

Toyota's Principles of Set-Based Concurrent Engineering

How Toyota's product design and development process helps find the best solutions and develop successful products.

Toyota Motor Corporation is an industry leader in product development lead time while using fewer engineers than its U.S. competitors. It has also shown remarkable consistency in market share growth and profit per vehicle, which led to cash reserves of \$21 billion, exceeding those of the “Big Three” automakers combined.(1) The Toyota Production System (TPS), dubbed “lean manufacturing,” has been critical in these accomplishments,(2) but we believe that Toyota's product design and development system is also an important contributor.(3)

While Taiichi Ohno and others have meticulously described the TPS, the Toyota development system has not been well documented.(4) Indeed, Toyota does not use many of the practices often considered critical to successful concurrent engineering and associated with Japanese manufacturers. Its development teams are not colocated. Personnel, with the exception of the chief engineer and his staff, are not dedicated to one vehicle program. Cross-functional job rotation is unusual for the first ten to twenty years of an engineer's career. Engineering and test functions rarely use quality function deployment (QFD) and Taguchi methods. Toyota excels at value engineering (VE) and value analysis (VA), yet Toyota engineers say they do not use any of the text-book tools and matrices for VE or VA. And there is nothing remarkable about Toyota's CAD or CAE systems. These practices, then, do not explain Toyota's effectiveness in developing new vehicles.

In a previous article, we called Toyota's product development system the “second Toyota paradox.”(5) TPS was the first; its features seem wasteful but result in a more efficient overall system, such as changing over manufacturing processes more frequently (presumably inefficient) in order to create short manufacturing lead times. The second paradox can be summarized in this way: Toyota considers a broader range of possible designs and delays certain decisions longer than other automotive companies do, yet has what may be the fastest and most efficient vehicle development cycles in the industry.

Traditional design practice, whether concurrent or not, tends to quickly converge on a solution, a point in the solution space, and then modify that solution until it meets the design objectives. This seems an effective approach unless one picks the wrong starting point; subsequent iterations to refine that solution can be very time consuming and lead to a suboptimal design. (6)

By contrast, what we call “set-based concurrent engineering” (SBCE) begins by broadly considering sets of possible solutions and gradually narrowing the set of possibilities to converge on a final solution. A wide net from the start, and gradual elimination of weaker solutions, makes finding the best or better solutions more likely. As a result, Toyota may take more time early on to define the solutions, but can then move more quickly toward convergence and, ultimately, production than its point-based counterparts.

In this article, we develop the SBCE idea by describing three principles that guide Toyota’s decision making in design. We present the conceptual framework of SBCE in more detail, tying it in with other characteristics of the Toyota development system, and discuss why the SBCE principles lead to highly effective product development systems.

Background

Our earlier article on the “second Toyota paradox” generated much interest in what Toyota does and how and also much skepticism: Does Toyota really do what we claimed? Many of the challenges focused on the more extreme examples we offered. We said, for example, that Toyota broadly explored body styles and could consider anywhere from five to twenty different styling alternatives. And we suggested that final styling decisions could wait as long as the second full-vehicle prototype, at the extreme. These extreme cases were intended to be just that —extremes to demonstrate a point, not averages.

More important than the specific numbers were the underlying principles of design that Toyota followed. We chose these examples to illustrate ideas, not to suggest that if a company makes lots of prototypes or waits until the very last minute to make decisions, its development process will improve. In fact, a good job exploring solutions on one project can lead to a very focused search and much more rapid convergence on a design in later projects.

Both the novelty of the idea and the skepticism we encountered led us to develop the paradigm of SBCE further. We began by collecting more data. The first author, Durward Sobek, learned Japanese and went to Japan for six months to gain a deeper understanding of Toyota’s development process. He interviewed managers and engineers from a broad range of design specialties including styling, body engineering, chassis engineering, power train engineering, vehicle evaluation, production engineering, and prototyping, and a number of closely affiliated Toyota suppliers. He quickly substantiated our previous claims and concluded that Toyota was in fact “set based” in more ways than we had originally thought.

In addition, we interviewed Japanese and U.S. managers and engineers at the Toyota Technical Center (TTC) in Ann Arbor, Michigan. TTC, Toyota's first attempt to develop vehicles outside Japan, has co-developed the Avalon and the 1997 Camry with Toyota's technical center in Japan. Bringing its development system to the United States has forced Toyota to make its design philosophy and principles explicit. The training materials and process for U.S. engineers provide great insights into Toyota product development.

What Is Set-Based Concurrent Engineering?

The puzzle to explain Toyota's product development practices began when we observed that Toyota does not use many of the practices often considered critical to successful concurrent engineering. How is Toyota able to do concurrent engineering so well?

Traditional, serial engineering is a series of functions, each designing to a single solution or point (see top of Figure 1). In this illustration, styling(7) generates its best single solution based on its criteria and "throws it over the wall" to marketing, which develops the best marketing plan based on what styling has handed it, and so on. Of course, this is a simplification; there are feedback loops, but the feedback from downstream functions comes later, often after upstream functions have committed to a particular solution. And, typically, the feedback consists of specific critiques that lead to minor changes to the base design. Smith and Eppinger have developed a quantitative model of "sequential iteration" in engineering design to help determine an optimal sequence of design tasks given that interdependencies exist between different tasks.

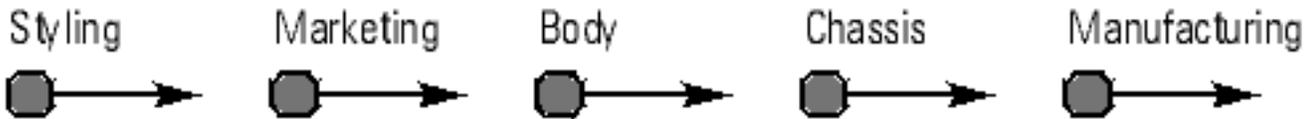
(8) This approach can help reduce the length of the feedback loops in the cases of the most critical interdependencies, but it is still an incremental improvement that stays within the paradigm of point-based engineering. Serial engineering is fraught with shortcomings due to the delayed feedback loops. In fact, the major rationale for concurrent engineering (CE) is to shift away from a serial "throw it over the wall" approach to parallel processing of activities. As usually practiced, CE attempts to bring more feedback upstream earlier, generally through face-to-face meetings.

