# Design Methods in the Aerospace Industry: Looking for Evidence of Set-Based Practices

by
Joshua I. Bernstein

Master of Engineering Aeronautics and Astronautics
Massachusetts Institute of Technology, 1997

Bachelor of Science Mechanical Engineering
The Johns Hopkins University, 1996

Submitted to the Technology and Policy Program in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
May, 1998



Author ______________________________________________
Technology and Policy Program
May 8, 1998

Certified by ______________________________________________
John Deyst
Professor, Departments of Aeronautics and Astronautics

Accepted by ______________________________________________
Richard de Neufville
Chairman, Technology and Policy Program

# Design Methods in the Aerospace Industry: Looking for Evidence of Set-Based Practices

by

Joshua I. Bernstein

Submitted to the Technology and Policy Program on May 18, 1998 in partial fulfillment of the requirements for the degree of Master of Science in Technology and Policy.

## Abstract

A new paradigm in engineering design, known as set-based concurrent engineering (SBCE), has been proposed which seems to offer advantages over more traditional techniques. This research, therefore, had three goals: 1) to develop a clear understanding of the definition of SBCE and to contrast that definition with other theories, 2) to assess the "set-basedness" of the aerospace industry, and 3) based on the assessment, to propose a model for implementing SBCE within an aerospace development project. While set-based concurrent engineering consists of a wide variety of design techniques, the basic notions can be stated in two principles: 1) engineers should consider a large number of design alternatives, i.e., sets of designs, which are gradually narrowed to a final design, and 2) in a multidisciplinary environment, engineering specialists should independently review a design from their own perspectives, generate sets of possible solutions, and then look for regions of overlap between those sets to develop an integrated final solution. This research found that while no company's design process completely fulfilled both of these criteria, many set-based techniques are used within the aerospace industry. Building on some of the observed industry practices, a design process model is proposed which combines concepts from lean manufacturing, such as "flow" and "pull," to implement set-based concurrent engineering.

Thesis Supervisor: John Deyst
Title: Professor of Aeronautics and Astronautics, MIT

# Acknowledgements

Many people have assisted me in the endeavor now represented by this thesis. Without their time, effort, advice, and support I could never have completed this work.

I first owe my gratitude to the Lean Aerospace Initiative, which funded this research. LAI represents a unique opportunity for a graduate student to work directly with members of the aerospace industry, and I consider myself very fortunate to have been able to make a contribution to this effort.

LAI is not simply an abstract organization, however. I owe a great debt to the many people who are LAI. First, I must thank my site visit coordinators and escorts at the companies which I visited. They were willing to take one, two, or even three days off from their busy work schedules to show a wide-eyed graduate student how aerospace systems really are designed and developed. In addition, I thank the many engineers and managers who were willing to spend an hour or two sharing the details of their work. Though these people must remain nameless in this document, I thank them all.

I also must thank my advisors. First, Professor John Deyst. John has advised me now for two years, over the course of two Masters degrees and two theses. I have valued his time and assistance and his genuine concern for my academic experience.

During this work I have also been fortunate to be able to call upon an array of additional advisors. Within LAI, I thank Professor Earll Murman, Professor Stan Weiss, and especially Dr. Joyce Warmkessel and Dr. Eric Rebentisch for their time and advice. I was fortunate enough to meet Dr. Bill Finch, a "kindred spirit" here at MIT in the realm of set-based concurrent engineering. Without his willingness to listen to my ramblings on several late nights, I would not have been able to pull this work together.

I must also thank my "advisors at large," Professor Al Ward, at the University of Michigan, and Professor Durward Sobek, at the University of Montana. Al and Durward are some of the original developers of the theory of set-based concurrent engineering, and their advice and input were essential to my work.

Finally, I must thank my family:  Mom, Pop, Jesse, and especially my wife, Ritu.  Without their support, none of this would have been possible.

# Table of Contents

# List of Figures

# List of Tables

# 1. Introduction

## 1.1 Motivation and Objectives of This Research

A new paradigm in engineering design, known as set-based concurrent engineering, has been proposed which seems to offer advantages over more traditional techniques. Traditional methods of engineering design focus on setting requirements early, converging on a design concept rapidly, and then iterating over the design until it meets all specifications. In contrast, set-based methods delay fixing requirements and suggest that it is better to design a variety of concepts which would meet a range of requirements. Over time, as the customer's needs are better understood and the design problem itself becomes more clear, the range of designs is gradually narrowed, ultimately leading to one, globally optimal design. Set-based approaches to design seem to offer advantages over other methods in terms of improved design quality, reduced development risk, and shorter cycle times.

This research project, therefore, has several goals. The first is to develop a clear understanding of the definition of set-based design -- what it entails, what benefits it offers, what requirements it places on organizations, etc. Second, based on this definition, the primary aim of this research is to assess the "set-basedness" of design practices in the aerospace industry. Finally, based on the results of the industry assessment, the thesis will propose a model for how aerospace companies might implement a design process incorporating set-based concurrent engineering. Several policy recommendations will also be proposed, including actions that the government could take to facilitate better design practices.

To facilitate these objectives, this research attempts to provide answers to the following key questions about aerospace product development practices:

- *Defining Requirements.* How are requirements usually specified -- as single values, as ranges, or as combinations of both? When are requirements set in the design process -- early, late, etc.?
- *Number of alternative concepts considered.* How many alternatives are considered during the design process? How do these alternatives differ, i.e., are completely different systems being compared or are comparisons made between "variations on a theme"? Are different alternatives considered at each level of design, i.e., complete systems, subsystems, small parts? How long are the different alternatives considered, and are multiple concepts developed in parallel? What means are used to eliminate designs from consideration?
- *Iteration.* What are the major causes of iteration in design? Do design schedules include plans for iteration? In this context, iteration refers to the negotiations which take place between functional specialties during the design process. For example, a warhead size could be made larger or smaller depending upon the accuracy of the targeting system. Are these tradeoffs considered in series (iterated) or in parallel (set-based)?

## *1.2 Thesis Overview and Outline*

This thesis is essentially divided into three parts, each corresponding to one of the three goals outlined above. Chapters 2, 3 and 4 comprise the first part. Chapter 2 presents an overview of "traditional" design methods, while Chapter 3 introduces the primary concepts behind set-based concurrent engineering. Chapter 4 then describes some features of the Toyota development process, which demonstrates many set-based methods. Chapters 5 through 10 present the industry assessment, including twelve examples of current design practices. The final part of the thesis includes Chapter 11, which presents the model for implementing set-based concurrent engineering, and Chapter 12, which discusses several policy recommendations and makes suggestions for further research.

# 2. Point-Based Approaches and Concurrent Engineering

## *2.1 Chapter Introduction*

In order to understand a new idea, it is often useful to contrast it with an old one. To that end, this chapter briefly reviews some important concepts about traditional approaches to design and concurrent engineering. The discussion of this material will then provide a point of comparison for the introduction of set-based concurrent engineering, which is explained in the next chapter.

This chapter begins with an introduction to point-based design strategies and the communications challenges posed by the separation of knowledge in complex product development. Subsequent sections then discuss concurrent engineering and some of the various tools and methods used to aide in implementing this approach to engineering design. The chapter concludes by setting the stage for the introduction of set-based concurrent engineering.

## *2.2 Point-Based Strategies*

Typical design processes can be characterized as point-based or iterative approaches: They seek to develop and select a single concept, i.e., a single *point* in the design space, as quickly as possible. In general, point-based strategies consist of five basic steps (Liker *et al.*, p. 167):

1. First, *the problem is defined*. This step typically means understanding the customer's needs and establishing product requirements.
2. Once the problem is clearly stated, engineers and designers *generate a large number of alternative design concepts*, usually through individual or group brainstorming sessions.

3. Engineers then *conduct preliminary analyses* on the alternatives, leading to the *selection of a single concept for further development*.
4. The remaining concept is then further *analyzed and modified* until all of the product's goals and requirements are met.
5. If the selected concept fails to meet the stated goals, the process begins again, either from step 1 or 2, until a solution is found.

The overall aim of these strategies is to identify the "best" solution to a design problem as early in the development process as possible, and to avoid wasting time considering other options. If the selected solution falls short of customer needs in some respect, it is modified as much as possible, or it is simply discarded in favor of a new concept.

## *2.3 Point-Based Strategies and Concurrent Engineering*

Applying point-based strategies in a concurrent engineering framework has a variety of consequences which complicate the design process. The remaining sections will, therefore, first review traditional approaches to engineering design and discuss the problem of knowledge separation in product development. The definition of concurrent engineering will then be reviewed, along with strategies and tools to help implement this design method.

### 2.3.1 Traditional Approaches to Product Development: Over the Wall

Traditional models of engineering design processes tended to group engineering specialists together, into functional groups. One functional group would begin to design a new product and would then "throw it over the wall" to another group, with little or no communication. This method led to development delays, often associated with "omitting important design considerations early in the design" (Anderson, p. 26).

This type of development process is illustrated in Figure 1. As can be seen, there are a significant number of feedback loops, and each loop is typically associated with the need to modify or rework some aspect of the design. A description of one company's method based on this process illustrates the major problems:

> A considerable amount of rework and backtracking was required to get from the initial concept to a complete production ready design. (The primary cause of the backtracking was that the design didn't fit the needs of the next specialist in the line -- "there's not enough room for my control panel," etcetera -- and was sent back for modification. A frequently employed alternative to sending the design back was to secretly redesign it.) (Womack and Jones, p. 107)

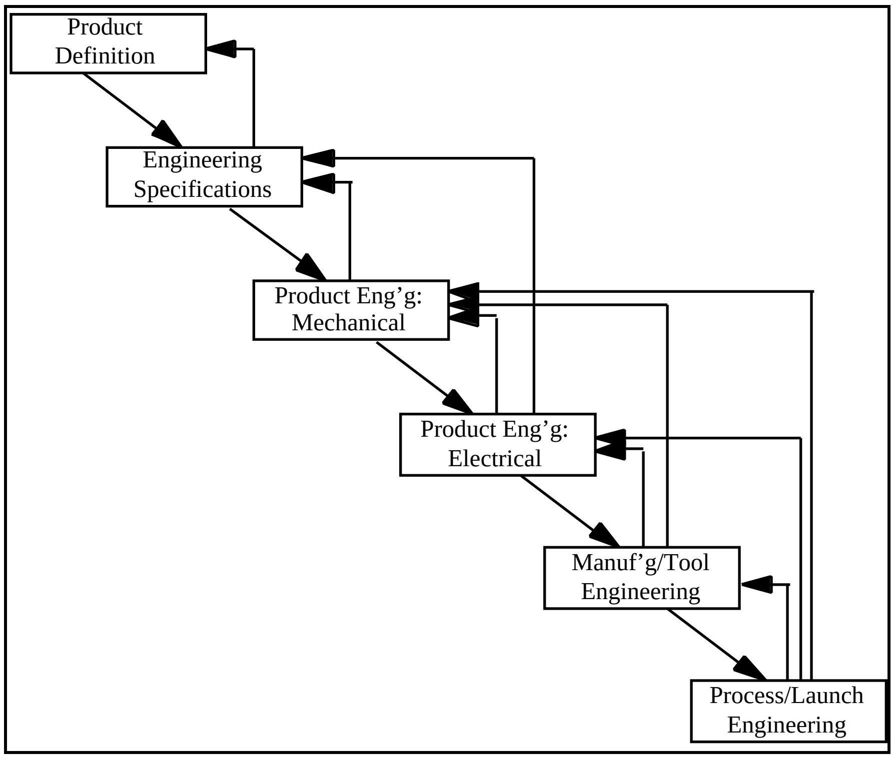



**Figure 1: "Over the wall" product development.** Traditional methods of engineering design employed little coordination between upstream and downstream tasks, leading to the need for considerable backtracking and rework in the development process. (Adapted from Womack and Jones, p. 107)

### 2.3.2 The Separation and Integration of Knowledge

This highly iterative approach to design becomes inefficient for the development of complex systems. To develop such products requires that a wide and diverse set of skills be brought to bare on a problem. These skills tend to be beyond the grasp of any one individual, and, consequently, groups of engineers and designers must bring together their individual knowledge to collectively solve a problem (Krishnan, 1997, p. 485). A critical issue in new product development, therefore, becomes the effectiveness with which engineers communicate.

As noted above, highly iterative methods tend to be inefficient at achieving such communication. This dilemma is an issue to which a substantial amount of research and literature is dedicated, and it is not the intent of this thesis to review it in detail. To understand set-based approaches, however, it is important to review several concepts in this area.

Gulati and Eppinger define three critical phases of communication: the availability of information, the transfer of that information, and the proper use and application of the information. They suggest, therefore, that not only must engineers exchange information, but that "it is important to assure that the *correct* information exchange takes place" (p. 14). But, as von Hippel (1990) notes, when problem solving extends beyond a single person, organizational and physical boundaries can degrade cross-boundary communication and coordination, reducing the effectiveness of engineers' attempts to share their knowledge (p. 409). Thus, in addition to designing a product, a product development team must design its organization to facilitate the proper information flow.

This information flow is complicated, however, by what von Hippel (1995) defines as "sticky information." Sticky information is typically in the form of tacit knowledge -- rules of thumb or guidelines -- which a person or organization takes for granted and would not necessarily think to communicate to others. Though von Hippel describes this theory in the context of a user or customer working with a manufacturer, the theory can also be applied to multidisciplinary design. In this context, the "user" is a downstream design process, such as production system design, and the "manufacturer" is an upstream process, such as conceptual design. Continuing the analogy, upstream design processes can be viewed as manufacturers of design information, and downstream design processes can be thought of as users of that information. To develop a product, however, requires the combination of both groups' knowledge.

The difficulty posed by sticky information is that it limits the ability of two groups to effectively communicate, reducing the success of their combined efforts (von Hippel, 1995). The problem is not that the two groups are unwilling to assist one another or that they do not possess the knowledge required for joint problem solving. Instead, the problem is either that one group may not realize that the other group possesses some needed knowledge, or that one group can not effectively communicate some knowledge to the other. The basic issue, therefore, becomes how to best transfer knowledge from one group to another.

