value it offered customers (who consisted of both gas station owners and motorists). The process led to a popular new product that has helped sustain growth in sales and market share.

## ABOUT THE RESEARCH

This article draws on our collective research and experience in lean manufacturing and lean product development over the past two decades. Two of us (Morgan and Sobek) conducted our doctoral research on Toyota's product development system. This work codified many of the foundational ideas underpinning current models of lean product development. Morgan subsequently worked at Ford as a global engineering director, where he deepened his understanding of lean product development through an in-depth collaboration with Mazda and helped transform Ford's product development capability. Sobek continued to develop the concepts through experiments and observation in academic and industrial settings. Our third author (Ballé) did part of his doctoral research on Toyota's collaboration with a key supplier and improvement of both the manufacturing process and product design. He has studied organization transformation problems for years through a large number of "lean transformation" initiatives. The ideas presented in this article are intended to assimilate our collective knowledge from research and practice into a useful framework.

Building in quality in engineering means conducting a value analysis of how products can be improved and then figuring out which features customers would like in a new product. With lean product development, the idea is to look at products as evolving value streams and each product release as an opportunity to learn about where the market is going.

**QUESTION 2: How do we learn what we need to know?** Just as important as what developers need to learn is *how* they learn. Improving each new product in the value stream depends on individual or team competence in being able to solve the immediate technical problems and to interface with what others are doing. Developers need to understand how their decisions impact manufacturing and the company's supply chain. This, of course, is a tall order, requiring both knowledge and rapid learning. The faster a development team is able to learn and the more knowledge the team has access to, the leaner (and more productive) the result.

The educational process in which people work and learn together by grappling with real issues is known as action learning.[9] Rather than acquiring knowledge through traditional methods, they learn on the job through a mentored process[10] as the work is carried out.[11] In a lean product development framework, action learning revolves around using standards, solving problems creatively, and testing models against the physical world.

> “Deciding what’s fixed and what’s flexible isn’t simply a feature of the development process — it’s part of the training process for skilled engineers.

**Standards** Lean differs from the typical development workflow in that it doesn’t try to specify or “freeze” technical solutions up front. In fact, the goal is to postpone key decisions as much as possible to avoid hitting unforeseen barriers later on.[12] What lean does try to specify are the things that should be fixed and the things that should be flexible. By making these determinations early, engineers know where they have flexibility and where they must operate within fixed constraints. Deciding what’s fixed and what’s flexible isn’t simply a feature of the development process — it’s part of the training process for skilled engineers.

Tackling the fixed and the flexible require different approaches. Standards apply to the fixed elements. They are most powerful when they are based on experience with previous products, because the impact of specific decisions is known, and not following them may lead to problems. In their most elaborate form, standards inform developers about the known performance limits and tradeoffs.[13] Lean developers use a variety of standards, including design standards that relate parameter values to performance, manufacturing standards that define current manufacturing capability, and development process standards that establish quality or test criteria.

Standards help developers make good decisions quickly because learning passes from one project to the next; there is no need to invest time and resources to learn the same things again. At the same time, new team members quickly acquire experiential knowledge by learning and applying it to their first projects, while experienced developers use standards to scaffold their own learning and transfer that learning to others.

**Creative Problem-Solving** Areas identified as flexible or to which current standards do not apply require creative problem-solving. However, rather than responding to the first thought that comes to mind and then improving on it through iteration, it’s important to explore many different options at once and to pursue them until it’s clear, through aggressive testing and cross-functional evaluation, that they aren’t feasible. The aim is to fully understand the design space underlying the problem in order to find a solution that works before committing to it.[14] Ideally, this “set-based” process will not only lead to a viable solution but also result in a preliminary set of standards that can be applied to the next project.

Whereas applying standards is a convergent thinking process intended to solve a problem rapidly by reusing existing knowledge, set-based concurrent engineering is a divergent thinking process that encourages developers to generate differing theories about a specific situation and test them until they are disproved or a clear winner emerges. Both thinking processes are essential for creativity, but they can be misapplied if fixed and flexible are confused. Developers need to know how to move seamlessly between the two modes, and leaders must recognize when to force the appropriate mode of thinking.