Figure 1- Traditional Point-Based Approaches to Product Development

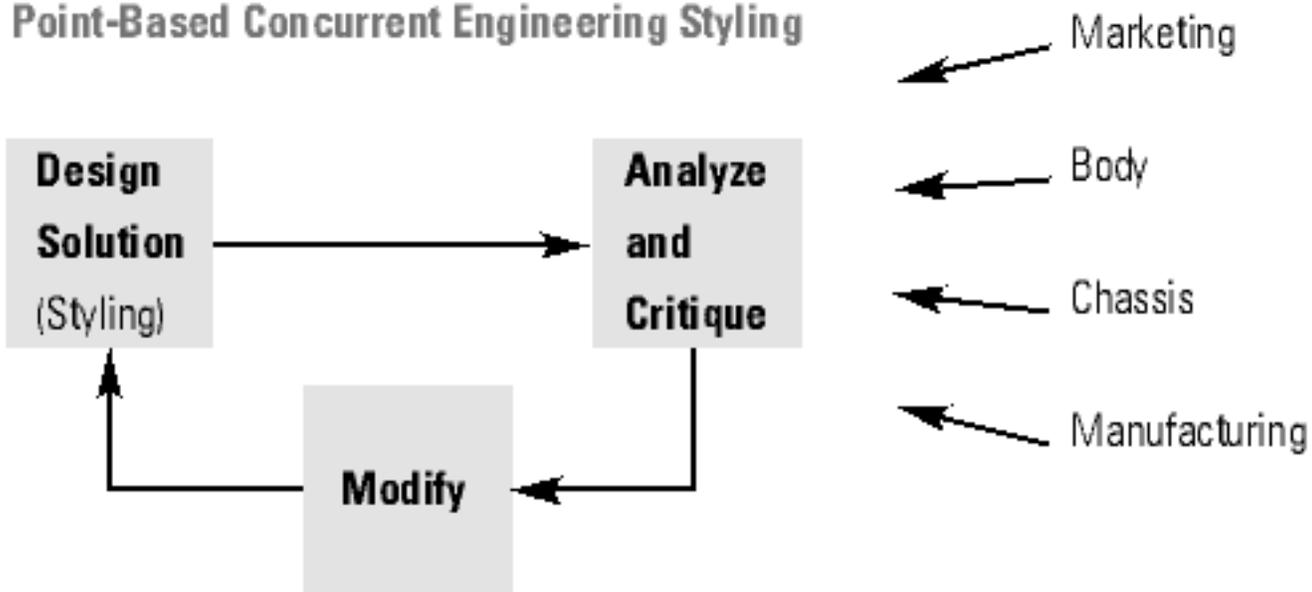
Figure 1

Traditional Point-Based Approaches to Product Development

Point-Based Serial Engineering



Point-Based Concurrent Engineering Styling



Typical CE in the United States is a refinement of point-based design, but still does not break out of the paradigm. The typical CE process looks something like the lower part of Figure 1. A function such as styling comes up with a design solution and very early in the process shows it to other functions for input. These downstream functions analyze and critique the design from their perspective. (For example, the top members of a Chrysler design team meet for an entire day every week.) Since this is done early, changes to the styling design are relatively easy and inexpensive, and ideally, the design team soon arrives at a solution that will satisfy all parties. While an improvement over serial engineering, the basic picture remains the same: the design team is iterating on one solution. We call it “point-based concurrent engineering.”

Problems with such an approach arise when engineers try to work concurrently with other development team members. As the design passes from group to group for critique from different functional perspectives (or even if they are critiquing it as a cross-functional team), every change causes further changes and analysis, resulting in rework and additional communication demands. There is no theoretical guarantee that the process will ever converge, and hundreds of engineers have told us that it often does not: the team simply stops designing when it runs out of time. Since the development organization never gets a clear picture of the possibilities, the resulting design can be far from optimal.

Despite these drawbacks, many companies have been successful with iterative, apparently point-based models. Cusumano describes Microsoft's approach to software development as a "synch and stabilize" approach.⁽⁹⁾ Microsoft development teams compile their code at very frequent intervals, usually weekly if not daily. Combining this very fast iteration with modular product architectures and extremely skilled programmers enables Microsoft to remain a leader in the software industry. Similarly, Terwiesch et al., in a case study of automotive climate control system development, suggest that an iterative strategy may be optimal when iteration or feedback cycles are fast, the cost of rework is low, and the quality of the initial starting point (i.e., the "first guess") is high.⁽¹⁰⁾ Also, fast iteration was a key ingredient behind the successful firms in Eisenhardt and Tabrizi's study of the computer industry.⁽¹¹⁾

Toyota's SBCE process, however, differs significantly from either of the models in Figure 1. Design participants reason about, develop, and communicate sets of solutions in parallel and relatively independently. As the design progresses, they gradually narrow their respective sets of solutions based on additional information from development, testing, the customer, and other participants' sets. As designs converge, participants commit to staying within the set(s), barring extreme circumstances, so that others can rely on their communication.

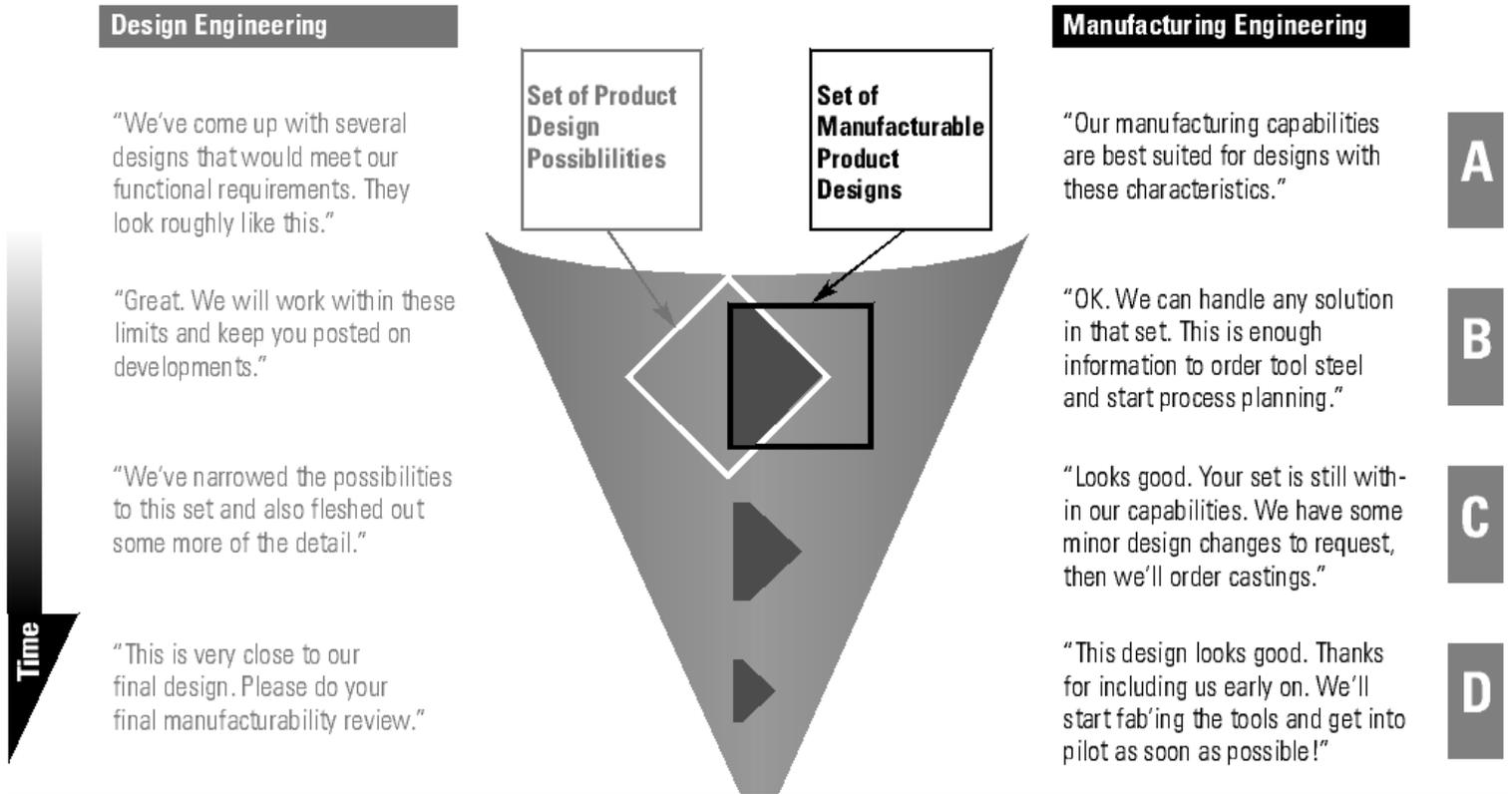
The example in Figure 2 illustrates the key characteristics of SBCE. In Part A, the two functions, design engineering and manufacturing engineering, define broad sets of feasible solutions from their respective areas of expertise (principle 1 — map the design space). In Part B, design engineering then smoothly refines the set over time by eliminating ideas not feasible from the manufacturing perspective (principle 2 — integrate by intersection). Design engineering continues to refine the set through further design and development work, while manufacturing engineering is also designing and refining at this stage. In Part C, the two groups continue to communicate about the sets under consideration, ensuring producible product designs while enabling manufacturing to get a jump start on design and fabrication of the production process (principle 3 — establish feasibility before commitment). The gradual convergence to a final design, Part D, helps the development team make sound design decisions at each stage. Gradual convergence also allows both functions to work in tandem with little risk of rework. Figure 2 is highly simplified, with only

two actors. SBCE works in the context of many actors defining sets, communicating sets, and converging to mutually acceptable solutions that optimize system performance, not individual subsystem performance.