Von Hippel identifies four strategies to cope with the problem of sticky information, and these concepts can be adapted for use in a multidisciplinary engineering environment. The first two, upstream-based design and downstream-based design, are illustrated in Figure 2 and Figure 3, respectively. In upstream-based design, knowledge possessed by the downstream group is transferred to the upstream group. The upstream group then uses this knowledge along with their own to develop a design solution. To achieve this knowledge transfer generally requires that the downstream group (or a portion of it) be co-located with the upstream group, at least temporarily. Once the downstream group's knowledge has been incorporated by the upstream group, the need for co-location ends. Downstream-based design is essentially the same, except in this case upstream knowledge is assimilated by the downstream process, which then manages the design effort.

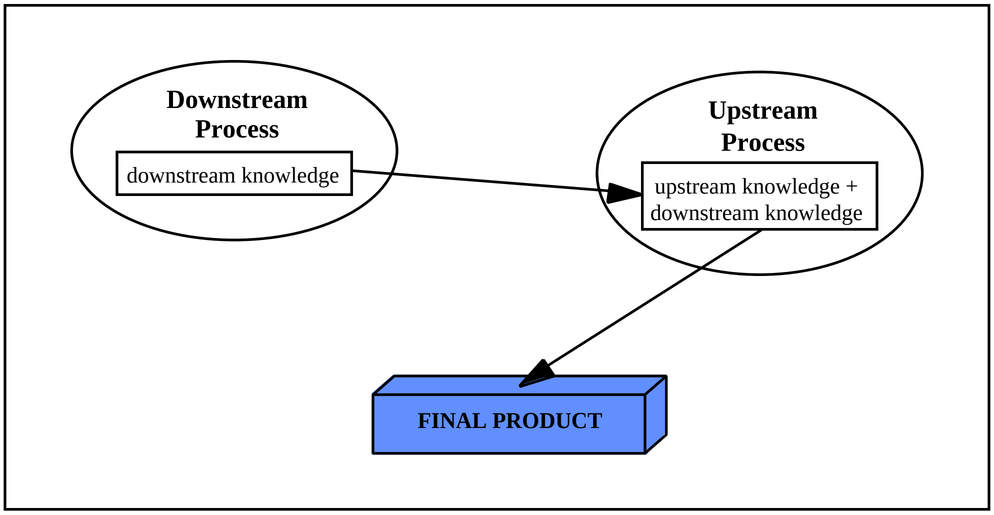



**Figure 2: Upstream-based design.** Downstream design knowledge is assimilated by the upstream design group, allowing the upstream group to possess all of the knowledge required to solve the design problem. (Adapted from von Hippel, 1995)

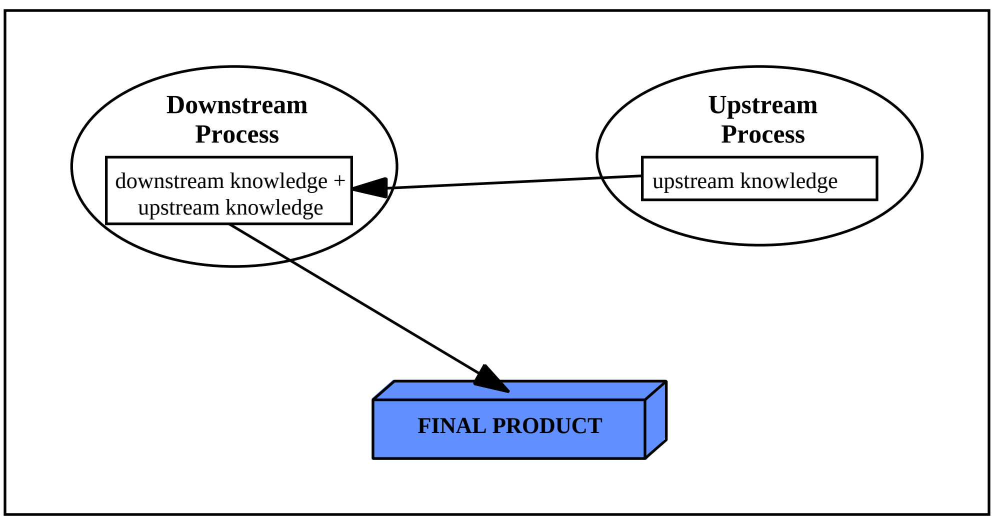



**Figure 3: Downstream-based design.** Upstream design knowledge is assimilated by the downstream design group, allowing the downstream group to possess all of the knowledge required to solve the design problem. (Adapted from von Hippel, 1995)

A third strategy is iterative design, shown in Figure 4. In this strategy, the upstream group develops a prototype of a design, which is then sent to the downstream group. The downstream group experiments with the prototype, providing feedback to the upstream group. The upstream group modifies the design and the prototype, which is again reviewed by the downstream group. The process continues until each group is satisfied with the design. Note that in this method, as opposed to the previous two, the location of the development work must switch back and forth between groups. This changing emphasis from one group to the other can result in waste, both in time and money, as design activities are started and then stopped at each group (von Hippel, 1995).

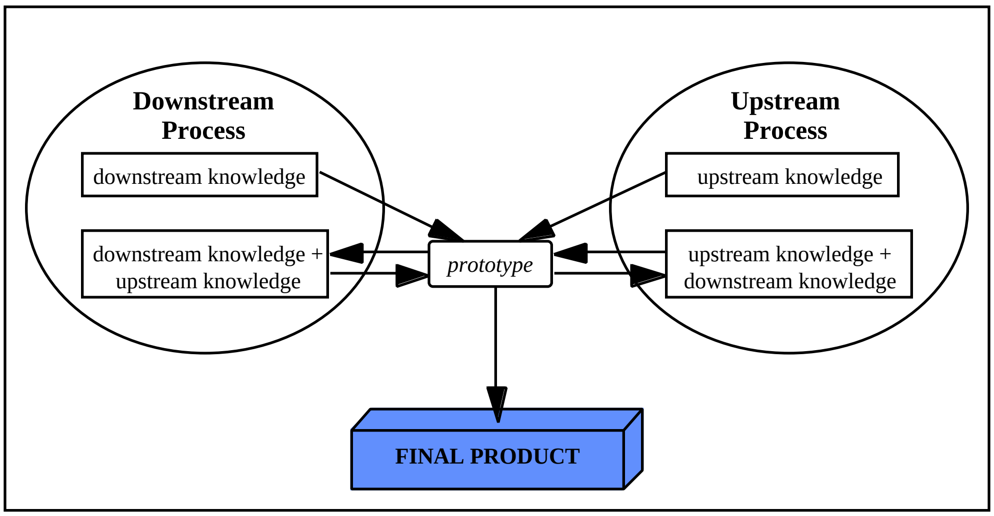



**Figure 4: Iterative design.** The upstream design group first develops a prototype which the downstream group reviews and modifies. The process continues until both groups are satisfied with the design. (Adapted from von Hippel, 1995)

The final strategy is sub-problem design. Illustrated in Figure 5, this method decomposes the initial design problem into several smaller ones, each of which can be independently handled by either the upstream group or the downstream group. These independent solutions are then combined to form the final solution.

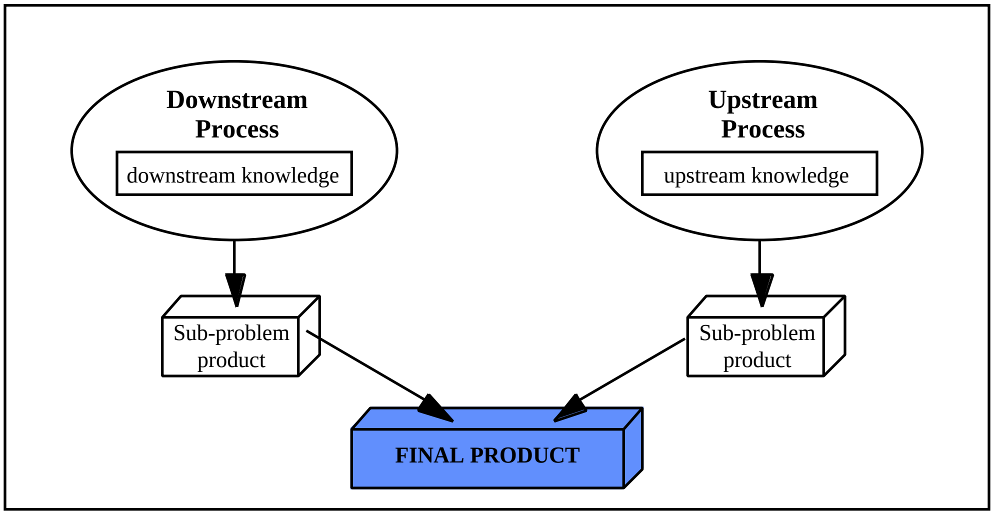



**Figure 5: Sub-problem design.** The design problem is decomposed into smaller problems which each group can solve independently. (Adapted from von Hippel, 1995)

The decision as to which method to choose for a given project is based primarily on the costs and feasibility of transferring knowledge between the upstream and downstream groups (von Hippel, 1995). If this knowledge transfer is easy, a downstream- or upstream-based approach would most likely work best. When knowledge transfer is difficult, on the other hand, iterative or sub-problem design would provide the best strategy. These methods illustrate that the interactions between two design groups should be based both on the sequence in which their design work must be completed (upstream vs. downstream) and also on the ease or difficulty associated with transferring knowledge between the groups.

In a similar vein, Wheelwright and Clark define four modes in which upstream-downstream interactions might occur, illustrated in Figure 6. In mode 1, the upstream group does not communicate with the downstream group until it has completed its work, and the downstream group does not begin to do any work until it receives this communication. Mode 2 is similar, except in this case the downstream group makes assumptions about the upstream group's work. The downstream group then begins its design process prior to actually receiving any information from the upstream group. Though this mode could be faster than the first, the downstream group runs a high risk of having to redo a significant amount of work if their assumptions prove incorrect. Modes 3 and 4 increase the intensity of the communication between the upstream and downstream groups, and in mode 4 this communication occurs much earlier than in other modes. This final mode of interaction, mode 4, is the one which best facilitates the incorporation of upstream and downstream knowledge into a product's design.

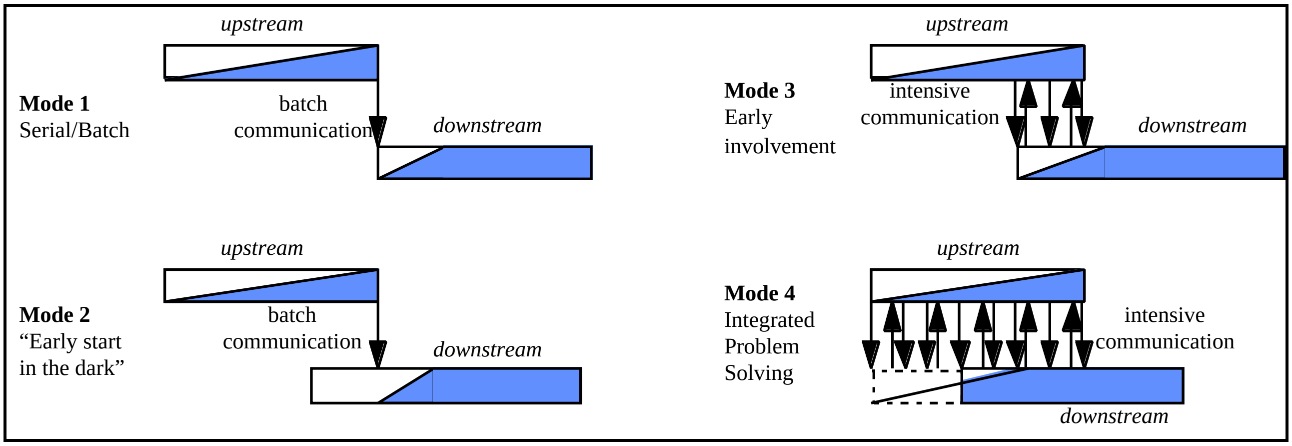



**Figure 6: Modes of upstream and downstream interaction.** Modes 1 and 2 permit little (if any) interdisciplinary problem solving between upstream and downstream groups, while modes 3 and 4 allow progressively greater interaction. (Adapted from Wheelwright and Clark, p. 178)

### 2.3.3 Transitioning to Concurrent Engineering

Based on these models of communication, it is clear that companies could improve their product development processes by facilitating better interactions between their upstream and downstream design groups. To this end, many companies have moved to implement concurrent engineering (CE) and cross-functional design teams.

Smith (p. 67-68) defines concurrent engineering in terms of four principles:

- an increased role for manufacturing process design in product design decisions;
- formation of cross-functional teams to jointly develop new products and processes;
- a focus on the customer throughout the development process; and

- the use of lead time as a source of competitive advantage.

Hauptman and Hirji broaden this definition slightly, such that CE intends to address as many downstream issues as possible early in the design process: Concurrent engineering is "the integrated and parallel design of products and their related processes, including manufacturing, test and support" (p. 154). By facilitating such integrated design, concurrent engineering shortens development lead times. Note, however, that this decreased development time is not achieved by reducing the time required to complete design tasks, but rather by executing these tasks simultaneously (Hauptman and Hirji, p. 154). This reduction in lead time can be significant and represents a competitive advantage to those firms which can apply it well.

To facilitate CE, design organizations have moved away from the traditional functional configuration to adopt cross-functional design teams or integrated product teams (IPTs), illustrated in Figure 7. The primary aim of assembling such a cross-functional team is to enable a variety of perspectives to develop a product design concurrently, so that the final product is better integrated, and is developed more quickly and at a lower cost (Anderson; Krishnan, 1992; Pomponi). These interdisciplinary teams typically include engineers from upstream functions (such as design) as well as downstream functions (such as manufacturing). In addition, representatives from support groups such as marketing, test, finance, etc. also participate on the team (Anderson).