To illustrate, we learned at Toyota’s technical center in Japan that manufacturing process standards were fed into the development process at the earliest stages, and in most situations the product design was forced to conform to the standards.[15] In essence, the process was predesigned, and the product design followed. It went beyond simply using the same processes and sequences across several products; based on experience, Toyota process engineers identified the key elements required to set a world-class standard for manufacturing excellence in all of their manufacturing processes (for example, limits on machinability of specific materials and geometries, or locations of grab points for material handling, part location, etc.), and they developed standards based on that knowledge. As new products were designed, the same processes and equipment could be used as long as the product designs complied with the standards. From the early stages in the process, the process engineers knew that the product design either *could* or *could not* be produced. In special cases where the need for expanded manufacturing capability arose, Toyota assigned a high-powered team to coinnovate new products and new processes as one integrated system.

**Testing Models** The nature of development work requires developers to use models as representations of what they hope to build — anything from sketches and hypotheses to 3-D models and computer simulations. Most engineers recognize that engineering problems get solved through a mixture of going back to first principles and tinkering with concrete solutions. However, lean thinking has taught us that the quality of the models contributes enormously to both the quality and the speed of the solutions. Simulating solutions and testing them against existing standards, whether digitally, through analytical modeling, or even by trying things out with cardboard and tape, is central to learning for developers.

Models are incredibly important tools, because they are the medium of expression of new ideas and the means of assessing the suitability of ideas without creating the actual thing — making it feasible to look at many more ideas and to learn about them quickly and efficiently. However, models can't represent reality exactly. Indeed, problems can arise when decisions are made with an overreliance on models. As a result, the need to discover how well models predict the physical world leads to several practices that have come to characterize truly "lean" product development. For example:

- A design may seem great on paper, but a detailed review by one or more experienced designers can identify potential weaknesses. Design reviews provide an excellent opportunity to make the actual progress of development visible and to train younger developers. When conducted with cross-functional partners, such reviews can lead to tremendous learning at the intersections of disciplinary boundaries.
- Computer simulations may predict the behavior of a design concept, but only by observing a design in its intended context (what's known as the "gemba" in lean terminology) can we verify that the computer model has accounted for all of the significant variables.
- Building a prototype may be the best way to get a preliminary glimpse at how the item can be built. Developers relying solely on mental or virtual visualizations often overlook critical details that can make a huge difference in manufacturability.
- Physical testing should be done early and often to determine how well the concept meets requirements and to understand the design trade-offs.[16] Rather than testing merely to see if the design meets specifications or requirements, development teams should attempt to learn as much as possible from tests about how a design performs over a range of parameter values and its performance limits.

**QUESTION 3: What organizational structures and routines will support the learning?** Given that the ability to develop successful new products depends on learning, what structures do managers need to establish to extend the reach of learning? One important structure is the development process itself. From a lean development perspective, the best processes encourage learning and teamwork rather than demanding adherence to a rigorously detailed workflow. With that in mind, an effective lean development process consists of five overlapping phases:[17]

1. An early concept phase in which the project owner or core team defines the value proposition

> "Design reviews provide an excellent opportunity to make the actual progress of development visible and to train younger developers.

and where the fixed/flexible aspects of the product are fleshed out.

2. A preliminary design, or study, phase where the main flexible domains are explored for alternative solutions and functional departments seek agreement in how to realize the desired concept at the subsystem level.
3. A detailed design phase that is based on applying standards.
4. A preproduction phase for ironing out how to organize the manufacturing system and supply chain to produce the new product.
5. A tooling and prototype phase that involves interacting with suppliers.