Figure 2 - Example of Set-Based Concurrent Engineering

Figure 2

Example of Set-Based Concurrent Engineering



SBCE assumes that reasoning and communicating about sets of ideas leads to more robust, optimized systems and greater overall efficiency than working with one idea at a time, even though the individual steps may look inefficient. **(12)** In theory, SBCE could be conducted with no back-tracking or redoing at all. In practice, the costs of eliminating all back-tracking could probably not be justified. But a focus on convergence, rather than on tweaking a good idea to optimize it, can dramatically reduce the amount of back-tracking in the process.

Perhaps the best way to clarify SBCE is through examples of how Toyota practices it. The following examples demonstrate a range of approaches (explored in detail later) that are all consistent with the underlying philosophy:

In developing a vehicle's styling, Toyota makes more one-fifth scale clay models than most competitors do. Toyota maintains at least two full-scale models in parallel (typically from two studios), while most competitors pick one styling design, create one full-scale clay model, and go immediately to detailed design. Simultaneous with the development of the two to three

full-scale models, Toyota engineers develop structural plans for multiple styling design ideas and analyze them for manufacturability. This example illustrates Toyota stylists' and engineers' broader exploration of the solution space than in other auto companies, followed by a more gradual narrowing.

By the time a vehicle program reaches the die-making stage, U.S. car makers have long frozen the nominal dimensions and tolerances. Toyota (and other Japanese automakers), though, still views specifications as targets for die makers to refine. Die makers make the dies as close as they can to the CAD database, stamp out parts, and modify the dies so the body parts fit together (called "functional build"). Manufacturing engineers then set the tolerances based on their understanding of current manufacturing capabilities. Fit and appearance to the customer override concern for exactly matching specifications. The resulting dies, then, define the final specifications for the vehicle, not the CAD database. This example illustrates a belief that a nominal dimension, which appears to be a fixed, single point, really implies a range of acceptable solutions. Die makers have developed a tacit understanding of the range allowed in the design passed on from product engineering.

For every major part of the car, the engineers responsible for that part develop, maintain, and update an engineering checklist, which represents current capabilities — the set of feasible designs. Product engineers and production engineers also maintain checklists. When a product engineer begins a design, the production engineer sends the latest checklist so the product engineer knows the current constraints on the solutions space. As long as the product engineer's design meets those constraints, the design will probably be acceptable to manufacturing. This example illustrates how organizational memory can be facilitated by mapping the feasible solution space. Taking time up front to explore and document feasible solutions from design and manufacturing perspectives leads to tremendous gains in efficiency and product integration later in the process and for subsequent development cycles.

All three examples involve reasoning about sets of alternatives and a sophisticated understanding of the boundaries on the solution space. Later, we describe the underlying principles of SBCE in greater depth, along with additional detailed examples from Toyota automotive development.

Related Research

Students of design and creativity have traditionally emphasized looking at many ideas.**(13)** Many researchers also recognize that selecting the best idea from a large set is difficult, often requiring several narrowing iterations. For example, Pugh recommends a "controlled convergence" method for conceptual design: put all the concepts in an evaluation matrix, try to generate new concepts (expand the set), try to

improve concepts based on relative strengths and weaknesses, eliminate truly weak concepts, and iterate until a final concept emerges. **(14)** Pugh instructs designers to expand the concepts into greater and greater detail as they converge toward the best solution. This method, Pugh claims, can apply to any phase of the design process, not just to concept selection. Ulrich and Eppinger propose very similar methods, as do Wheelwright and Clark, with their “development funnel” and Dubinskas, with his “fermentation vat.” **(15)** Recent research indicates that “flexibility” of development systems is an important factor to success, particularly in unpredictable, rapidly changing environments. **(16)** Flexibility refers to the ability to make design changes in response to a changing environment with little or no penalty. **(17)** One way to maintain flexibility is to foster a number of ideas at the same time, converging to a final solution as close to market introduction as possible when making crucial design decisions. Iansiti describes a product development team at NEC that carried four distinct product concepts in parallel and worked for two years on design and development to arrive at a final concept. **(18)**

The principles of SBCE (and Toyota’s practices) more clearly focus the idea of working with sets. For example, many of the authors mentioned above seem to assume that a colocated team looks at the sets together, allowing informal communication. But we’ve observed that teams tend to focus quickly on one solution (Figure 1). At Toyota, communication about sets is explicit. These authors also imply that the sets are discrete lists of alternatives, ignoring other ways of representing sets. And the literature emphasizes the use of sets only in the “concept phase” of the process, although several authors recommend overlapping concept development with later phases of development, notably Iansiti and Bhattacharya et al. **(19)**

Clark and Fujimoto attribute the Japanese auto industry’s ability to do concurrent engineering to rich, bilateral, frequent communication. **(20)** At Toyota, communicating about sets of solutions, about regions of the design space, appears to increase the richness of communication while decreasing the length and frequency of meetings. Liker et al.’s data show that Toyota meets with its suppliers less often for shorter periods of time than do other major auto companies in the United States or Japan, even though Toyota suppliers appear to have greater design responsibility and fewer communication problems. **(21)**

Otto and Antonsson, Ward, Lozano-Perez, and Seering, and others have focused on formal representations of and inferences about sets of possibilities. **(22)** Smith and Eppinger have quantitatively modeled “parallel iteration” in order to aid project management in engineering design. **(23)** However, Toyota’s set-based approach does not depend on new computer tools.

A large body of literature looks at the evolution of fundamental science and engineering innovations at the macro-level over time. **(24)** When technologies are tracked over time, it becomes clear that a dominant design develops, and for some time, new products are incremental modifications of that design.

Technology cycles are defined by dominant designs and subsequent technological discontinuities. **(25)**

Nelson and Winter argue that designers of a technology have beliefs about what is technically feasible and

worth trying, and their search for alternatives is generally constrained by the dominant design.(26) Thus, limited exploration of solutions around a starting point, in this case, the dominant design, seems to be a natural tendency.

In this article, we focus on Toyota's detailed development of a mature product using proven technology, not fundamental technological discontinuities. Product development of mature technologies involves integration of detailed design decisions about thousands of parts and interrelated subsystems. Toyota excels at this integration by keeping options open longer, communicating about sets, and breaking free of some of the cognitive constraints described by Nelson and Winter.(27)

The Context of SBCE

Many factors contribute to the efficacy of the Toyota product development system; no one secret explains its success. SBCE is a critical aspect of the system, but it operates in concert with other, equally important principles on system design and the use of knowledge. Lengthy discussion of those factors is beyond our scope here,(28) but we emphasize that Toyota's set-based practices work in an engineering culture that is, in many ways, unique compared to most of the U.S. companies we've studied. For example, Toyota develops deep technical expertise in both its engineering and management ranks. Managers are excellent, experienced engineers who continue to view technical engineering as at least the second most critical aspect of their jobs (the most critical may be developing the engineers they supervise). Correspondingly, the principles require both suppliers' high engineering capability and a close but demanding relationship between the parent company and the suppliers.(29)

Toyota's chief engineer system is another critical factor. Toyota's three vehicle development centers have a matrix structure, with general managers heading functional organizations and chief engineers (CEs) leading vehicle programs. Clark and Fujimoto label chief engineers "heavy-weight program managers."(30) We prefer Toyota's term because the chief engineers are, in fact, the system architects (lead designers) for the vehicle, the most important technical decision makers on the team. Outside their small staff, they have no direct authority over functional engineers who report to functional general managers.

However, CEs are totally responsible for their vehicles, from the early concept stages through launch and into the initial marketing campaign. They perform vital systems integration, for while each function is responsible for its subsystem, the chief engineers are responsible for the total vehicles.(31) As such, their activities and their staffs focus on integrating across functions. The CEs make the set-based process work by controlling the narrowing process, insisting on broad exploration, resolving any disagreements across functions, and, when needed, making decisions on competing alternatives based on an analysis of trade-offs.

Principles of Set-Based Concurrent Engineering

In our earlier article, we described many examples of SBCE but had not yet systematized these practices into an overall framework.⁽³²⁾ We have now identified three broad principles, each with three different approaches to implementing the principle (see the sidebar). Together, the principles create a framework in which design participants can work on pieces of the design in parallel yet knit them together into a system. The remainder of the article discusses these principles in detail.