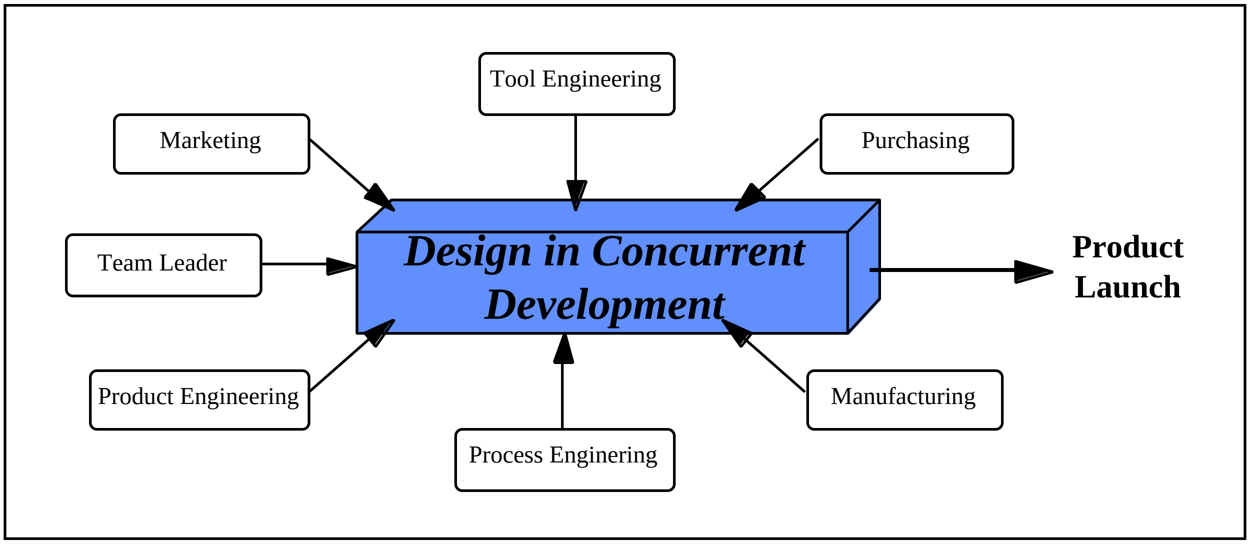



**Figure 7: Design in the CE environment.** In concurrent engineering, both upstream and downstream design groups simultaneously participate in the development of a product. (Adapted from Womack and Jones, p. 120)

Womack and Jones further characterize cross-functional, concurrent engineering and design as lean product development. They note that an ideal design process would operate much like a single-piece flow in a manufacturing system. Such an analogy suggests that in a lean development process, a new product design would move continuously from concept to production, without stopping due to bureaucratic needs and without backflow to correct mistakes (Womack and Jones, p. 119). As discussed earlier, a functional organization, with its strict separation of engineering specialties, often requires such backflow and rework. By allowing for direct communication between specialties, however, concurrent engineering moves closer to this continuous flow model. A product design no longer needs to be passed around to several independent engineering departments, but is instead worked upon by multiple engineers within the same team.

In terms of the communication models described previously, cross-functional teams and IPTs facilitate mode 4 interactions (integrated problem solving), and are a means of implementing upstream-based design (downstream groups are co-located with upstream groups). Many design organizations also make use of prototypes during product development efforts, enabling cross-functional teams to use iterative design methods in conjunction with the other strategies. These cross-functional teams are, consequently, the key enabler in concurrent engineering: They bring together the individual knowledge of engineers required to develop a single, integrated design solution.

### 2.3.4 Interteam Communication: An Integration Problem

Typically, a cross-functional design team might be established to develop both a single product and its manufacturing plan. For more complex systems, the system itself might be decomposed, i.e., broken into several smaller elements, and an IPT assigned to develop each element. In such cases, not only must multiple skills be brought together within an IPT, but multiple skills must be brought together across several IPTs.

Under these conditions, the communication problem can viewed as an integration problem, i.e., how to integrate the knowledge of these multiple individuals or groups. Browning (1997) thus defines integrative mechanisms (IMs) as “strategies and tools for effectively coordinating actions between multiple” design organizations, such as integrated product teams (p. 86). These IMs have two primary purposes: to facilitate information flow and to regulate information flow (Browning, 1997). Examples of IMs include information and communication technologies (such as linked CAD tools and common databases), co-location, face-to-face meetings, manager mediation, and interface contracts and scorecards (Browning, 1997). Note that the goal of all of these methods and tools is the same: to help ensure that the right information is delivered to the right people at the right time and in the proper amount. These IMs are used to supplement the basic structure of the design team so as to further enhance the members’ abilities to share and combine their knowledge.

### 2.3.5 Task Sequencing

Despite the effort placed on integrated problem solving, concurrent engineering, and the associated design tools and methods, many problems in engineering design are coupled; that is, one design group is dependent upon information from another group (Krishnan *et al.*, 1992). Examples of such coupling abound in the aerospace environment: in aircraft design, for example, engine selection is tied to wing design, while in spacecraft design, solar array sizing is linked to antenna size and power. In such cases, Group *A* might depend on Group *B* to provide data on variables *x* and *y*. Group *B*, however, might require information *w* and *z* from Group *C*. But, to provide this data, Group *C* might require Group *A*'s results (Browning, 1997, p. 83). Such interdependent, chicken and egg problems are difficult to sort out, because it is "rarely possible to identify an unambiguous sequence for making decisions" (Chang *et al.*, p. 212). The concurrent environment previously described can, therefore, be thrown into disarray, while each engineer waits for information from other engineers.

To cope with these coupled design problems, many firms resort to a sequential decision making strategy (Krishnan et al., 1997, p. 485). Such strategies require that a design problem first be partitioned -- divided up -- into a "number of subtasks that may then be distributed among a number of individuals...[or] firms" (von Hippel, 1990, p. 407). Once a problem is partitioned, a tool such as a design structure matrix can be used to sequence the tasks to minimize the impact of iteration in the design (Browning, 1996). By properly sequencing tasks, the costs of downstream changes to a design can be minimized (von Hippel, 1990 p. 409).

The drawback of such tools, however, is that while they might minimize the extent of the changes which need to be made during an iteration, they may not help engineers determine the best sequence in which to make decisions (Krishnan *et al.*, 1991). As decisions are made in series, choices made by upstream engineers will constrain the design options available to downstream engineers. Depending upon the nature of these constraints, the design quality of the product might suffer or the product might fail to meet requirements associated with downstream issues (Krishnan *et al.*, 1991). Thus, not only should engineers attempt to sequence their decision-making strategy to limit the extent of iterations, they should also initially sequence the decisions to limit the loss of design freedom by downstream designers (Krishnan *et al.*, 1997, p. 488). By properly sequencing the steps in a design process, the coupling between problems can be reduced, allowing more effective use of other integrated problem solving techniques, such as the ones noted above.

### 2.3.6 Establishing Requirements in the Point-Based Approach and Doing It Right the First Time

To help cope with all of the difficulties associated with both integrating information across design groups and limiting the effects of iteration, conventional practice suggests that it is best to establish firm requirements early in the development process (Ward *et al.*). Requirements are finalized quickly to impose “as much constraint as possible in order to simplify the interactions among subsystems” and other members of a design team (Sobek, 1997, p. 224). The logic behind this approach is that since one design group does not necessarily know or understand the constraints faced by another group, each group must specify their subsystems in great detail to ensure their functionality and that they interface properly with other systems (Sobek, 1997, p. 224).

Therefore, conventional approaches to design, which rely on iteration to develop a product, lead to goals of freezing requirements and specifications as early as possible and philosophies of “doing it right the first time” (Ward *et al.*, p. 48). This reliance, however, leads to two paradoxes. The first paradox relates directly to the development of requirements: System design methods emphasize establishing requirements early, but iterative methods imply that the requirements will change over the course of successive iterations (Ward, 1990, p. 50). Solving this paradox is not a simple matter. Flexible requirements, for example, run exactly counter to the idea of finalizing requirements early, yet allowing a firm requirement to change could invalidate previous design decisions (Ward, 1990). To facilitate iterative approaches to design, therefore, engineers and managers continuously emphasize the need for firm requirements at the very start of a program, but must also be adaptable as the requirements change over a product's development.

The second paradox relates to the very reason for initiating a development effort. Reinertsen states that "the purpose of a design process is to generate information" (p. 11). An efficient design process, therefore, is one which generates information cost-effectively. When developing a design, the feasibility and appropriateness of the concept is determined through testing, so to be cost-effective, each test should generate as much information as possible. When conducting tests, more information is contained in the results of a failure than in the results of a success (Reinertsen). A design process, therefore, does *not* maximize the amount of information which it generates by maximizing test success rates, but by ensuring "an adequate failure rate to generate sufficient information" (Reinertsen, p. 71). A philosophy of "do it right the first time" implies that a design would always successfully complete its development tests. Such outcomes, however, would not necessarily produce information in a cost-effective manner. The second paradox of iterative methods, therefore, is that a "do it right the first time" mentality actually decreases the cost-effectiveness of a design process by degrading the amount of information which the process produces.

These two paradoxes limit the efficiency and effectiveness of iterative design methods. While engineers and managers have developed many tools to aide in coping with these problems, current engineering methods do little to actually eliminate the sources of these paradoxes.

## 2.4 The Risks and Limitations of the CE Solution

While IPTs, IMs, task sequencing, and early requirements definition help to make CE approaches effective, working in a non-serial, concurrent environment still introduces additional risks to the development process. A CE-approach to product and process development "requires that the downstream phases be able to operate... using early upstream information" (Krishnan *et al.*, 1996, p. 210). Making use of such information entails risk because the information may not be finalized (Chang *et al.*; Krishnan *et al.*, 1996). Changes which are made to this information, therefore, force changes to both the upstream phases providing the information and the downstream phases attempting to make use of it.

The continued reliance on iterative, point-based strategies, therefore, represents a limitation of current CE approaches to design. When multiple views are involved in developing a new product, each group will recommend changes to a proposed solution to better reflect that group's constraints and requirements (Sobek, 1997, p. 201). This scheme of proposition and change, however, leads to several possible problems. Since each group in the design process does not necessarily understand the limits and needs of every other group, recommended changes can produce conflicts, leading to waste in the development process (Sobek *et al.*, p. 16). Further, "the more tightly parts [of a product] are coupled to other components, the greater the impact changes in one component have on other components, and the more rapidly and pervasively such changes propagate through the system" (Liker *et al.*, p. 170).

There is the potential, therefore, that a change made by one group in a design team could invalidate previous decisions made by other groups (Ward *et al.*, p. 58). This potential for work to become obsolete tends to "deter simultaneous design" because engineers would rather wait to make a decision than have to redo their work (Liker *et al.*, p. 165). Work that could be conducted in parallel then reverts to a truly sequential pattern, and the emphasis in project planning becomes the establishment of the proper sequence in which to make decisions (Liker *et al.*, p. 167). Finally, there exist no guarantees that the iterative, point-to-point approach will ever converge on a final solution (Sobek *et al.*, p. 7). Instead, the iterative loops are never quite closed, and, in the worst case, "problems are resolved when production begins, by selecting whatever last minute compromise is easiest" (Ward, 1990, p. 50).

### 2.5 Summarizing the CE Solution

Although concurrent engineering and IPTs have dramatically changed (and improved) engineering design, they have not significantly altered the *nature* of the design process. Non-CE methods were typified by one engineering group throwing a design over the wall to another group. What CE and IPTs have done is to “lower the wall:” upstream design groups now receive quick and extensive input and feedback on their design decisions from downstream organizations. But, as illustrated in Figure 8, *within the team* the nature of the design process has not changed: one group or person establishes requirements, another proposes a design solution, several others make comments about and recommend changes to the solution, etc. (Liker *et al.*, p. 165).

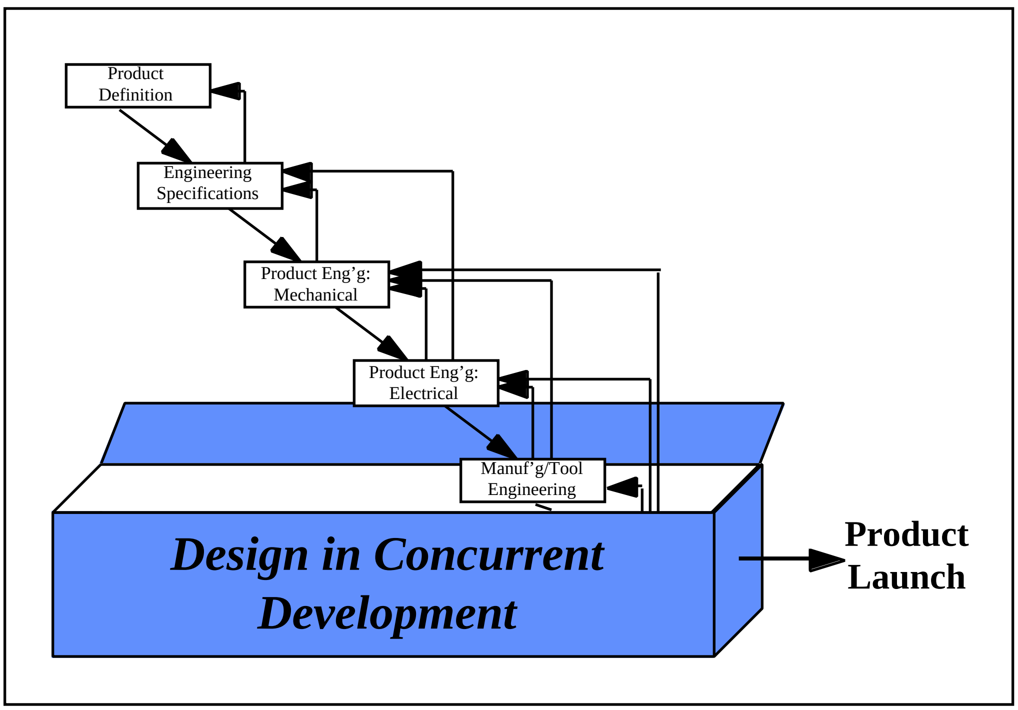



**Figure 8: Looking inside the black box of CE.** Looking inside the black box of concurrent design reveals that typical CE practices have simply moved feedback loops upstream in the development process, rather than altering the nature of the design process.