The different phases reflect the inherently creative nature of product development. Because they depend on learning — that is, what's needed next depends on what's being learned in the current activity — highly detailed project planning is difficult. Phase 1, for example, includes both customer "immersion" to assess customer needs and technological immersion to understand the limits and capabilities of existing solutions. Based on this learning, developers can create product concepts or draft product brochures to share with expert groups. Then, based on discussions, the team will make decisions about which aspects of the concept will conform to the standard (which may require changes to the initial concept) and which areas are flexible (where products may need to go beyond existing standards, requiring innovation). Phase 1 concludes with a sharply defined product concept, timeline, and resource plan that all cross-functional partners agree to support. At Toyota, concept approval is a board-level event.

Phase 2, which is essentially a study phase, focuses on aspects of the product that are viewed as flexible. The idea is to consider several alternatives for each subsystem, to test and weed out the weak ideas, and to ensure compatibility with interfacing subsystems and with manufacturing capability before committing to a given solution. The process continues until the design converges on a solution that works from every relevant perspective. For example, a new car concept might be based on significant amounts of weight reduction while simultaneously having high levels of torsional stiffness (for better handling) and new standards for pedestrian protection. Manufacturing engineering and the designers of interfacing parts for capability with existing systems would review the various options. The alternatives that best meet the system of standards and constraints would be selected for further refinement.

Making the study phase lean requires pursuing the minimum information needed to kill an idea and holding off on detailed design and product simulation until the last possible moment. Developing detailed designs too quickly (only to abandon the idea later) is wasteful if the necessary information could just as easily be obtained from a quick sketch or mock-up. Lean development teams eliminate risk by having a backup solution ready to go if the new ideas do not pan out. The study phase goes beyond simply determining whether a concept meets requirements to understanding the design limits and the nature of trade-offs. Having a deep understanding enables the team to make good decisions and limits the number of iterations required in later phases. Phase 2 concludes with a realistic architectural plan for each subsystem and major component, along with preliminary standards for novel designs that will be adopted.

If Phases 1 and 2 are appropriately resourced,[18] the subsequent phases should run reasonably smoothly and not require the level of rework that often plagues conventional development processes. Since teams rely heavily on standards to ensure robust designs, they can eliminate much of the unknown, which allows for effective application of

> "Having a deep understanding enables the team to make good decisions and limits the number of iterations required in later phases.

> The chief engineer has no formal authority over design engineers but is the person who pulls the new product through the entire company.

detailed project planning tools and precise coordination of work between development subteams. Without the up-front work in Phases 1 and 2, there is too much uncertainty in Phases 3-5, which makes detailed project planning extremely difficult.

In addition to the development process, organizational structures should also support the learning processes of developers.[19] The chief engineer is responsible for the market success of the product and has the final sign-off on technical decisions. He or she is the product architect and makes the judgment calls on what should be included in the product and what should be left out. The chief engineer has no formal authority over design engineers but is the person who pulls the new product through the entire company, from design to manufacturing and supply chain.

The rest of the development organization is organized into teams within core areas of responsibility. The primary role of these functional groups is not to "manage" the developers but to act as on-the-job learning centers where senior practitioners pass on both the theory and the traditions of the job to less-experienced people in the form of knowledge standards and ways of practice. Product design is therefore a melding between the chief engineer's driving vision and the knowledge constraints of each development area. The various reviews that dot the development process are used as cross-functional learning events intended to make sure that engineering, marketing, manufacturing, and purchasing are on the same page about the fixed/flexible boundaries within a project.

Larger organizations can afford to have a group of chief engineers or a project management office and separate functional groups as departments or divisions. However, this may not be feasible for smaller organizations, where individuals need to wear several hats at once. However, the important thing is that someone acts in the chief engineer role — in other words, is responsible for vision, system architecture, and making sure that decisions are consistent with customer needs — and that individuals (or groups) maintain and apply knowledge standards in core areas.

## Implications for Managers

If you are a manager responsible for developing new products or services, you can take several steps to advance the development of your people.