Principles of Set-Based Concurrent Engineering

1. MAP THE DESIGN SPACE.

- Define feasible regions.

- Explore trade-offs by designing multiple alternatives.

- Communicate sets of possibilities.

Principle 1 — Map the Design Space

“Map the design space” is how Toyota develops and characterizes sets of alternatives used in the convergence process. In product development, Toyota applies this principle on two levels. First, on individual projects, Toyota engineers and designers explore and communicate many alternatives. The exploration, analysis, and communication help the development team “map out” the possibilities, along with associated feasibilities and relative benefits or costs, for systems and subsystems pertaining to the vehicle and its production. The goal is a thorough understanding of the set of design possibilities that apply to the problem, or what design theorists call the “design space.” Second, the principle applies on an ongoing basis as Toyota engineers capture what they’ve learned from each project by documenting alternatives, trade-offs, and technical design standards. Next we explore three elements of mapping the design space.

Define Feasible Regions

Functions within Toyota’s development system simultaneously define feasible regions from their perspective. Each functional department (e.g., body engineering, chassis engineering, or production engineering), in parallel and relatively independently, determines the primary design constraints on its subsystem — what can or cannot be done or should or should not be done — based on past experience, analysis, experimentation and testing, and outside information (from the chief engineer and other groups such as production engineering).

Engineering checklists (or design standards) are one embodiment of this principle.**(33)** Every engineering function maintains checklists that detail design guidelines in any number of areas including: functionality (e.g., piston rings of standard material should have thickness of at least 1.8 mm to provide proper seal), manufacturability (e.g., bounds on acceptable curvature radii for sheet metal bending), government regulation (e.g., minimum strength characteristics for door members to meet side-impact crash tests), reliability, and so on. As an example, styling may have a checklist for the license plate well that contains dimensions, bolt-hole locations, regulations on tilt angles and illumination for various world markets, and restrictions on curvature radii and on the depth of draw, and so on. This documentation may also contain descriptions of what can and cannot be economically produced along with solutions to past problems, information on how to accommodate new production methods like new automation, suggestions to improve quality, reduce cost, enhance manufacturability, and so forth.

In the very early stages of a vehicle program, functions pass along their checklists to update each other on what is possible, what new technologies have become available since the last program, and what new problems they've been able to solve. When the chief engineer asks production engineering to participate in vehicle development, for example, the first step is to pull the checklists from the files and customize them for the particular vehicle program. Once completed, as one production engineering manager phrased it, "We hand over our experience" to body engineering. Body engineering then updates its own checklists based on these documents in preparation for the body design phase and, in turn, gives its updated checklists to styling.

Most U.S. manufacturing engineers we've interviewed typically wait to see a drawing before providing constraint information on manufacturability of a product design (i.e., iteration on a single solution). Written manufacturability guidelines are sketchy at best, with most communication between manufacturing and design engineering being oral.

At Toyota, engineering checklists are not simply long lists of design rules or guidelines imposed by a staff section. They explicitly define current capabilities as understood by the responsible designers. For example, each part on the body has a separate manufacturing checklist maintained by the engineering group that designs the stamping dies for that part. It shows, for example, the range of flange angles that produces a good part, what kinds of interfaces avoid assembly problems, how to design slip joints for a robust fit, what areas of the part tend to have formability issues, and quick calculations on the risks of curvatures and deformations, and so on. The engineers do not maintain detailed product histories but abstract their experience with each design to modify the checklist, further refining the possibilities.

Throughout the vehicle development process, design engineers use the checklists to guide the design and facilitate design reviews. If the design conforms to the checklist, the part will almost certainly meet a certain

level of functionality, manufacturability, quality, and so on. If it does not, discrepancies between the checklists and the design become the focal points of targeted discussions between functional groups. Our research in U.S. automotive companies has found that most do not maintain design standards, although older engineers have told us that it used to be standard practice before the 1980s. One reason may be that U.S. design standards tend to prescribe single solutions (e.g., “piston rings shall meet specification xxx” or “flange angles shall be y”), rather than describing a range of acceptable alternatives, resulting in a rigid, stifling design environment. Companies that do not keep design standards must rely heavily on verbal communication between functional groups and mental maps of the design space acquired through experience.

Explore Trade-offs by Designing Multiple Alternatives

Merely identifying alternatives is insufficient. To intelligently decide among alternatives, Toyota and supplier engineers explore trade-offs by designing and prototyping or simulating alternative systems and especially subsystems. **(34)** They make “best guesses” based on judgment and experience only when the decision is obvious, unimportant, or subjective; otherwise, they invest as required to gather quantifiable data.

Often a trade-off curve that establishes a relationship between two or more parameters is more useful than trade-off data on two or three alternatives. Whenever possible, engineers abstract from prototype data to establish mathematical relationships between design parameters and performance outcomes or use test data from a number of test pieces and interpolate relationships.

Many U.S. managers balk at the idea of “wasting” time and resources on ideas or projects that never reach fruition or are “thrown away.” And engineering processes we’ve observed in the United States tend to converge quickly on a “best guess” and then test it to see if it works. It usually doesn’t, so the iteration begins. But Toyota has a high regard for the learning acquired in working on multiple ideas. It seems to have faith that the skills and knowledge generated will pay off later, either directly through incorporation into the next project or indirectly through expanded skills sets and knowledge. Toyota seems to value highly the reassurance that the chosen solution truly is the best and deems it worthwhile to spend the resources for that assurance.

Perhaps the most striking example of exploring multiple (subsystem) alternatives is found in the planning phase **(35)** of engineering design that falls between concept design and detail design. In body engineering, the main output of the planning phase is the kozo-keikaku document (K4), roughly translated “design structures plan,” which is circulated for information and approval to all affected engineering groups.

Leading up to the K4 release, body engineers perform many subsystem studies of the body structure at a detailed planning level. These pre-K4 studies explore the engineering implications of vehicle styling. By the time body engineering is ready to create the K4 for a given vehicle program, engineers have created two,

three, or more designs of the key subsystems of the body structure to support alternatives for the overall structural plan. These designs have been evaluated by body structures experts as well as production engineering and other affected groups (e.g., chassis engineering for trunk compartment designs) and have incorporated feedback.

The body design process at Chrysler is markedly different (and remarkably point-based).⁽³⁶⁾ Even though the styling group may be considering a number of alternatives, the body engineering group studies only the “most likely” design in any depth. And even this study is little more than a simple beam-and-joint model of the body structure. Once the styling is set, body engineering begins its detailed design work. Subsystems of the body structure are detailed and incorporated into the overall structural model as they are completed. Analysis is performed on the model periodically, often resulting in changes and further modification. The Chrysler process then tends to iterate on a best idea, while Toyota’s process explores the engineering trade-offs explicitly through parallel development.

From this description, the Toyota process may appear to be slow and cumbersome, but in fact, body design at Toyota takes several months less on average than at Chrysler. One major reason, we believe, is that the initial design decisions Toyota engineers make are more sound than many competitors make, resulting in much less time “tweaking” the design.

The “explore trade-offs by designing multiple alternatives” principle is also seen in Toyota’s supplier relations (called “design-in”). One of Toyota’s exhaust system suppliers reported supplying ten to twenty exhaust systems for the first prototype of a Toyota car program. The supplier once made up to fifty for one car program, an extreme case, because the chief engineer wanted to know the trade-offs. In contrast, suppliers at Nissan and Chrysler produce only one exhaust system prototype per vehicle program. The Toyota supplier tests the prototyped systems on an engine on loan from Toyota, so test data accompany each exhaust design. The supplier combines the data from the set of systems to create trade-off curves for different variables, e.g., the gradient between back pressure and noise reduction for different values of a variable. Toyota uses the data along with tests on the vehicle to determine the optimum subsystem. There are similar processes for a variety of suppliers, including purely functional (brakes, power steer pumps, and so on) and appearance parts (exterior trim), although the numbers may be higher for exhaust systems since they are relatively simple and cheap to prototype.