Rather than fundamentally altering the nature of engineering design, the effect of concurrent engineering and IPTs is to move engineering feedback loops upstream in the design process

(Sobek *et al*., p. 7), and to tighten and shorten these loops (Liker *et al.*, p. 165). By decreasing the amount of time required for changes to circulate through a design team, CE- and IPT-based solutions aim to limit wasted time and effort spent pursuing designs which will not meet downstream requirements (Sobek, 1997, p. 226). The nature of the interaction between upstream and downstream groups, however, has not changed.

# 3. Defining Set-Based Concurrent Engineering

## *3.1 Chapter Introduction*

Concurrent engineering has improved product development processes significantly, but it has not altered the fundamental nature of the interactions which go on during that process. This chapter introduces the primary concepts of set-based concurrent engineering, and illustrates how set-based concepts have the potential to fundamentally change the manner in which engineers interact.

The chapter begins by introducing several concepts which suggest why making decisions early, a core tenant of point-based methods, is not always the best strategy in product development. The principal concepts of set-based concurrent engineering are then summarized, followed by a more detailed description of those concepts. Other design methods which support the use of sets are then discussed, as well as how these methods, though different, are compatible with set-based concurrent engineering. The chapter concludes by summarizing the principles of set-based concurrent engineering, providing the foundation for the remainder of the thesis: examples of set-based practices and how set-based concurrent engineering could be applied in the aerospace industry.

## *3.2 The Triple Problem: Factors Motivating the Need to Delay Decision-Making*

In addition to the challenges posed by attempting to work in a concurrent environment, design is complicated by three other factors: the evolution of a product's cost, management's ability to affect these costs, and the evolution of designers' knowledge about a design problem. As will be discussed, these factors all lead to a direct conflict with traditional concepts of design: Rather than making decisions as early as possible, there are advantages to making decisions as late as possible.

Consider first the issue of a product's cost. As a new product is designed, engineers will make decisions affecting how much the product will cost: how expensive it will be to manufacture, for how much it will have to be sold in order to earn a profit, how much the product will cost to support and maintain, etc. The difficulty associated with these decisions, however, is that "[t]he earliest decisions about designs have the largest impact on the ultimate quality and cost, but these decisions are made with the least data" (Ward, *et al.*, p. 59). As shown in Figure 9, sixty percent of a product's total life-cycle costs have been designed into the product by the end of its conceptual development -- even before preliminary or detail design has begun (Anderson, p. 133). Thus, decisions made very early in the product's development will have long-lasting consequences on the total cost of the system, while decisions made late in its development will have little success in lowering these costs (Anderson, p. 230).

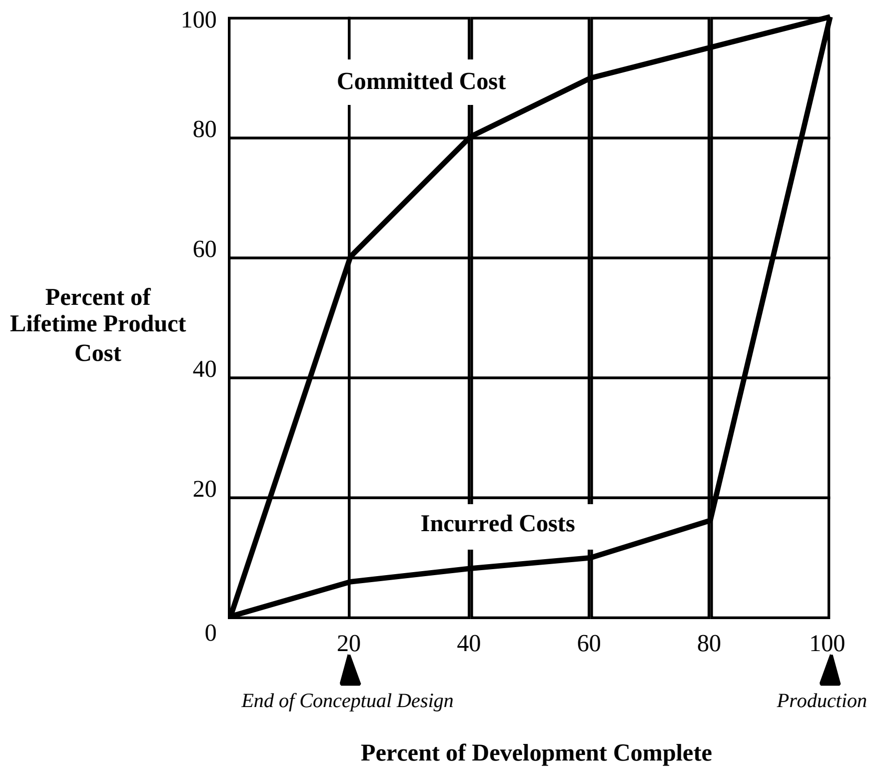



**Figure 9: Designing-in costs.** Although the majority of the costs associated with a development program are not incurred until late in the project, costs are committed to the product's lifecycle very early. (Adapted from Anderson, p. 132)

This problem becomes even more difficult when one considers management's ability to influence a product's design. As illustrated in Figure 10, management can exert its greatest influence on a product early in the development cycle. As the cycle progresses, however, this power is greatly diminished (Wheelwright and Clark, p. 33). In addition, the costs associated with making any changes to a product's design rise exponentially during the design process (Reinertsen, p. 14). As Anderson summarizes, "*The further one progresses into a design, the harder it will be to start satisfying additional needs*" (p. 229). This trend appears because every decision made by engineers constrains the options available for future decisions (Anderson; Krishnan *et al.*, 1991). Thus, late in the design process, engineers have already made numerous decisions, severely limiting the alternatives available at that time. As a consequence, not only do early decisions have long-term effects on a product's costs, but management's power to influence these costs declines rapidly as the product develops.

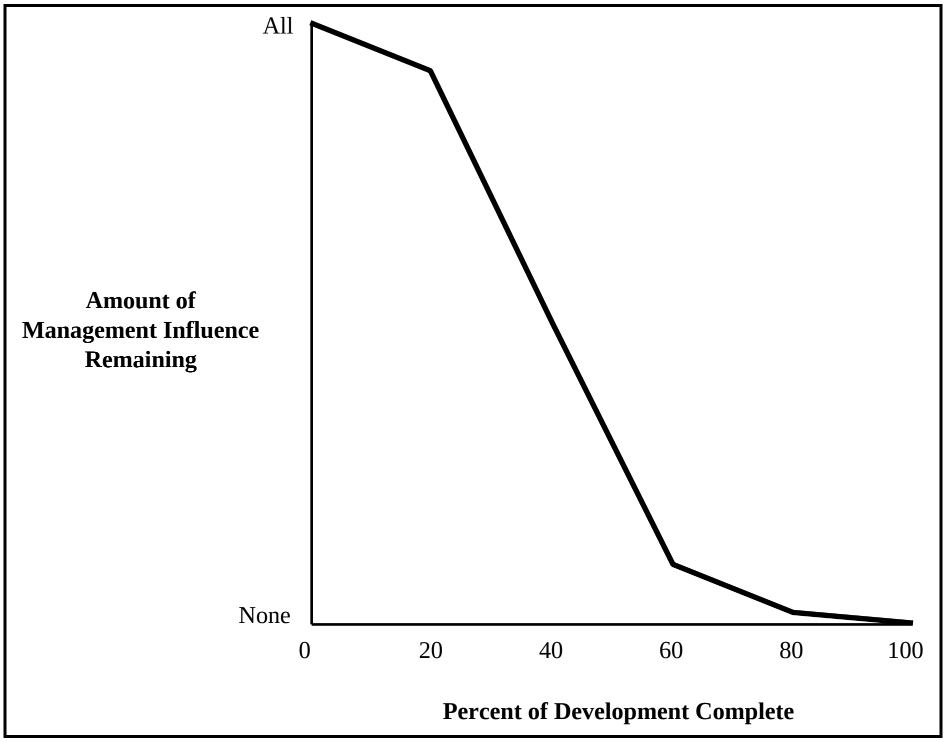



**Figure 10: The diminishing power to make changes.** The further a development project progresses, the less power both managers and engineers will have to influence its final outcome. (Adapted from Wheelwright and Clark, p. 33)

These factors conspire to make early design decisions more powerful than later ones. But they are further complicated by yet a third factor: lack of knowledge. As shown in Figure 11, early in a product's development (such as during conceptual design), engineers' and managers' knowledge about virtually every aspect of the product will be at a minimum (Reinertsen, p. 15). Their early decisions, therefore, will be relatively uninformed. As they "work the problem" and become more familiar with the constraints which they face and better understand the customer's needs, they will be able to make better design decisions. But, as was just discussed, it is the early decisions which will have the most dramatic effects on the long-term costs of the product.

Engineers and managers, therefore, are placed in the awkward position of having to make high-leverage decisions with very little knowledge.

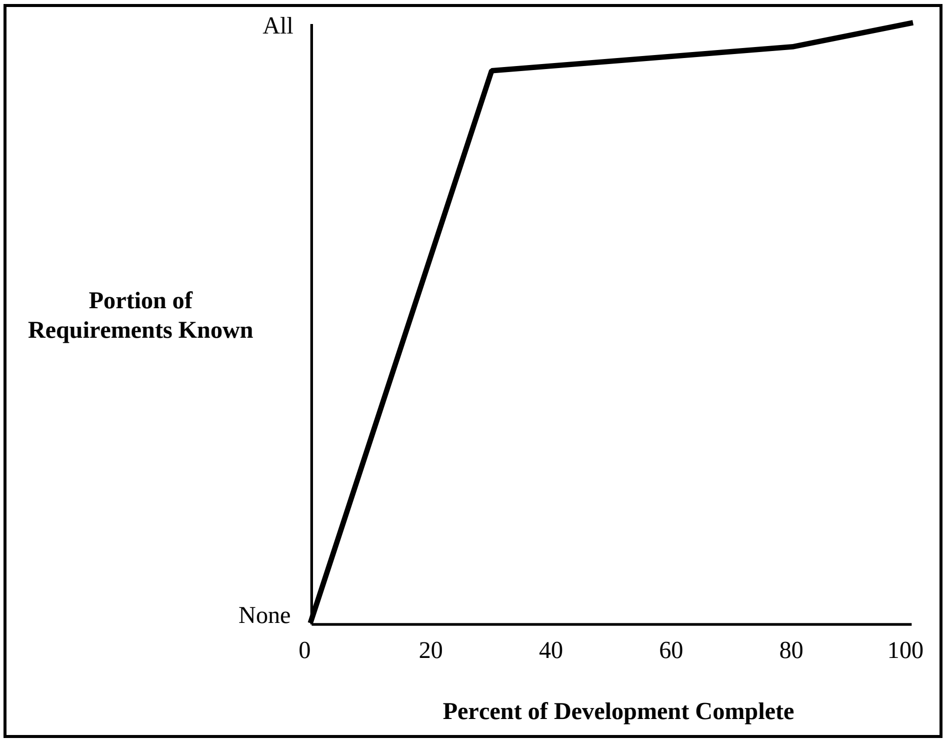



**Figure 11: Evolution of design knowledge.** Though it increases fairly rapidly, designers' knowledge about a new product is quite low early in the product's development. (Adapted from Reinertsen, p. 15)

In summary, early decisions have the greatest effects on a product's costs, yet these decisions are based on the least information. When engineers and managers have acquired the knowledge to make better informed decisions, however, their ability to significantly affect the product's costs is severely limited.

As shown in Figure 12, a useful advance in product development practices would either delay the commitment of these costs until greater knowledge was available or would increase managers' ability to affect the design late in the process. (This proposition assumes, of course, that one cannot know the future, and therefore, cannot increase the rate at which knowledge about the new product becomes available.) In fact, Kalyanarum and Krishnan go so far as to suggest that an ideal development program would enable designers to adapt a product to current market realities at the time of the product's launch (p. 277). They note additional benefits of such delayed decision making would include:

- allowing the product to achieve a better balance between what the customer desires and what is technically feasible;
- allowing for the inclusion of the latest technology; and
- allowing competitive products and changes in customer desires to be better tracked (Kalyanarum and Krishnan).

While point-based strategies have continuously emphasized making decisions as early as possible, it is clear that there are equally valid reasons to delay decision making.

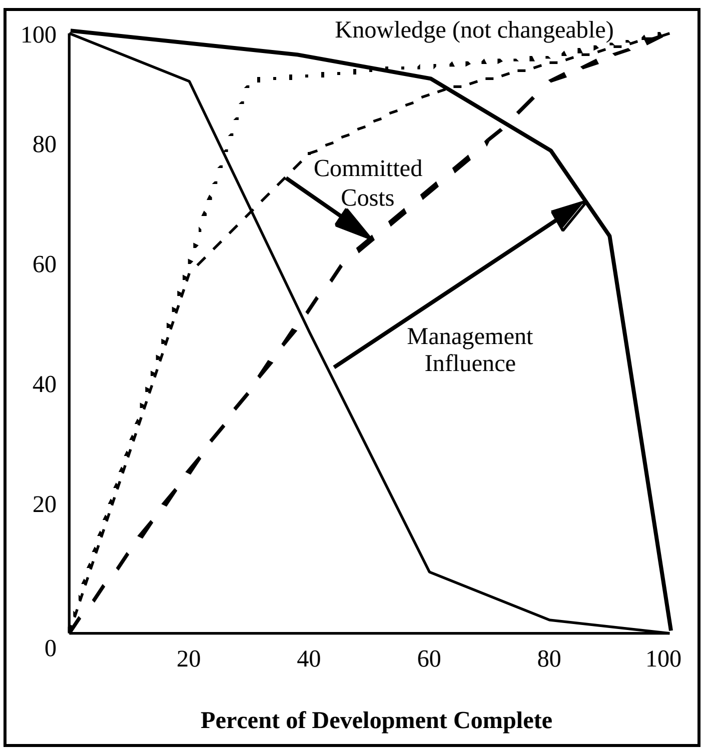



**Figure 12: Advancing product development practices.** Useful advances in product development practices would help to both delay the commitment of costs to a product and to increase management influence late in the development cycle.