First, you can make technical mastery an expectation of your organization and build it into the reward system and into the way you work every day. Toyota has made developing "towering technical competence" central in grooming new engineers and made mentoring fundamental to engineering leadership requirements.[20] Ford Motor Co. did this by creating a technical maturity model for each functional area within body and stamping engineering and supported it in the way the company made job assignments, ran design reviews, and rewarded its engineers. Individuals were assessed on their technical capabilities according to the model, and then plans were made to increase the individual's technical maturity and were incorporated into the individual's performance evaluation. This created a strong incentive to pursue technical mastery — and encouraged people to stay within their functional areas longer in order to build deeper expertise, for which they were rewarded.

Second, you should develop design standards and use them. You can start with the knowledge that already resides within the organization — spend a couple of hours in front of a smart board with your experienced developers and ask them to lay out how they would design a particular subsystem or how they would advise a novice designer. After codifying that information into a user-friendly format, use the design guide as the starting point of the next development project. Once you have design standards, you need to systematically update them based on the learning gained on each development project. Toyota and other companies set aside one to two weeks of the development project timeline to pause and reflect on what they have learned on the current project that should be incorporated into their design standards, and then they do the additional development work to codify that learning into a reusable format. You will need to make explicit who owns which design standards. Ideally, it will be the same group of people who will be using them, to ensure that they are easy to use and relevant.

Third, you should hold regular (for example, weekly) technical design reviews with the explicit aim of growing people through action learning and cross-functional collaboration. Some of the key questions to ask (repeatedly, even *ad nauseam*) are: (1) What is the design standard for the particular device or test? (2) How does the current design compare to standard? and (3) Where are the data to prove it? As much as possible, hold the design reviews on location (for instance, in the test lab, prototype shop, or factory) rather than in a conference room, and with actual artifacts, so that people can touch, handle, and point to what they agree with or not.

Fourth, you should take a critical look at your organization's formal development process and ask the following questions:

- Who is responsible for deeply understanding the customer, creating the system architecture, and coordinating efforts to ensure that all decisions align with customer interests?
- What problems do you intend to solve for the customer, and what additional value should the company offer?
- To what extent are people able to identify early the fixed aspects of the design (where we're not going to deviate from standard) versus the flexible aspects? Are you having candid enough conversations with the development team about what is straightforward versus what is tricky or even impossible?
- Are you investing enough resources in the study phase to investigate the flexible areas? Does that phase conclude with enough clarity and certainty about the remaining challenges of the project?
- To what extent do process checkpoints encourage learning as opposed to meeting requirements or task lists? Do you have the right number of checkpoints, do they occur at the right points in time, and are the right people involved?

Finally, you should take stock of the leadership culture within the organization. To support learning, ask the leadership team to focus less on decision making and assigning work and more on instructing and improving. Design and manufacturing standards (what we currently know about the product and the technical processes) are the main tools for deepening the organization's understanding of products and production processes. Encouraging problem-solving to resolve performance gaps with standards deepens the autonomy and insight of the responsible developers. Developer capacity is further enhanced by asking revealing questions about what people should be learning and how they are learning it.

Great people make great products. The explicit aim of lean product development is to grow better developers, who are increasingly knowledgeable and capable of solving problems and generating new solutions. People and people systems are the most important parts of a product development system, because people generate the knowledge necessary for innovation, and people apply that knowledge to designs for new products, new manufacturing systems, and more robust supply chains. Unfortunately, not every organization subscribes to this view — after all, it's easier to think in terms of process or tools solutions. However, both our research and our experience working with lean development organizations have helped us better understand what developers need to learn,

> “Are you having candid enough conversations with the development team about what is straightforward versus what is tricky or even impossible?

> People and people systems are the most important parts of a product development system, because people generate the knowledge necessary for innovation.

how they learn it, and what organizational structures best support them. By applying these insights and making people the backbone of the development system, companies can achieve a triple win: increased innovation, faster time to market, and lower costs.