Toyota also applies this principle when choosing suppliers for a vehicle program. In exterior plastic moldings, for example, Toyota expects interested suppliers to provide five to six design ideas based on some preliminary information. Submissions include detailed comparisons of material cost and performance, quality, tool and mold investment, total weight and cost, and so on. Toyota studies the proposal ideas from the competing companies and decides which supplier is most suitable for the job.

Then Toyota and the supplier begin discussing design details like the final styling, mounting locations, separations, and so on, reducing the set until arriving at final specifications.

In this case, Toyota gathers ideas and information on trade-offs from its supply base before making a basic decision on who will supply the part. Choosing a supplier that already produces a similar part may reduce time and investment in mold development, but new technology at another supplier may offset those costs. After supplier selection, design and price negotiations continue. Designs may not be finalized until eleven months before full production begins, and price may not be settled until even later.

The usual bid process in the United States differs substantially. Delphi, for example, receives drawings or detailed specifications from General Motors, and returns a bid (i.e., one design idea) to supply the part at a fixed price.⁽³⁷⁾ In this system, the customer receives only the best idea from each potential supplier, a limited view of the possibilities, and their tradeoffs. Toyota explicitly seeks to understand the possibilities in order to choose the best supplier (and ultimately design) for the vehicle and for the company's long-term interests.

Communicate Sets of Possibilities

Toyota engineers communicate about sets of ideas and regions of the design space, not about one idea at a time. If the feasible regions outline sets of possibilities, and trade-offs help one understand the implications of choosing an alternative over another, then communicating sets enables functions to understand the feasible regions of others. An excellent solution from one perspective may be a poor solution from another, making it suboptimal for the overall system. Additionally, single-idea proposals prompt responses that critique the design and suggest changes to accommodate another's considerations. Doing so initiates an iterative process of making decisions and changes. When a function proposes only its best idea, it does not give other functions a clear idea of the possibilities, so the iterative process is likely to involve much waste.

Just as knowledge creation involves making tacit knowledge explicit, U.S. companies wanting to implement set-based strategies must realize that the communication function needs to be explicit.⁽³⁸⁾ The most straightforward of sets are discrete alternatives, lists of ideas, drawings, and models. Subsystem options are often important, as are constraints on interfaces. Sets might also be bounded intervals for parameters (noise reduction will fall somewhere between fifteen and thirty decibels) or open-ended intervals (this part will need at least x cubic cm of space).⁽³⁹⁾ Other ways to communicate information about sets of alternatives may include trade-off curves, nomograms, performance charts, and best estimates with "design tolerances."

Toyota's engineering groups use standard design matrices to communicate subsystem alternatives or to provide feedback or suggestions for a design problem. On one axis are various design alternatives; on the

other, key criteria for evaluation. The grid contains the relative performances of alternatives along the criteria. The sample matrix shown uses a qualitative rating scale, but matrices can also include hard numbers (see Figure 3). Pugh and others advocate similar selection matrices.⁽⁴⁰⁾ But Toyota's engineers use them to communicate about alternatives at all levels of design, not just concept selection.

Figure 3 - Matrix for Communicating Alternatives

Figure 3

Matrix for Communicating Alternatives

		Evaluation Criteria				
Potential Solutions		Function 1	Function 2	Cost	Space	etc.
	X	○	○	▲	×	
	Y	▲	×	○	○	
	Z	○	▲	▲	○	

○ - Excellent ○ - Acceptable ▲ - Marginal × - Unacceptable

In summary, an important first step of SBCE at Toyota is the mapping of design spaces. This principle is accomplished by defining regions of feasibility (not just a single best idea), thoroughly understanding trade-offs through parallel design and development, and communicating about sets of alternatives (rather than just the best idea).

Principle 2 – Integrate by Intersection

As various functional groups begin to understand the considerations from their own perspective and others, design teams integrate subsystems by identifying solutions workable for all. Toyota uses three distinct approaches to system integration.

Look for Intersections of Feasible Sets

Having communicated the possibilities, teams can look for the intersections of different functions, i.e., where feasible regions overlap. If Toyota can identify an intersection, it finds a solution acceptable to all. Organizations that do not communicate sets and therefore cannot look for intersections often wind up trying to marry independently optimized components.

Toyota, on the other hand, looks for solutions that optimize total system performance.

Toyota's approach to nemawashi illustrates the principle. Nemawashi broadly refers to the Japanese practice of doing the "groundwork" (i.e., meeting and discussing with all affected parties) to get consensus before formally proposing a specific course of action.⁽⁴¹⁾ At Toyota, the emphasis of nemawashi is not simply on consensus or common ground, but on finding the best solutions for the overall system. The initiating party outlines alternatives and solicits input from affected functional areas to identify the best solution. The recipients study the proposals and critique them, identifying which solutions work best from their perspective and why, suggesting modifications and possibly proposing new solutions. The originating party then collects and integrates the feedback into a package that satisfies all parties, e.g., it locates a solution at the intersection of the feasible regions.⁽⁴²⁾

The styling development process is a case in point.⁽⁴³⁾ Typically, styling development at Toyota begins with the chief engineer's vehicle concept. From the concept, the styling team develops many ideas in a two-dimensional medium (sketches, renderings, elevations, and so on). From this large set, the team typically chooses five to ten ideas for each body type for the one-to-five scale clay model stage. All functions view the clay models, evaluate the different alternatives, and feed the information to the review committee. Simultaneously, body engineering begins developing structural layouts for the most promising alternatives. The review committee typically chooses two alternatives for the first phase of the full-scale scale model stage. The styling team then creates CAD data for the skin if it hasn't already done so and sends it to the design and manufacturing engineering functions. The engineering functions use the data to explore design alternatives for their own subsystem design. For body engineering, this may mean body structural design in CAD of detailed cross-sections for all structural panels, fastener and weld locations, preliminary crash analysis, and so on. Other functions do likewise (although body engineering is the function most affected by styling decisions). Engineering feedback to styling, then, is based on this preliminary but thorough design planning. Design engineers develop their subsystems in parallel with styling development, which often means engineers develop plans for more than one styling alternative. Based on engineering, manufacturing, marketing, and other feedback, the review committee decides on a final styling design, which may be one option or a combination of the two. Styling makes the master clay model for approval by the board of directors.

The styling development process illustrates several principles. Styling develops and communicates set of styling possibilities. Body engineering develops its own set of structural layouts possibilities in parallel. The design studies help body engineering understand the structural implications of the styling alternatives, which they communicate to the styling team and the chief engineer. This design-feedback-negotiation cycle is an explicit search for the intersection of acceptable designs from both aesthetic and structural perspectives. Additionally, in design reviews, production engineers ensure that designs conform to engineering checklists. The checklists define feasible regions. Design reviews explicitly seek out the incompatibilities of the set of styling proposals.

Impose Minimum Constraint

In the conventional U.S. approach, key decisions are made early on in order to simplify interactions among subsystems. These decisions maximally constrain design to achieve the desired effect. For example, a U.S. chief engineer told us that an early freeze on hard points (the key vehicle dimensions, such as window angle) is essential to avoid confusion among styling, body engineering, and manufacturing engineering. Next, styling approval freezes all the external shapes, then body engineering follows with part drawings and tolerances that ensure proper fit.

By contrast, Toyota often imposes the minimum constraint needed at the time, ensuring flexibility for further exploration or adjustments that improve integration. “Make each decision in its time” reflects Toyota’s thinking more than the U.S. practice, which seems to be “make decisions as early as possible to avoid confusion.” Bhattacharya et al. and Kalyanaram and Krishnan also claim that early freezing of specifications may be disadvantageous in many situations.⁽⁴⁴⁾ They describe trade-offs associated with the timing of specification freezes: early sharp definition allows more time to address integration needs and reduce product cost, but deliberately delaying specifications enables the development team to make last-minute adjustments (and thus reduce market risk). We have argued, however, that deliberate delays can help improve integration.⁽⁴⁵⁾ Teams have less chance of being “locked into” a suboptimal solution if they impose just enough constraint for one’s system to function properly and thus allow interfacing groups room to adjust and optimize.