### 3.3 The Need for a Paradigm Shift

The intent of concurrent engineering has been to increase the influence of downstream groups on upstream design decisions and to improve the quality of critical, early design decisions. In an ideal environment, concurrent engineering also intends for downstream groups to work simultaneously with upstream groups. The reliance of point-based methods on iterative techniques, however, increases the risks associated with this parallelism. In addition, iterative methods lead the paradoxical situation of needing to establish requirements early, yet knowing these requirements are likely to change as iterations are completed. They also seek to develop a design "right the first time," though such an approach is not necessarily cost-effective. Finally, point-based approaches require that decisions be made as early as possible, even though there can be benefits to delaying design decisions. As Liker *et al.* note, "Full implementation of CE involves a corresponding revolution in the underlying paradigm of design" (p. 165). Set-based concurrent engineering is offered as such a paradigm shift.

### 3.4 Set-Based Concurrent Engineering Defined

#### 3.4.1 An Introductory Summary

Sobek (1997) summarizes the definition of set-based concurrent engineering (SBCE) as engineers and product designers "reasoning, developing, and communicating about sets of solutions in parallel and relatively independently" (p. 202). This definition is best understood by analyzing it one piece at a time. The first component of SBCE is to develop sets of designs, i.e., groups of design alternatives, for a given design problem. Rather than trying to identify one solution, engineers should instead develop a variety of design options, and then gradually eliminate alternatives, until only one option remains.

The second component of SBCE involves inter- and intra-team communication for concurrent engineering. Given that a complex design problem will require the involvement of multiple engineers or functional groups, SBCE suggests that these groups should each develop sets of solutions to the problem from their own perspectives. The groups then interact by comparing these sets, looking for regions of overlap in their design alternatives. These regions of overlap are then narrowed in parallel, until one solution remains.

By using sets, therefore, engineers are able to implement a design strategy very nearly akin to von Hippel's sub-problem design. As was noted during the discussion of von Hippel's work, sub-problem design is best suited to design problems in which it is difficult to share knowledge between groups. The knowledge possessed by highly specialized engineering functions often has this characteristic, and, therefore, sub-problem design is an appropriate strategy. Independent exploration of design sets enables several engineering specialties to consider a design problem from their own perspective (i.e., to allow each specialty to work on a sub-problem) and then to effectively re-combine those independent alternatives into an integrated final solution. These concepts are illustrated in Figure 13, and the following sections will explore these ideas in greater detail.

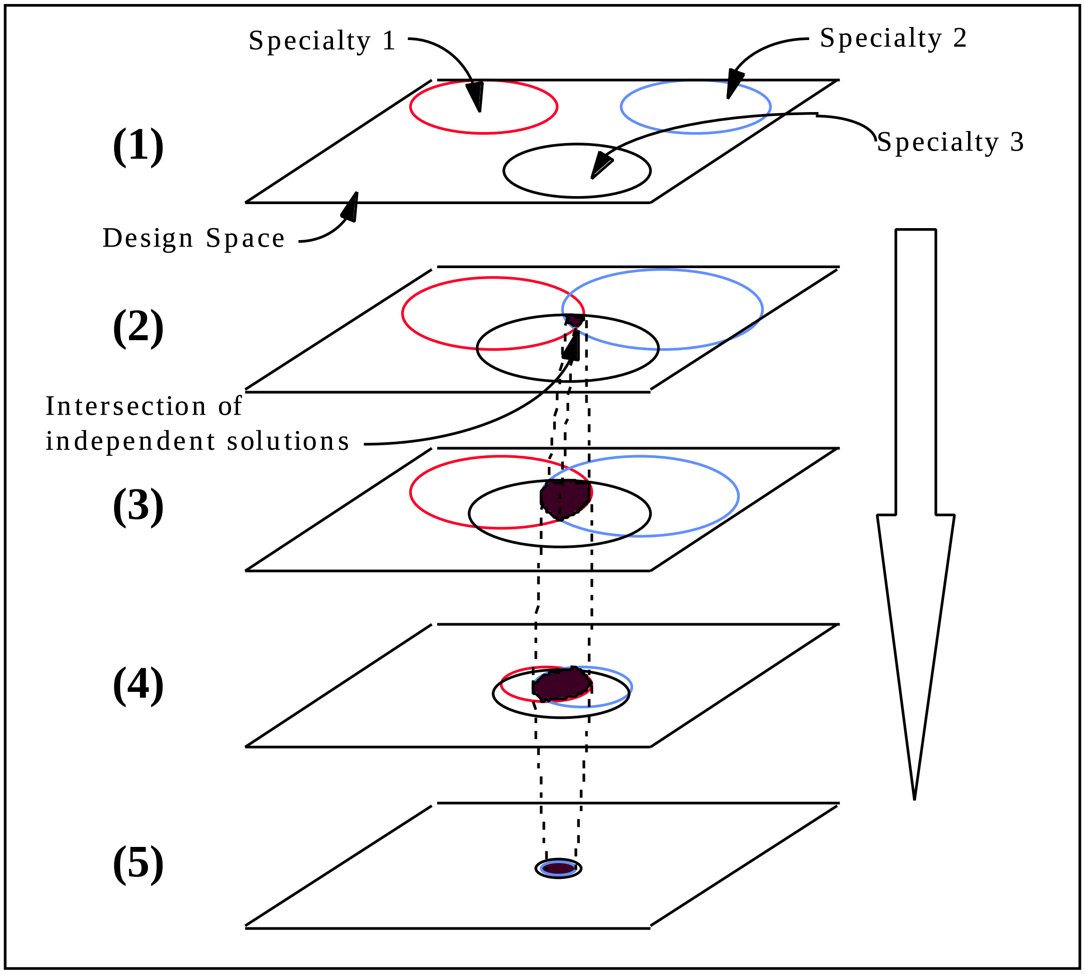



**Figure 13: Set-Based Concurrent Engineering.** (1) Three specialties, or functional groups, are illustrated within the design space (which contains all possible solutions) for a product development problem. (2) First, the specialties expand the number of options which they consider, establishing a small region of overlap between their design solutions. (3) They work together to expand this region of overlap, increasing the number of solutions which will satisfy all of the product's requirements. (4) The specialties then begin to eliminate options, and the region of overlap shrinks. (5) The solution space then is narrowed until only one design remains, that design being the final solution. (Illustration concept developed with Dr. William Finch.)

### 3.4.2 Developing Sets of Alternatives

As noted, the first element of SBCE is the development of sets of alternative solutions for a design problem. These sets might include several discrete design options or a range of parameter values (Liker *et al.*, p. 167). For example, a set might consist of one aircraft design based on a canard-wing configuration, one based on a conventional tail arrangement, and one based on a V-tail, while another set might simply vary the surface area of the V-tails.

To arrive at these sets of options, SBCE approaches the initial concept development stage of design differently than do point-based methods. In a point-based approach, initial development activities typically seek to define several potential *solutions* to the design problem. Set-based approaches, in contrast, seek to define *regions* of the design space: to characterize boundaries of current capabilities and to discover where these boundaries can be expanded (Sobek, 1997, p. 205). The emphasis is not on finding a solution, but rather on *defining what is possible*. An initial set, therefore, is likely to be relatively large and to contain a number of possible design solutions.

### 3.4.3 Using Sets to Communicate and Guide Development

Once an initial set has been developed, engineers begin to narrow the set, gradually eliminating design alternatives. As the set narrows, engineers also increase the level of detail in their design work, so that “the resolution of each idea or design grows sharper” (Sobek *et al.*, p. 22). An important principle to which to adhere during this process is that of “establishing feasibility before commitment” (Sobek *et al.*, p. 22). Before committing to keep a certain design concept, engineers must ensure that, at the current level of detail, the design is in fact feasible, i.e., that it does meet the product’s requirements. The goal of this principle is to avoid uncovering problems late in the design process.

Note, however, that the intent is *not* to necessarily test every concept to the highest level of detail possible. On the contrary, designs should only be tested to the level of detail appropriate to the size of the set. If the set is large, implying that the product is still early in its development, tests should be simple and quick, just detailed enough to expose problems that are near the surface. This testing philosophy is consistent with Reinertsen’s notion that cost-effective tests are those that demonstrate failures rather than successes. Then, as the set narrows and the level of detail in the designs increases, the fidelity of tests should also increase. SBCE, therefore avoids the less cost-effective approach of attempting to "do it right the first time." In addition, the size of the set provides a clear guide to analysis groups regarding how detailed their tests should be.

Sets are also useful in coordinating the design activities of multiple design groups. As noted previously, SBCE suggests that each functional group should be developing its design solution relatively independently. But as Krishnan *et al.* (1992) note, "the decisions made by the various functional decision makers are generally coupled and often in conflict," (p. 2), and, consequently, "weakly coordinated cross-functional [teams] can perform worse than fully integrated teams..., because... confusion can arise from conflicting goals and cause iterations, delays, and rework" (p. 1). Therefore, communication between functional groups must not only transfer design knowledge, it must also help to coordinate the groups.

The narrowing process used in set-based concurrent engineering helps to facilitate such coordination. Sets help to organize the evolution of a design because the alternatives discarded by one designer guide the decisions of other designers (VanDyke Parunak *et al*., p. 289). In addition, as designs are eliminated in stages, engineers have "time to consider the most important alternatives more fully... [allowing] them time to react and to influence one another's narrowing process" (Sobek, 1997, p. 218). SBCE thereby provides a communication mechanism that is naturally suited to coordinating diverse groups of engineers and designers.

Set-based concurrent engineering also increases the likelihood that engineers communicate the proper information between groups. Chang *et al.* note that engineers "need to provide feedback to the designers of neighboring components about their preferred values for… connecting variables and the cost of deviation from those values" (p. 214). Rather than only communicating a single design concept, engineers using set-based practices communicate about several. This set then provides information about the limits and boundaries faced by each group, allowing designers to better understand what decisions they can make to accommodate the needs of other engineers. Further, Liker *et al.* state that when communicating between functional groups early in the design process, "[p]artial information is most useful if it involves boundaries on parameters" (p. 170). Such bounding information is exactly the type of knowledge contained within a set.

Set-based communication thus directly aides in the implementation of concurrent engineering by improving the use of early design information. As discussed earlier, concurrent engineering requires downstream engineers to make use of preliminary data from upstream engineers. Using such information entails risk -- there is a high probability that the information will change. In addition, in a study of effective concurrent engineering practices, researchers found that "the use of incomplete and uncertain information -- a task structure requirement in CE -- had a negative effect on [a] team's satisfaction [with the development process]" (Hauptman and Hirji, p. 161). Set-based methods, however, "[allow] team members to communicate information they actually possess in the early stages of the design process, rather than sharply defined decisions that they cannot be confident will survive" (Liker *et al.*, p. 166). Instead of communicating about a single, specific design that will likely change, set-based practices allow engineers to discuss designs at a level that more precisely indicates the certainty of design data. This more precise communication then facilitates more effective use of preliminary design information, improving engineers' ability to implement concurrent and simultaneous design.

Another set-based principle which helps to facilitate early use of preliminary information is that of conceptual robustness. This concept is based on Taguchi's notion of functional robustness. Taguchi states that a product is functionally robust "if it inherently tends to diminish the effect of input variation on performance" (p. 9). Similarly, design decisions are conceptually robust if the decisions remain valid regardless of the choices made by other engineers working on the product (Sobek *et al.*, p. 21; Chang *et al.*, p. 212). As engineers narrow their sets and communicate about their remaining alternatives, they should also seek to make future choices that will accommodate as many of the other engineers' options as possible. When engineers are able to follow this principle, they significantly reduce the likelihood of having to redo their work because of decisions made by other designers.

A simple example helps to illustrate this point[1]. In the typical concurrent engineering environment, structural design engineers and manufacturing die designers will be working simultaneously to develop the structural design of a product and the dies used to stamp out the parts. In a traditional approach, the structural designers might show the die designers preliminary drawings of their parts. The die designers would then use this information to begin designing dies. If the structural designers alter the part design, however, the die designers will also have to alter their work.

A set-based approach, however, eliminates this potential for scrap and re-work. Early in the design process, structural engineers would show die designers the likely range, i.e., set, for their parts. This range would illustrate the maximum size likely for the parts as well as the minimum size. Based on this range, die designers could begin to design the dies, so long as the designs accommodated all of the possible shapes which the structural designers were considering -- that is, so long as the die designs were conceptually robust to the decisions of the structural designers. As the structural designers refined their set and narrowed the possible span of dimensions, the die designers could do the same, increasing the detail in their die designs. Ultimately, each group would narrow down to a single design, and would have done so (ideally) without ever having to backtrack through the concurrent development process.