***Michael Ballé** is an associate researcher at Télécom ParisTech's Project Lean Enterprise in Paris, France. **James Morgan** is a senior adviser to the Lean Enterprise Institute in Cambridge, Massachusetts. **Durward K. Sobek II** is a professor of mechanical and industrial engineering at Montana State University in Bozeman, Montana. Comment on this article at http://sloanreview.mit.edu/x/57309, or contact the authors at smrfeedback@mit.edu.*

## REFERENCES

**1.** M. Ballé, G. Beauvallet, A. Smalley, and D.K. Sobek, "The Thinking Production System," Reflections 7, no. 2 (2006): 1-12.

**2.** "Lean product development" is also referred to as "lean product and process development" to highlight the innovation involved both in new products and in the manufacturing systems needed to produce them. We use the shorter phrase in order to avoid confusion between manufacturing process and development process. Some similarities can also be found in the work on "lean startups," in which rapid learning techniques are applied to help entrepreneurs create new products or services with less waste and better chance for success.

**3.** J.P. Womack, D.T. Jones, and D. Roos, "The Machine That Changed the World" (New York: Rawson Associates, 1990).

**4.** See D.K. Sobek II, J.K. Liker, and A. Ward, "Another Look at Toyota's Integrated Product Development," Harvard Business Review 76, no. 4, (July-August 1998): 36-49; and J.M. Morgan and J.K. Liker, "The Toyota Product Development System: Integrating People, Process, and Technology" (New York: Productivity Press, 2006).

**5.** See, for example, A.C. Ward and D.K. Sobek II, "Lean Product and Process Development," 2nd ed. (Cambridge, Massachusetts: Lean Enterprise Institute, 2014); R. Mascitelli, "Mastering Lean Product Development: A Practical, Event-Driven Process for Maximizing Speed, Profits and Quality" (Northridge, California: Technology Perspectives, 2011); D.G. Reinertsen, "The Principles of Product Development Flow" (Redondo Beach, California: Celeritas Publishing, 2009); T. Schipper and M. Swets, "Innovative Lean Development" (New York: Productivity Press, 2010); and M.N. Kennedy, "Product Development for the Lean Enterprise" (Richmond, Virginia: Oaklea Press, 2003).

**6.** J. Shook, "Managing to Learn" (Cambridge, Massachusetts: Lean Enterprise Institute, 2008).

**7.** By "developer," we mean any person who plays a substantive role in the development of new products, including but not limited to marketers, product engineers, industrial designers, production engineers, prototypers, test engineers, and purchasers.

**8.** E. Ries, "The Lean Startup" (New York: Random House, 2011).

**9.** R.W. Revans, "Action Learning: New Techniques for Management" (London: Blond & Briggs, 1980).

**10.** S. Spear, "Learning to Lead at Toyota," Harvard Business Review 82, no 5 (May 2004): 78-86.

**11.** M. Ballé and P. Handlinger, "Learning Lean: Don't Implement Lean, Become Lean," Reflections 12, no. 1 (2012): 17-31.

**12.** A. Ward, J.K. Liker, J.J. Cristiano, and D.K. Sobek II, "The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster," Sloan Management Review 36, no. 3 (spring 1995): 43-61.

**13.** Ward and Sobek, "Lean Product and Process Development."

**14.** B.M. Kennedy, D.K. Sobek II, and M.N. Kennedy, "Reducing Rework by Applying Set-Based Practices Early in the Systems Engineering Process," Systems Engineering 17, no. 3, (autumn 2014): 278-296.

**15.** Morgan and Liker, "The Toyota Product Development System."

**16.** For a thorough discussion of the power of experimentation, see S. Thomke, "Experimentation Matters" (Boston, Massachusetts: Harvard Business Press, 2003).

**17.** F. Ballé and M. Ballé, "Lean Development," Business Strategy Review 16, no. 3 (autumn 2005): 17-22.

**18.** S. Thomke and T. Fujimoto, "The Effect of 'Front-Loading' Problem-Solving on Product Development Performance," Journal of Product Innovation Management 17, no. 2 (March 2000): 128-142.

**19.** Ward and Sobek, "Lean Product and Process Development."

**20.** Morgan and Liker, "The Toyota Product Development System."

**Reprint 57309.** *For ordering information, see page 6.*