One illustration of the minimum constraint principle can be seen in Toyota’s interpretation of “styling approval.” Even after styling approval, vehicle hard points retain a centimeter or so of flexibility so that problems can be resolved. Body engineering can make small changes to exterior sheet metal, provided the changes do not affect the customer’s perception of the vehicle’s style. By contrast, after styling approval in U.S. companies, the hard points cannot be moved.

Toyota’s functional build process is perhaps an even more striking example of minimum constraint.⁽⁴⁶⁾ Body engineers send their completed drawings (CAD data) to the manufacturing engineering group with

nominal dimensions only, no tolerances. The manufacturing group designs dies that will make parts as close to nominal as possible. It produces the dies and then, as part of the die tryout process, stamps out parts and rivets them together into a vehicle body. Manufacturing evaluates the screw body for imperfections and decides which dies need to change to get a perfect fit. It makes the most cost-effective adjustments to the dies, and the approved screw body parts become the master parts. The CAD data are then updated to reflect the parts coming off the dies. Manufacturing sets the part tolerances based on process capability.

In typical U.S. practice, manufacturing engineers are required to “make to print” or design and produce dies that make parts that meet design specifications (nominal dimensions plus or minus tolerances that the body engineer sets). During die tryout, parts must meet specification before proceeding to assembly, which means that expensive die changes may be made before the stamped parts are assembled. Once the dies pass this phase of tryout, they move to assembly where further changes to get the parts to fit together are likely. The result is a substantially more costly and lengthy process than Toyota’s. Allowing manufacturing engineers to set tolerances on product dimensions is unheard of in the U.S. auto industry.

The Toyota functional build process illustrates minimum constraint because body engineers do not specify parts more than necessary. They recognize that the overall fit and look of the vehicle body matters most to the customer, not any individual part. And since manufacturing has the experts for fine-tuning the body, it stands to reason that it should be responsible for the fit-and-finish fine-tuning and final tolerancing. Thus the constraint placed on manufacturing by the body engineers is not “make all the parts to nominal plus or minus 0.5 mm,” but “make a great looking body with tight, uniform gaps and use the nominal dimensions as a guide.” That any single part is off the specified dimension by a few millimeters is secondary to the look and integrity of the system.

Toyota often applies the minimum constraint principle in communicating “black box” requirements to its suppliers.⁽⁴⁷⁾ Toyota gives suppliers information such as performance requirements, interface requirements, and cost and weight targets. The supplier then designs the “black box” without Toyota intervention as long as it meets Toyota’s requirements and expectations. Toyota applies the minimum constraint necessary to achieve the required performance levels, then leaves it up to the supplier to complete the details. Minimum constraint is important because in applying it, development team members implicitly recognize that more than one solution may work.

Seek Conceptual Robustness

Taguchi popularized the concept of robust design, that is, designs that are functional regardless of physical variations such as wear, manufacturing variations, and weather.⁽⁴⁸⁾ Recently, robustness in market variations has also increased.⁽⁴⁹⁾ Strategies such as short development cycles, manufacturing flexibility, and standardization help get a product idea to the market faster and thereby decrease design susceptibility

to changes in market demand or competition.(50) The conceptual robustness principle embodies these two thrusts and includes design uncertainty: create designs that work regardless of what the rest of the team decides to do. If one function can create a design that works well with all the possibilities in another function's set, it can proceed with further development without waiting for additional information from that function. Such "conceptually robust" strategies can collapse development time significantly while providing other benefits such as ease of module upgrades and serviceability.(51)

Our best example of robust development practices comes from Denso, Toyota's major radiator supplier. Several years ago, Denso's design team projected a ten-year improvement in heat rejection versus size, aiming for an improvement so good that customers would design around Denso's product in order to get the technology in their vehicles. It then mapped out the current and, where possible, projected requirements of all its customers. Next, the team decomposed the problem into two parts, treating the radiator core (the actual cooling mechanism) and the upper and lower tank and ancillary tubing separately. It designed a family of cores that would meet all the customers' requirements it had mapped, while simultaneously designing a single production line to produce the entire family of cores with change-over in minutes. Tanks and tubing were customized to each vehicle.

In this case, robustness in market variation and robustness in design variation are the same. Denso no longer needs to design a radiator core, or new production lines, for each new vehicle. Rather, a specified set of core variations meets all customer requirements. Tanks and tubing are readily designed to provide appropriate interfaces; the same production line handles all the cores.

Principle 3 — Establish Feasibility before Commitment

lansiti claims that the "flexible" approach to product development enables overall system optimization because design participants seek to understand all the possibilities and interactions before committing to a particular design.(52) In fact, Toyota's entire set-based development process might be viewed as a system to fulfill the third and last principle: ensure that designs are feasible before committing to them.

By exploring multiple designs in parallel, and gradually converging on a single one, Toyota not only avoids late problems, but can intelligently make decisions that optimize system-level performance. Next we describe three additional ways in which Toyota ensures feasibility before commitment.

Narrow Sets Gradually While Increasing Detail

The set-based process involves not only the generation and communication of sets but a decision process that gradually eliminates possibilities until the final solution remains, rather than just picking the best from a set. As the set grows smaller, the resolution of each idea or design within the set grows sharper as

designers use increasingly detailed models (e.g., from sketches to CAD drawings to simulation to nonfunctional prototypes to fully functional prototypes). In this way, the design team more fully understands the relevant considerations before committing to a design.

Functions narrow their respective sets in parallel, communicating throughout to ensure that each function converges to a solution that integrates with the overall system (e.g., the styling and early body development example above). Eliminating ideas in stages allows participants to consider the most important alternatives more fully and gives them time to influence each other's narrowing process. Iansiti found that such gradual elimination of alternatives was key to the flexible product development systems among the computer manufacturers he studied. **(53)**

However, to complete the design in a timely fashion requires a certain level of decisiveness; the team has to make decisions sometime or the design will never happen. Thus the engineering team must balance deeper understanding of the problem time and resources for gaining that understanding. Knowing when to decide becomes a central task of the project manager (the chief engineer at Toyota). Every project is different, so the chief engineer often delays a decision to collect more data or presses for a decision to keep the program on schedule.

An example of gradual narrowing occurs in Toyota's interactions with a major supplier. Air conditioning system development for a particular vehicle program starts when the supplier receives general design requirements based on the overall concept. During the next several months, the supplier typically generates five to seven different ideas based on the relatively general design information and discusses them with Toyota. Later, at styling approval (the point at which Toyota senior management approves the exterior and interior designs for the overall vehicle), Toyota has narrowed the field to three or four air conditioning designs. The supplier then prototypes and tests each remaining alternative and gives Toyota the designs, test results, and trade-off data. Toyota considers these data as it evaluates each air conditioning system alternative. Typically, Toyota narrows the designs to two and then decides on a final design. The supplier details this design and provides prototypes for in-vehicle testing. Throughout the process, Toyota and the supplier negotiate the subsystem specifications and design.

Usually, Toyota orders prototypes of only one air conditioning system for its first official vehicle prototype build, but often chief engineers ask the supplier to maintain a second, back-up design well into vehicle testing. Toyota may request design changes as it tests the air conditioning system in the vehicle, but as time goes on, the permissible changes become smaller and smaller. When Toyota is satisfied, the final specifications are settled (including cost), and the program enters pilot production and full production. The same air conditioner supplier also supplies units to a U.S. automaker. The U.S. engineers, when shown the five to seven preliminary ideas, almost immediately choose what appears to be the best design at the

time and initiate development. Inevitably, the customer issues engineering change orders to the initial design, some of which can be very late and costly.

Stay within Sets Once Committed

The value of communicating about sets is limited if a team member jumps to a solution outside the originally communicated set. Participants must stay within the narrowing funnel so other team members know that they can proceed with further design work without concern for changes that cause rework. This approach can be followed only if design teams maintain robust sets that contain at least one workable solution. One technique for ensuring robust sets is to always have a fall-back design. If a new solution does not work by a specified cut-off date, the team resorts to the back-up solution.

Denso uses this principle when it targets a product for strategic development — radical advances in performance and cost leading to a decisive market advantage. First, it breaks the problem into manageable pieces. For each subproblem, the engineering team develops multiple alternatives, one of which must be a conservative solution (e.g., an existing or similar design). The elements are modularized to work with any combination of the other elements.⁽⁵⁴⁾ At a cut-off date, the team determines whether the radical solution of an element is successful. If not, the team tables the radical solution and focuses on the conservative solution.