This illustration also demonstrates another important principle for the effective implementation of set-based concurrent engineering: stay within a set once committed to it (Sobek *et al.*, p. 24). Once engineers have narrowed a set, it is important that one engineer not "jump out" of the set to introduce a new design concept, unless absolutely necessary. This commitment is essential "so that other [engineers] can rely on [each other's] communication" (Sobek *et al.*, p. 24). By remaining within a set, "as the set narrows, the earlier communications remain valid but are supplemented with further, more precise information" (Ward *et al.*, p. 58). For example, if a set of designs, *B*, is a subset of *A*, then a property that is universally true for all of the designs in set *A* is also true for all of the designs in set *B* (Sobek, 1997, p. 232). Since downstream sets are

[1] This example is based on material in Ward *et al.*

simply subsets of upstream ones, "any work or communication based on upstream sets is also valid for all downstream sets, including the final solution" (Sobek, 1997, p. 232).

In many ways, therefore, these concepts illustrate a key notion of SBCE: In a set-based approach, "designs converge rather than evolve" (Liker *et al.*, p. 168). Each functional group narrows "their respective sets in parallel, communicating throughout to ensure that each function converges to a solution that integrates with the overall system" (Sobek, 1997, p. 218). In this manner, set-based methods allow for "truly parallel" design, enhancing the benefits of concurrent engineering practices (Liker *et al.*, p. 168).

### 3.4.4 Using Sets to Integrate and Optimize a Design

In general, any design problem requires that engineers satisfy multiple and often conflicting goals (Otto and Antonsson; Krishnan *et al.*, 1997.). In addition, different functional organizations might prefer opposing values for a given performance characteristic (Krishnan *et al.*, 1997). Developing an optimal design can also be further complicated because performance features might not always have mathematical descriptions which lend themselves to optimization (Krishnan *et al.*, 1997). In addition, an optimal system design is not necessarily one which optimizes specific features individually, but one which maximizes the overall performance of a design (Otto and Antonsson). Authors such as Pugh recommend a variety of techniques to optimize product designs. These authors, however, do "not explore the power of different actors independently defining sets of possibilities from their own perspectives, then looking for intersections, thereby achieving parallel development" (Sobek *et al.*, p. 8).

In set-based design, optimization is intimately tied to integration. Following typical design methods, set-based design first decomposes a complex design problem into numerous smaller ones, each of which is then assigned to individual engineers or teams. Each group then explores their design space using sets. This exploration allows each group to define their feasible regions, their sets of possible design solutions. The integration process, then, consists of the groups looking for regions of overlap in these sets (Sobek, 1997, p. 216).

Attempting to integrate the solutions then leads to the optimization of the system. If an intersection exists, for example, its existence implies that the solution is acceptable to all of the groups (Sobek *et al.*, p. 17). Further, the solution contained in the intersection is likely to be a *globally* optimal solution -- "the one that best achieves integration, the one that best meets the objectives of *all* of the stakeholders in the process" (Sobek, 1997, p. 222). In fact, if no intersection is found, it is likely the result of engineers attempting to "marry independently optimized concepts" (Sobek *et al.*, p. 17). In such cases, engineers must broaden their search, i.e., expand their sets, to increase the potential for finding an intersection.

### 3.4.5 Requirements in the Set-Based Context

The communication, integration, and optimization processes described above all have a common implication for the manner in which requirements should be specified. Like a new product's design, a new product's requirements should gradually evolve and then converge during the course of the development process. By using evolving requirements, rather than attempting to quickly finalize requirements that will only change later, set-based methods are able to avoid much of the requirements paradox common to iterative methods.

Set-based requirements can be stated in two basic forms: ranges or minimum/maximum constraints[2]. When requirements are given as ranges, the parameter is specified as being allowed to be "no less than *x* and no greater than *y*." Using ranges on requirements "provides freedom and focuses effort on exploring regions of the design space so that they can be narrowed rationally," to help facilitate the integration and optimization process (Ward *et al.*, p. 49). The second mode for stating requirements is to use minimum or maximum constraints -- specifying that the parameter

---

[2] The SBCE literature tends to refer simply to "minimum constraints," without formally noting the opposite but conceptually similar concept of maximum constraints.

“can be no less (or can be no greater) than *x*.” Like ranges, minimum or maximum constraints give other engineers “the opportunity to optimize the design from their perspective, and if that makes no difference [to another engineer’s] subsystem, then the minimum [or maximum] constraint helps to optimize the system as a whole” (Sobek, 1997, p. 224).

In both instances, the goal of such requirements is to incorporate some flexibility in the design process, so that a product’s performance goals can be adjusted as engineers become better informed about what is possible. Liker *et al.*, for example, found that companies using set-based requirements practices tended not to use these methods based on any formal intent to do so. Instead, they were simply unsure of exactly what the requirements should be early in the design process (Liker *et al.*, p. 177). By allowing the requirements to evolve, these companies avoided placing unnecessary and inappropriate constraints on their designs, while also clearly stating the goals for the final product.

### 3.4.6 Managing Set-Based Concurrent Engineering

Supervising these narrowing processes for both designs and requirements mandates a slightly different management approach than is normally used in engineering design. SBCE “views the [design] process as more of a flow than as an arrangement of tasks” (Sobek, 1997, p. 233). Consequently, management practices must be geared to regulating this flow. Specifically, the major issue for managing set-based design is to ensure that the narrowing process does in fact occur.

An efficient means of controlling this development is through the use of process gates. As oppossed to design reviews or checkpoints, gates are based on integration activities such as the construction of prototypes (Sobek, 1997, p. 233). As Wheelwright and Clark state, prototypes provide “feedback about the choices made thus far and highlight remaining unresolved issues” (p. 259). In addition, prototypes serve as bridges “between individuals and groups with very different backgrounds, experiences, and interests” (Wheelwright and Clark, p. 274). A prototype, therefore, provides an effective mechanism for integrating the various solutions being pursued by different functional groups.

In addition to prototypes, sets themselves are actually a useful tool for managing the design process. The size of a set is an indication of the feasibility of the remaining design alternatives (Sobek, 1997). If a set is still very large, it indicates that engineers are still uncertain about the design. When a set becomes small, on the other hand, it demonstrates that engineers are confident that they are nearly at the final solution. Thus, process gates serve as formal points in the development process at which managers and engineers can review how large various sets are. Sets that are larger than others provide managers and engineers with a clear signal as to which aspects of the design are progressing more slowly than others, and, therefore, which aspects require more attention. Hence, in addition to aiding engineers communicate with one another, sets help engineers and managers control the design process.

## *3.5 Other Methods Which Recommend Carrying Options*

### 3.5.1 Two Examples

Many authors on design and engineering emphasize the importance of considering multiple design alternatives, especially early in the design process. Pahl, for example, states that “the solution field should be as wide as possible” (p. 112). Ulrich and Eppinger cite several common failures in concept development which stem from not evaluating enough design options, including:

- only considering one or two alternatives;
- failing to consider concepts used by other firms “in related or unrelated products;”
- failing to successfully integrate several promising “partial solutions;” and
- failing to consider “entire categories of solutions” (p. 79).

In addition they note that “[t]horough exploration of alternatives early in the development process greatly reduces the likelihood that the team will stumble upon a superior concept late in the process or that a competitor will introduce a product with dramatically better performance than the product under development" (p. 78). Two specific approaches help to illustrate how designers might consider multiple design alternatives.

Pugh recommends a method of "controlled convergence." In this approach to design, engineers and designers initially develop a very large number of design concepts. These concepts are then compared to the customer's needs and to one another. Concepts which rate highly are retained, while others are discarded. Once the initial number of designs has been reduced, the design team again considers additional alternatives -- either modifications of some of the initial concepts or entirely new concepts. This set of designs is narrowed further, and then, again, new options are added. The process continues in this fashion, with the generation of ideas followed by convergence. Each successive iteration of the process results in a narrower and narrower field of alternatives, until only one option remains. This repetitive expansion and contraction approach is illustrated in Figure 14.

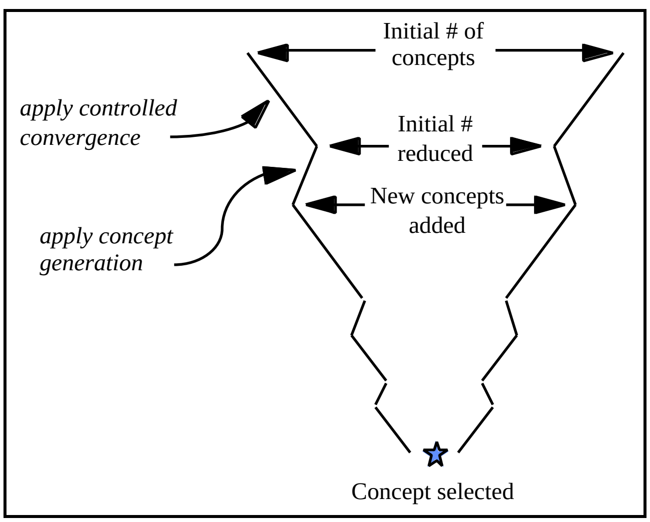



**Figure 14: Method of controlled convergence.** A number of design concepts are generated and then reduced, so that numerous design options are considered, until only one design remains. (Adapted from Pugh, p. 75)

Wheelwright and Clark propose a similar approach to design based upon the design-build-test cycle. Illustrated in Figure 15, the first step in the process is to frame the problem: establish product and manufacturing process requirements, clarify user needs, etc. Once the requirements are clearly understood, several alternative designs are developed, the purpose of which "may be to explore the relationship between design parameters and specific customer attributes" (Wheelwright and Clark, p. 223). Models, either physical or computer-based, are then constructed and "subjected to tests that simulate product use" (Wheelwright and Clark, p. 6). If the models fall short of performance requirements, "engineers search for design changes that will close the gap" between the models' performance and the required performance (Wheelwright and Clark, p. 6). The design-build-test cycle is then repeated, until all of the requirements are fulfilled.

This approach to design, therefore, is built around the concept of repetition. A single design-build-test cycle is used to provide information to the next iteration of the cycle. The effectiveness of this method, then, depends upon the effectiveness of a single cycle, the number of cycles that are completed, and how well the results of individual cycles are combined into coherent solutions (Wheelwright and Clark, p. 226).

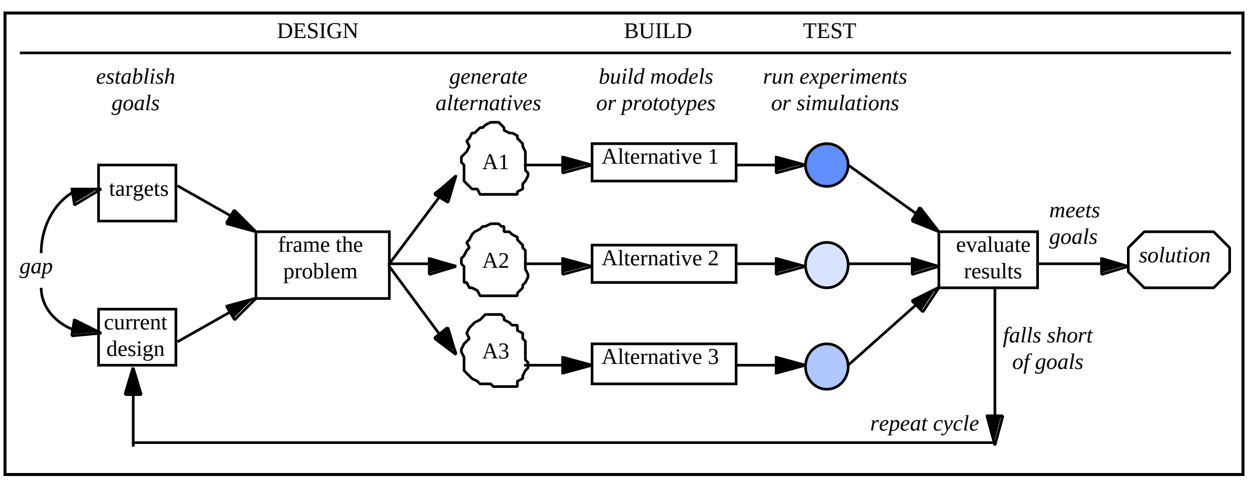



**Figure 15: The design-build-test cycle.** Product and process goals are first established, leading to the development of several design alternatives. The alternatives are modeled and tested, and the results of the tests are used to determine if a design meets the goals or if further development is required.

### 3.5.2 Contrasting These Methods with SBCE: Different but Highly Complimentary

Both of these approaches emphasize the importance of considering multiple alternatives. They differ, however, from SBCE in one key respect: how the options are used. Controlled convergence and design-build-test both suggest that rather than considering multiple alternatives and then quickly selecting only one, several should be carried for a protracted period of time. This concept is analogous to one of the two primary principles of SBCE -- considering multiple design alternatives. Where these methods differ from SBCE, however, is the manner in which the design options are used. In both Pugh's controlled convergence method and Wheelwright's and Clark's design-build-test cycle, alternatives are generated to understand how different design parameters impact a concept's ability to satisfy a user's requirements. SBCE uses design options in this manner, but *set-based methods also use options to allow each specialty group working on a product to explore the design space independently*. By allowing specialty groups to independently analyze their design options, set-based methods eliminate the iterative paths which can be so problematic in point-based approaches. Controlled convergence and design-build-test do not necessarily emphasize this use of design options.

It is important to recognize, however, that set-based methods and approaches such as controlled convergence and design-build-test are not mutually exclusive. In fact, they are highly complimentary. For example, set-based methods rely on periodic tests to help narrow sets of design options, and these testing cycles are essentially the same as those described by Wheelwright and Clark. The principal difference is that whereas Wheelwright and Clark use tests to determine whether or not to consider new design options, SBCE uses tests to eliminate previously generated options and to determine when to increase the level of detail in the remaining design alternatives.

Similarly, Pugh's technique and SBCE can also be used together. Controlled convergence accepts a reality of design: Rarely (if ever!) is it possible to identify every design option at the start of a development project. Instead, an initial set of designs is generated, and, based on the analysis of these designs, engineers develop new concepts. Pugh's method provides a means for allowing the inclusion of such additional ideas while preventing engineers from simply developing new options without ever driving toward a final solution.

Ideally, a set-based approach would begin by considering every possible design option and would then eliminate designs, so that number of solutions was always reduced and never increased. If such a task were possible, however, the design process would in fact be quite dull: The ability to generate every possible option at the beginning of a project essentially implies the ability to foresee the future. Rather than clinging to the false belief that engineers could possess such vision and then attempting to prevent increases in the set of design options at all costs, the blending of controlled convergence with SBCE allows for the flexibility of considering additional options while at the same time maintaining the advantages offered by pure set-based practices, such as removing iterative paths between specialty groups. Though this practice violates the principle of remaining within a set, if used cautiously, controlled convergence could help engineers avoid narrowing their designs to a null set, that is, a final solution which in fact will not work.

Rather than suggesting that SBCE replace other approaches to design, therefore, it is important to identify the similarities between different methods and then to capitalize on the advantages offered by each. By combining elements of several design techniques, engineers will be able to develop robust design capabilities which will be highly adaptable to a wide variety of product development problems.

### *3.6 SBCE versus Platform Design*

At first glance, another strategy with similarities to SBCE is platform design. There are, however, important distinctions between SBCE and platform design. To understand these differences, one must first understand what is meant by platform design.

Platform design refers to a product development strategy in which a company develops a family of products which can share components and assests to target specific market segments (Meyer and Utterbeck). A product platform, therefore, consists of the design and components which are “shared by a set of products” (Meyer and Utterbeck, p. 30). By combining components in different arrangements, a platform can give rise to a product family, in which each member is tailored to the needs of a specific customer group.

Platform design, therefore, refers to the *product strategy* of developing product families. In contrast, SBCE refers specifically to the *design strategy* used to implement a product strategy. Thus, a platform product could be developed using SBCE, but it could also be developed using a point-based, iterative strategy. Furthermore, SBCE can be used to develop a specific product, one that is not part of a family.

Note, however, that SBCE is highly conducive to a platform design: As a set of product components are developed, individual elements of that set could give rise to the product variations used to implement a product family. On the other hand, when pursuing a specific product design (one that will not lead to a family), individual elements of a set are discarded when they are shown to be inferior. The purpose of considering those alternatives is to ensure that the design space was thoroughly explored so that the best design is chosen. In the context of a platform product strategy, the alternatives might be used instead to develop the interchangeable elements of a platform.

In summary, therefore, *platform design* refers to the *product strategy*, while *SBCE* refers to the *design strategy* used to develop a product.

### 3.7 Summarizing Set-Based Concurrent Engineering

To summarize, SBCE consists of two principle concepts:

1. Consider a large number of design alternatives, i.e., *sets of designs*; and
2. Allow specialists to consider a design from their own perspective, using the overlap (intersection) between individual sets to optimize a design.

Table 1 expands these fundamental concepts of SBCE, and Table 2 illustrates how it differs from point-based, iterative methods. The overall effects of this method are depicted in Figure 16, a reproduction of Figure 12: Set-based methods are used to delay decisions so that engineers can delay designing costs into a system and increase their leverage late in the development process.

**Table 1: Principles of SBCE** (Adapted from Sobek *et al.*)

| |
|---|
| ***Understand the design space*** |
| Define feasible regions |
| Explore tradeoffs by designing multiple alternatives |
| Communicate sets of possibilities |
| ***Integrate by intersection*** |
| Look for intersections of feasible sets |
| Impose minimum (maximum) constraint |
| Seek conceptual robustness |
| ***Establish feasibility before commitment*** |
| Narrow sets gradually while increasing detail |
| Stay within set once committed |
| Control by managing uncertainty at process gates |

**Table 2: Comparison of Point-Based and Set-Based Approaches** (Adapted from Sobek, 1997)

| Function | Point-Based Approach | Set-Based Approach |
|---|---|---|
| *Search:* How should solutions be found? | Iterate on existing ideas.<br>Brainstorm new ideas. | Define feasible regions. |
| *Communication:* Which ideas are communicated to others? | Communicate the best idea. | Communicate sets of possibilities. |
| *Integration:* How should the system be integrated? | Pass the idea among the team for critique. | Look for intersections. |
| *Selection:* How is the best idea identified? | Formal schemes for selecting the best alternative.<br>Make prototypes to confirm that the solution works. | Design in parallel on each alternative until it is not worth pursuing.<br>Look for low cost tests to prove infeasibility. |
| *Optimization:* How should the design be optimized? | Analyze and test the design.<br>Modify the design as necessary to achieve objectives and improve performance. | Design in parallel on each alternative until it is not worth pursuing.<br>Look for low cost tests to prove infeasibility. |
| *Specification:* How should you constrain others with respect to your own subsystem design? | Maximize constraints in specifications to assure functionality and interface fit. | Use minimum control specifications to allow optimization and mutual adjustment. |
| *Decision risk control:* How should one minimize the risk of "going down the wrong path?" | Establish feedback channels.<br>Communicate often.<br>Respond quickly to changes. | Establish feasibility before commitment.<br>Pursue high-risk and conservative options in parallel.<br>Seek solutions robust to physical, market, and design variation. |
| *Rework risk control:* How should one minimize damage from unreliable communications? | Establish feedback channels.<br>Communicate often.<br>Respond quickly to changes. | Stay within sets once committed. |
| *Management:* How should the process be controlled? | Review designs and manage information at transition points. | Manage uncertainty at process gates. |

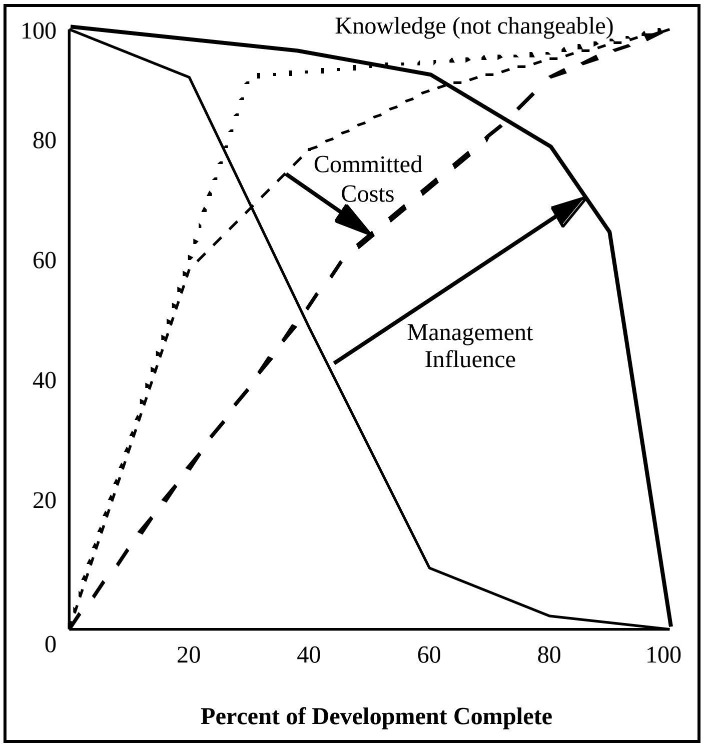



**Figure 16: The effects of SBCE.** In essence, SBCE provides a mechanism to allow managers and engineers to delay decisions while at the same time continuing to develop a product. The effects of SBCE, therefore, are to delay the commitment of costs and to increase management influence late in the development process.

# 4. Example of SBCE from Toyota and Its Suppliers

## *4.1 Introduction: "The Second Toyota Paradox"*

Toyota's design and development practices clearly form a tightly integrated process, and it is not the intent of this thesis to review it in detail[3]. Instead, the following paragraphs will illustrate some set-based practices which Ward, Sobek, and their associates have detailed. These examples are useful both to clarify the theory presented in the previous chapter and as reference points for comparison with aerospace practices (described in the coming chapters).

In their study of Japanese and American car manufacturers' development practices, Ward *et al.* had hoped to be able contrast design methods between the two countries. What they discovered, however, was that practices across the Pacific were becoming quite similar, except for one company, which stood out from the others: Toyota (Ward, *et al.*, p. 45). Based on their research, they coined the phrase "The Second Toyota Paradox" to describe Toyota's development process[4]. Sobek *et al.* summarize the paradox as "consider[ing] a broader range of possible designs and delay[ing] certain design decisions longer than other companies, yet [having] what may be the fastest and most efficient vehicle development cycles in the industry" (p. 3).

The following sections of this chapter will provide examples of set-based practices from Toyota. The next section will describe set-based practices within Toyota, while the subsequent section will illustrate set-based methods between Toyota and its suppliers.

[3] For an excellent description of the Toyota development process see Sobek, 1997, from which much of the material in these sections is drawn.
[4] The "first" Toyota paradox refers to the Toyota Production System (TPS).

## 4.2 Set-Based Practices within Toyota

### 4.2.1 Design Organization

At the corporate level, Toyota's design engineers are organized into platform divisions. Within each division, engineers on a specific project are arranged in a "chimney" structure based on a matrix organization (Ward *et al.*, p. 46). For each program, as was noted above, a team's structure follows functional lines, such as styling, body engineering, chassis, powertrain, prototype, and manufacturing (Sobek, 1997). In fact, the manufacturing organization is actually located in a separate building nearly twenty miles from the rest of the development engineers (Sobek, 1997). Engineers are usually assigned to multiple programs, working on the same aspect of a design for each project in which they are participating (Sobek, 1997, p. 139).

An important feature of Toyota's design organization is the chief engineer. Similar to a heavyweight program manager, the chief engineer is responsible for the entire development of a new car, from the very first notions of what type of car should be developed all the way through manufacturing and sales (Sobek, 1997; Ward *et al.*). The chief engineer is responsible for defining the architecture of a new vehicle, developing the vehicle's specifications, performance and cost targets, and schedule and product plan. Furthermore, when conflicts arise between engineering groups, the chief engineer has the final word in any design compromises (Sobek, 1997, p. 167).

To balance this considerable authority, the chief engineer actually has no formal control over any of the engineers which work for him (Sobek, 1997, p. 145). Instead, engineers assigned to a particular program report back to their own functional organizations. This apparent organizational paradox functions well because of the mutual dependence of the functional organizations and the chief engineer (Sobek, 1997). The chief engineer depends upon the functional groups to provide engineers to actually do the work required to develop a new car. Since the chief engineer is responsible for overseeing the development of new vehicles, however, the functional organizations depend upon chief engineers to provide them with work.

Furthermore, both groups, the functional engineers and the chief engineer, are evaluated within their own organizations based upon the success of a development program. These mutual dependencies, therefore, force cooperation between the two groups, allowing the process to work effectively.

### 4.2.2 A Quick Overview of the Process Highlighting the Use of Sets

Based on material presented in Ward *et al.*, Sobek (1997), and Sobek *et al.*, Figure 17 presents a simplified overview of the Toyota development process. The following paragraphs review their approach in greater detail, highlighting how Toyota employs set-based methods in their development process.

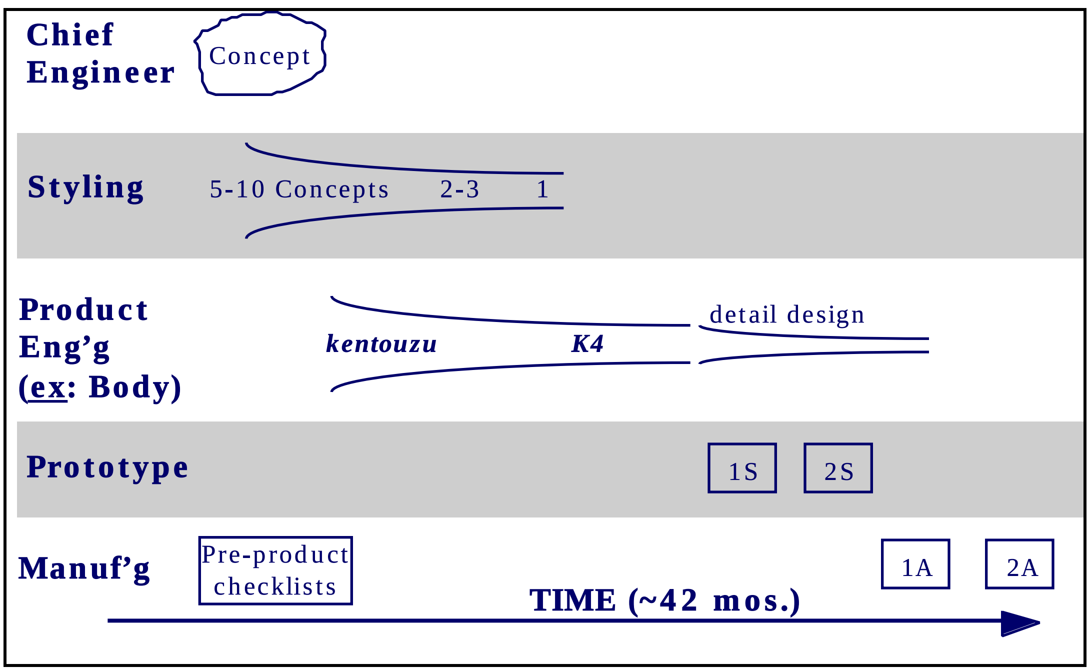



**Figure 17: A simplified overview of the Toyota development process.** As shown, the design process begins with the chief engineer's concept. Styling then develops a range of alternative designs, which are analyzed by the engineering departments. Two prototype cycles are used to finalize design decisions, leading to two trial production runs. Note that in this simplified illustration, only body engineering is listed under product engineering. Other product engineering groups include chassis and powertrain. (Based on material in Sobek, 1997; Sobek *et al.*; and Ward *et al.*)

Toyota's development process begins with the chief engineer and his support staff. Together, they conceive of an overall vision of the new vehicle that is to be developed, referred to as the chief engineer's concept. This concept will suggest the basic style of vehicle, what type of engine and suspension it will use, and what its performance and cost targets will be. Next, stylists begin to develop a range of potential design options, essentially creating a set of alternatives. Typically, they develop five to ten concepts which are then shown to the product engineering groups. Toyota's general manager of styling commented that they pursue so many designs because they "prefer lots of torpedoes to a single sniper bullet" (Ward *et al*., p. 47).