Control by Managing Uncertainty at Process Gates

U.S. companies view design processes as networks of tasks and control them by timing information hand-offs between the tasks, as in the familiar PERT chart.⁽⁵⁵⁾ But Toyota views its process as a continuous flow, with information exchanged as needed.

Toyota manages this process through a series of gates, each tied to an integrating event that brings all the pieces together, e.g., a vehicle prototype. Toyota controls the level of uncertainty at these gates, reducing it at each successive gate. Uncertainty includes both the size of the set still under consideration and the depth of knowledge acquired. Each area of the vehicle has different uncertainty requirements at different stages.

Two extreme examples might be transmissions and exhaust systems. Transmissions are very expensive, long lead-time items. For most vehicle programs, the transmission is decided on very early. Thus the “transmission gate” is at the vehicle concept stage when uncertainty is reduced to zero years before production. Exhaust systems, on the other hand, are relatively simple to design and make and may still be undetermined at the first vehicle prototype stage, when uncertainty is reduced to within 5 percent of the

final specification; the subsequent prototype stage reduces the uncertainty to zero just months before production.

Toyota appears to judge uncertainty based on experience and simple (often unwritten) rules like, “If there’s only one solution and we have not established what it will cost to produce, then the set is too small.” The Toyota standard process provides considerable guidance, but the chief engineer can customize the standard to his particular situation. Failures to reduce uncertainty enough at the proper time turn into emergencies, with all effort focused on resolving the problem.

This approach appears to provide much better control over the process than the standard U.S. approach. The control may contribute to the rigidity of Toyota’s development schedules. In the U.S. system, functions hand off partial solutions to each other, knowing that changes will result later. It is therefore difficult to evaluate the quality of the information being transferred, and managers frequently complain that the gates are not taken seriously. At Toyota, though, each gate obliges every function to report, in effect, “We know that a good solution lies within the set of possibilities defined here.” Thus managers can more accurately determine development status.

Eisenhardt and Tabrizi’s work seems to support the Toyota philosophy.⁽⁵⁶⁾ They found that more frequent milestones (or gates) with less time between them and increased testing at subsystem levels correlated highly with faster development times, faster adaptation to changing environments, and higher rates of innovation.

Conclusion

Many companies seem to be looking for a design process cookbook, a step-by-step method that, if properly executed, produces a high-quality product quickly and efficiently. But teams seeking to reengineer development processes are often frustrated because rearranging the steps does not offer much improvement.

The principles outlined above are not steps, prescriptions, or recipes. Rather, Toyota chief engineers apply the principles to each design project differently. Design engineers use the principles to develop and evaluate a design process. The key to success is the implementation of ideas as much as the principles themselves. We are still trying to understand all the determining factors, which we hope will lead to guidance on how to implement the system. Future work will address the implementation of these principles in U.S. companies (as well as deepening our understanding of Toyota), but a few issues seem clear. First, efforts to implement a few principles in isolation will often fail; the system is tightly integrated. Second, the ingrained responses of many U.S. managers and engineers, derived from both their education and their normal approaches, work against these principles. Expending resources on designs that are not ultimately used, for example, may require a substantial cultural shift. Third, we do not believe that the

cultural differences embedded in these design principles are consequences of fundamental differences between U.S. and Japanese culture. Indeed, we hypothesize that these principles were widely understood and practiced in the United States before World War II (for example, Thomas Edison's well-known process for inventing the light bulb). They may also be widely and effectively practiced in the U.S. computer industry and perhaps others.(57)

In fact, we increasingly hear of U.S. companies that successfully use set-based approaches. For example, when the aircraft engine division of General Electric wanted to reduce development lead times, it shifted to a set-based strategy in which functional groups together selected feasible solutions, carried them through in parallel until the preferred solutions were verified, and then narrowed the set. The process enabled the development team to avoid much of the rework that might normally occur late in the project and to achieve the aggressive lead-time targets.

The change in engineering to a distributed, concurrent environment should involve a corresponding change in design method to a set-based process. The SBCE principles, along with Toyota's principles for integrating systems and cultivating organizational knowledge, appear to form the basis for Toyota's exceptional vehicle development capability.(58) More important, any product development organization that can master these principles and their application may be able to radically improve design and development processes.

ABOUT THE AUTHORS

Durward K. Sobek II is assistant professor in mechanical and industrial engineering, Montana State University. Allen C. Ward is president of Ward Synthesis, Inc. Jeffrey K. Liker is an associate professor in industrial and operations engineering, University of Michigan.

REFERENCES (58)

1. A. Taylor III, "How Toyota Defies Gravity," *Fortune*, volume 136, 18 December 1997, pp. 100–108.
2. J.P. Womack, D.T. Jones, and D. Roos, *The Machine That Changed the World* (New York: HarperPerennial, 1990).
3. Taylor (1997).
4. T. Ohno, *Toyota Production System: Beyond Large-Scale Production* (Portland, Oregon: Productivity Press, 1988); and S. Shingo, *A Study of the Toyota Production System from an Industrial Engineering Viewpoint* (Cambridge, Massachusetts: Productivity Press, 1989).
5. A.Ward, J. Liker, J. Cristiano, and D. Sobek, "The Second Toyota Paradox," *Sloan Management Review*, volume 36, Spring 1995, pp. 43–61.

6. M. Iansiti, "Shooting the Rapids: Managing Product Development in Turbulent Environments," *California Management Review*, volume 38, Fall 1995, pp. 37–58; and G. Kalyanaram and V. Krishnan, "Deliberate Product Definition: Customizing the Product Definition Process," *Journal of Marketing Research*, volume 34, May 1997, pp. 276–285.
7. The first phase of automotive development is usually the design of the vehicle styling by industrial designers.
8. R.P. Smith and S.D. Eppinger (a), "A Predictive Model of Sequential Iteration in Engineering Design," *Management Science*, volume 43, August 1997, pp. 1104–1120
9. M.A. Cusumano, "How Microsoft Makes Large Teams Work like Small Teams," *Sloan Management Review*, volume 39, Fall 1997, pp. 9–20; and M.A. Cusumano and R.W. Selby, "How Microsoft Builds Software," *Communications of the ACM*, volume 40, June 1997, pp. 53–61.
10. C. Terwiesch, C.H. Loch, and A. DeMeyer, "A Framework for Exchanging Preliminary Information in Concurrent Development Processes" (San Diego, California: University of California, working paper, 1997).
11. K.M. Eisenhardt and B.N. Tabrizi, "Accelerating Adaptive Processes: Product Innovation in the Global Computer Industry," *Administrative Science Quarterly*, volume 40, March 1995, pp. 84–10.
12. J.K. Liker, D.K. Sobek II, A.C. Ward, and J.J. Cristiano, "Involving Suppliers in Product Development in the United States and Japan: Evidence for Set-Based Concurrent Engineering," *IEEE Transactions on Engineering Management*, volume 43, May 1996, pp. 165–178; and Ward et al. (1995).
13. H.R. Buhl, *Creative Engineering Design* (Ames, Iowa: Iowa State University Press, 1960); and D.H. Edel, ed., *Introduction to Creative Design* (Englewood Cliffs, New Jersey: Prentice-Hall, 1967).
14. S. Pugh, *Total Design* (Reading, Massachusetts: Addison-Wesley, 1991).
15. K. Ulrich and S. Eppinger, *Product Design and Development* (New York: McGraw-Hill, 1995); S.C. Wheelwright and K.B. Clark, *Revolutionizing Product Development* (New York: Free Press, 1992); and F.A. Dubinskas, "Modeling Cultures of Project Management," *Journal of Engineering and Technology Management*, volume 10, June 1993, pp. 129–160.
16. S. Bhattacharya, V. Krishnan, and V. Mahajan, "Managing New Product Definition in Highly Dynamic Environments" (Austin, Texas, University of Texas, working paper, 1997); Eisenhardt and Tabrizi (1995); Iansiti (1995); and Kalyanaram and Krishnan (1997).
17. S.H. Thomke, "The Role of Flexibility in the Development of New Products: An Empirical Study," *Research Policy*, volume 26, March 1997, pp. 105–119.
18. Iansiti (1995).
19. Ibid; and Bhattacharya et al. (1997).
20. K.B. Clark and T. Fujimoto, *Product Development Performance* (Boston: Harvard Business School Press, 1991). See also: R.L. Daft and R.H. Lengel, "Organizational Information Requirements, Media Richness and Structural Design," *Management Science*, volume 32, May 1986, pp. 554–571.
21. J.K. Liker, R.R. Kamath, S.N. Wasti, and M. Nagamachi, "Integrating Suppliers into Fast-Cycle Product Development," in J.K. Liker, J.E. Ettl, and J.C. Campbell, eds., *Engineered in Japan: Japanese Technology-Management Practices* (New York: Oxford University Press, 1995), pp. 152–191.
22. K.N. Otto and E.K. Antonsson, "Tradeoff strategies in Engineering Design," *Research in Engineering Design*, volume 3, number 2, 1991, pp. 87–103; A. Ward, T. Lozano-Perez, and W. Seering, "Extending the Constraint Propagation of Intervals," *Artificial Intelligence in Engineering Design, Analysis, and Manufacturing*, volume 4, January 1990, pp. 47–54; W.W. Finch, "Predicate Logic Representations for Design Constraints on Uncertainty Supporting the Set-Based Design Paradigm" (Ann Arbor, Michigan: University of Michigan, Ph.D. dissertation, 1997); A.C. Ward, "A Formal System for Quantitative Inferences of Sets of Artifacts" (Colorado Springs, California: Proceedings of the First International Workshop on Formal Methods in Engineering Design, Manufacturing, and Assembly, 1990); and K.L. Wood, K.N. Otto, and E.K. Antonsson, "Engineering Design Calculation with Fuzzy Parameters, Fuzzy Sets and Systems," *Fuzzy Sets and Systems*, volume 52, November 1992, pp. 1–20.
23. R.P. Smith and S.D. Eppinger (b), "Identifying Controlling Features of Engineering Design Iteration," *Management Science*, volume 43, March 1997, pp. 276–293.

24. M. Tushman and R. Nelson, "Technology, Organizations and Innovation: An Introduction," *Administrative Science Quarterly*, volume 35, March 1990, pp. 1–8; and P. Anderson and M. Tushman, "Technological Discontinuities and Dominant Designs: A Cyclical Model of Technological Change," *Administrative Science Quarterly*, volume 35, December 1990, pp. 1–8.
25. Anderson and Tushman (1990).
26. R. Nelson and S. Winter, *An Evolutionary Theory of Economic Change* (Cambridge, Massachusetts: Harvard University Press, 1982).
27. Ibid.
28. For an in-depth look at the Toyota system, see: D.K. Sobek II, "Principles That Shape Product Development Systems: A Toyota-Chrysler Comparison" (Ann Arbor, Michigan: University of Michigan, Ph.D. dissertation, 1997); and D.K. Sobek II, J.K. Liker, and A.C. Ward, "Another Look at How Toyota Integrates Product Development," *Harvard Business Review*, volume 76, July–August 1998, pp. 36–49.
29. R.R. Kamath and J.K. Liker, "A Second Look at Japanese Product Development," *Harvard Business Review*, volume 72, November–December 1994, pp. 154–173.
30. Clark and Fujimoto (1991).
31. Sobek et al. (1998).
32. Ward et al. (1995).
33. This Toyota practice is remarkably consistent with Funk's finding that a key learning tool in Japanese product development organizations is decentralized groups studying, standardizing, and documenting processes and designs, then making the documentation publicly available to development team members. Engineering checklists are also consistent with observations of numerous researchers on organizational learning in Japanese companies. See: J.L. Funk, "Japanese Product Development Strategies: A Summary and Propositions about Their Implementation," *IEEE Transactions on Engineering Management*, volume 40, August 1993, pp. 224–236; K. Imai, I. Nonaka, and H. Takeuchi, "Managing the New Product Development Process: How Japanese Companies Learn and Unlearn," in K.B. Clark, R.H. Hayes, and C. Lorenz, eds., *The Uneasy Alliance* (Boston: Harvard Business School Press, 1985), pp. 337–375; K. Nobeoka and M.A. Cusumano, "Multiproject Strategy, Design Transfer, and Project Performance: A Survey of Automobile Development Projects in the U.S. and Japan," *IEEE Transactions on Engineering Management*, volume 42, November 1995, pp. 397–409; and I. Nonaka and H. Takeuchi, *The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Innovation* (New York: Oxford University Press, 1995).
34. Our emphasis on subsystem exploration is crucial. Toyota engineers explore the implications of a proposed system design by designing alternatives of the subsystems that make up the larger system. Only in this way can engineers truly understand the ramifications of making certain system-level decisions.
35. Some authors call this the "system design" phase of development, which is virtually nonexistent in the development systems of many U.S. companies. See: Ulrich and Eppinger (1995).
36. This based on a study by Sobek. The Chrysler development process is still undergoing significant change in how CAD technology is being used, so the process today may be more set-based. See: Sobek (1997).
37. Delphi engineers, incidentally, report considerable difficulty working with Toyota's more set-based approach of broad guidelines and late assignment of prices.
38. Nonaka and Takeuchi (1995).
39. Finch (1997).
40. Pugh (1991); and Ulrich and Eppinger (1995).
41. S.J. Black and M. Mendenhall, "Resolving Conflict with the Japanese: Mission Impossible?," *Sloan Management Review*, volume 34, Spring 1993, pp. 49–58.
42. Of course, in reality, finding this intersection may require several iterations that gradually converge on the final solution.
43. The term "styling" in the automotive industry refers to the design of the exterior and interior "look" of the vehicle. Stylists are typically industrial designers, not engineers, by training.
44. Bhattacharya et al. (1997); and Kalyanaram and Krishnan (1997).
45. Ward et al. (1995).

46. P.C. Hammett, W.M. Hancock, and J.S. Baron, "Producing a World-Class Automotive Body," in Liker et al. (1995), pp. 237–262.
47. Clark and Fujimoto (1991); and Liker et al. (1995).
48. G. Taguchi, "The Development of Quality Engineering," *ASI Journal*, volume 1, Fall 1988, pp. 5–29.
49. Iansiti (1995); and Eisenhardt and Tabrizi (1995).
50. Bhattacharya et al. (1997).
51. T.S. Chang, A. Ward, J. Lee, and E.H. Jacox, "Conceptual Robustness in Simultaneous Engineering: An Extension of Taguchi's Parameter Design," *Research in Engineering Design*, volume 6, November 1994, pp. 211–222.
52. Iansiti (1995).
53. Ibid.
54. Modularity involves breaking the product into semiautonomous chunks with standard interfaces which allow different "modules" to be replaced. A good example is the personal computer. Most computers have a modular architecture that allows customers to swap out memory, monitors, hard drives, and so on without loss of functionality. See: K. Ulrich, "The Role of Product Architecture in the Manufacturing Firm," *Research Policy*, volume 24, May 1995, pp. 419–440.
55. For examples, see: Smith and Eppinger (1997a) (1997b).
56. Eisenhardt and Tabrizi (1995).
57. For evidence in support of our hypothesis, see: Iansiti (1995); and Eisenhardt and Tabrizi (1995).
58. See Sobek et al. (1998).

ACKNOWLEDGMENTS

An earlier version of this paper appears in the proceedings of the 1996 American Society of Mechanical Engineers Design Engineering Technical Conferences, Design Theory and Methodology division (ASME paper # 96-DETC/DTM-1510). We thank Toyota Motor Corporation for its generous support in allowing us access to engineers for interviews. We particularly thank Mike Masaki, president of the Toyota Technical Center, for his advice, support, and invaluable assistance. This research was funded by Air Force Office of Scientific Research contract DOD G-F49620-93-1-0612, administered by the Japan Technology Management Program at the University of Michigan.

TAGS: Business Models, Design, Product Development, Product Strategy, R&D, Toyota