Simultaneously, manufacturing engineers will review current and future manufacturing capabilities, and then summarize these capabilities on pre-product checklists. These checklists, which are given to the engineers and stylists developing a new design, detail only those process capabilities which will be relevant to a particular program. As one production manager explained, by providing the checklists, manufacturing hands over their design experience to the stylists and engineers (Sobek, 1997, p. 180).

Using body engineering as an example, the engineering groups review the pre-product checklists and the initial concepts provided by styling. They then develop initial engineering drawings, which in the case of body engineering, are referred to as *kentouzu*, or "study drawings" (Sobek, 1997, p. 91). These drawings present preliminary concepts for the structural design of each style under consideration by the stylists. Importantly, these drawings are then shared with the stylists, informing them as to what problems the body engineers have with a particular style and what improvements can be made.

Using the information provided by the *kentouzu*, the stylists then narrow the number of designs which they consider to two to three alternatives. These designs are then made into full-scale clay models for further review and critique. In parallel, body engineering develops the *kozokeikaku*, referred to as the K4 by American engineers, which are more detailed structural designs (Sobek, 1997, p. 92). These drawings refine the concepts initially developed in the *kentouzu*. In addition, body engineering conducts crash analyses, and uses checklists developed from past lessons learned to further understand the performance of each design. This data, along with similar information from other engineering groups, and the input of manufacturing engineers and the chief engineer, is then used to select a final style.

Detail design then begins. Stylists become less involved in the process, though the chief stylist will continue to observe the engineering process, ensuring that modifications made by the engineering groups do not spoil the intended style of the vehicle. In general, decisions during detail design are delayed for as long as possible. For example, the final locations of body hardpoints, which can significantly affect the shape of the vehicle, are made as late in the process as possible, even though such delays decrease the time available for die designers to finalize die shapes (Ward *et al.*, p. 48). The general manager of body engineering explained that such late decision-making is required to ensure that the customer's expectations are fully understood and that the design will be manufacturable (Ward *et al*., p. 52).

As detail design progresses, two prototype cycles are used to refine the design. The first set of prototypes, the 1S prototypes, are used to select between different subsystems and to test variations in specific design features. Based on the results of this testing, specifications are modified, and the 2S prototypes are constructed. As long as the test results of the 2S prototypes confirm the decisions made after the 1S vehicles, the final set of specifications will be released (Ward *et al.*, p. 23). Importantly, however, it is after this second round of prototypes that requirements are finalized, very late in the design process indeed.

Changes made during the prototype cycles are incorporated into the detailed engineering drawings, which are then released to manufacturing for the first pre-production build, referred to as the 1A production run. An interesting feature of this process is demonstrated in the construction of body panel stamping dies[5]. Drawings for the body panels will be released to manufacturing without tolerances. Manufacturing engineers then build the dies as closely as possible to the

[5] The following account is supplemented by material in Liker *et al.* in addition to information provided in Sobek (1997) and Ward *et al.*

nominal dimensions indicated by the body designers. The dies are then tested, and, if they yield satisfactory results, the final part dimensions are then sent back to body engineering, which updates their drawings to reflect what is actually built. A second trial production run, 2A, is then conducted to verify the final designs, followed by the start of full-scale production.

### 4.2.3 Discussion of the Process: The Relationship between Upstream and Downstream Groups and the Use of Standardized Processes

Two feature of the process described above are worthy of additional comment. The first is the relationship between upstream and downstream processes, and the second is the extensive use of checklists and standardized processes.

As suggested by SBCE theory, the different groups involved in the design process all clearly consider a set of design options. In addition, they independently analyze these options from their own perspectives. The Toyota development process, therefore, seems to satisfy both elements of the definition of SBCE. But what is particularly interesting is how the different groups coordinate their exploration of this set. For example, once the stylists present a set of design alternatives, body engineers use the *kentouzu* to provide information back to the stylists as to their preferred options. The pre-product checklists provided by manufacturing provide a similar function. These checklists inform the other design groups about the capabilities of the manufacturing department, and, consequently, what designs will be easiest to manufacture, i.e., which designs would be preferred by the manufacturing engineers. These communication tools, therefore, provide mechanisms for downstream groups to inform the upstream groups about their design preferences.

Another feature of the Toyota development cycle which stands out is its use of standardized design practices. A common manifestation of these standards is the use of design checklists. Functional groups exchange these checklists to share their knowledge and experience and also to help ensure that a design will be acceptable to other groups: "If the design conforms to the checklist, the part is guaranteed to meet a certain level of functionality, manufacturability, etc." (Sobek *et al*., p. 13).

While checklists can help to ensure success in the design effort, they could also stifle engineers' creativity, and, consequently, their ability to develop products which will satisfy the customer. To avoid such problems, Toyota uses an approach which Sobek (1997) refers to as "flexible standardization" (p. 194). For example, a design standard will not require that a specific flange radius be used for a certain part, but will instead recommend that the radius fall within a certain range depending upon the grade of steel used in the design. In addition, efforts are made to continuously improve design standards, so that the realm of possibilities is always increasing (Sobek, 1997, p. 194). Finally, Toyota does allow deviations from the standards, so long as an engineer can demonstrate that such a deviation will result in a significant improvement in a design (Sobek, 1997, p. 194).

Standardized practices are also important at Toyota for several additional reasons. First, they provide a basis for measuring improvements (Sobek, 1997, p. 192). If the design process itself varied every time, engineers would have no way of knowing whether or not problems and successes were the result of chance or something specific which they did. Once a process is standardized, on the other hand, changes which result in improved designs can readily be discerned when compared to prior techniques. Similarly, when problems do occur, engineers can trace their steps through the standard process to determine where a problem originated and then change the process so that the problem is avoided in the future (Sobek, 1997, p. 192). Finally, these gradual changes and improvements to standardized processes provide Toyota with a simple technique for capturing knowledge as engineers leave the company (Sobek, 1997, p. 193). By embedding the experience of past engineers in a standardized process, the engineering knowledge possessed by an individual is transferred to the company.

### *4.3 Set-Based Practices between Toyota and Its Suppliers*

In addition to practicing many set-based techniques within its own processes, many of the methods used by Toyota to interact with its suppliers also illustrate set-based techniques. Figure 18 presents on overview of the supplier development timeline, and the following paragraphs provide details of the process and some examples.

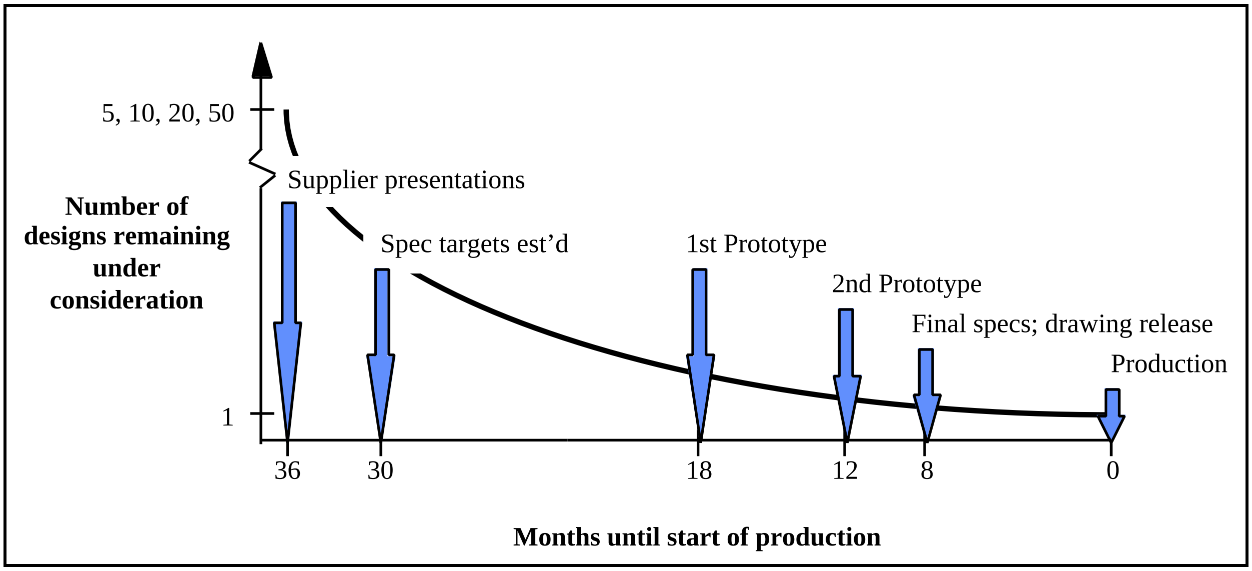



**Figure 18: The supplier timeline.** Suppliers first become involved in the design process when they present their latest developments to Toyota. Once selected for a specific program, target specifications are first established, which are used to guide the development of a subsystem. Two prototype cycles are used to validate the design, leading to the final specifications that drawing release, followed shortly thereafter by production. (Based on material in Carbone; Sobek, 1997; Sobek *et al.*; and Ward *et. al.*)

To begin to understand Toyota's relationships with its first tier (i.e., closest) suppliers, one must recognize that Toyota does not use a bid system, as is common practice at other companies. Instead, "when Toyota conceives an idea for a new car model it uses the design expertise of its suppliers to develop the specifications of parts and systems" (Carbone, p. 15). A supplier's involvement in a new product development effort begins when the supplier presents its latest technologies to Toyota engineers. Toyota will then use these presentations to establish preliminary target specifications for a new part or subsystem (Ward *et al.*, p. 57). The supplier is then expected to return several months later "with a number of alternative designs, analysis and test data, and, when possible, prototypes" (Sobek, 1997, p. 212). These practices "allow the [chief engineer] to understand the trade-offs and set [refined] targets to produce the best possible design" (Ward *et al*., p. 57). As the supplier refines its design, the target specifications are narrowed, until finally very precise requirements are stipulated for the supplier's product (Ward *et al.*). As shown in Figure 18, however, these final specifications are not released until very late in the development process, after Toyota and the supplier have been able to test the product extensively.

Interestingly, this gradual development of requirements seems to be complemented by the scheduling techniques used between Toyota and its major suppliers. Rather than establishing numerous, specific deadlines, Toyota only specifies major program milestones and then allows the suppliers to decide schedule details, such as when to initiate efforts to obtain information or to refine a design (Ward *et al*.). These practices are similar to those which Toyota uses in-house, to establish its own program schedules and timelines (Sobek, 1997).

Two examples help to illustrate these practices. Consider first the development of an exterior plastic mold[6]. Toyota designers first draw a sketch of the part which is then shown to several suppliers. Each supplier then develops five to six alternative concepts, which they share with the Toyota designers. Based on these sets of designs, Toyota selects a single supplier. Then, working together, the Toyota designers and engineers from the supplier refine the details of the design, "reducing the set until arriving at the final specifications" (Sobek *et al.*, p. 15).

A similar process is followed in the case of the exhaust system. In this instance, however, a single supplier will typically produce ten to twenty prototypes, and has constructed as many as fifty (Sobek *et al*., p. 14; Liker *et al*., p. 169). The motivation behind the development of so many prototypes was to allow "for knowledgeable tradeoffs to define the optimum subsystem" (Sobek *et al.*, p. 14).

These examples help to demonstrate how Toyota relates to its suppliers, but several additional points should be noted. First, Toyota does *not* use these techniques with all of its suppliers. Suppliers which are regarded highly are treated in the manner described above, while others receive very detailed specifications which must be fulfilled exactly (Liker *et al.*, p. 169). In addition, the communication skills required to use these techniques take time to evolve. As several supplier representatives explained, they "understand from the stage of the process, the body language of [Toyota's] engineers, and the atmosphere of [a] meeting how much flexibility remains around [a specification], i.e., an implicit range of requirements" (Liker *et al*., p. 176).

Finally, when a prime makes use of such set-based techniques with its suppliers, the prime "must trust that the supplier will not take advantage of the tolerance and send back the cheapest design that falls within the acceptable range," while the supplier "must understand that the customer has some implied tolerance and take responsibility for understanding the [prime's] intent and creatively explore designs to meet it" (Liker *et al.*, p. 169). Clearly Toyota's methods for supplier relations are innovative and can potentially improve such alliances. At the same time, however, managers at primes and at suppliers must trust one another to not abuse the system, so that both groups' needs are satisfied.

---

[6] This example is based on material presented in Sobek *et al.*

# 5. Assessing the "Set-Basedness" of the Aerospace Industry: Setting the Stage

## *5.1 Chapter Introduction*

The first goal of this research was to provide a clear definition of the principles of set-based concurrent engineering so that it could be compared and contrasted with other approaches, a goal accomplished in the preceding chapters. The second goal of this research was to assess whether or not current aerospace industry design methods demonstrated any SBCE-like practices. To that end, the following chapters will present several "mini-case studies" describing design processes at a variety of aerospace companies. As will be seen, no company used the term "set-based" to describe their techniques, though methods at some firms had elements of SBCE in them. Other companies, however, used approaches which much more closely resembled point-based methods of design.

The next section will provide a brief overview of the research design used to conduct the industry study. Following sections and chapters will then present the mini-case studies, along with discussion and comment relative to SBCE principles. Chapter 10 then concludes with an overall assessment of industry practices in the context of set-based concurrent engineering.

## *5.2 Research Design*

Design methods present a challenge for research. By its nature, "design" is typically an inexact science (Suh). It does not, therefore, readily lend itself to quantitative study, especially in the specific case of design methods -- *how* engineers do design. Furthermore, the intent of this research was not necessarily to propose that one design method is better than another -- a common use of quantitative measures. Instead, this research aimed to understand the current state of aerospace design practices and the *reasons* behind those practices. Based on that understanding, SBCE can be placed in the proper context within the aerospace engineering environment, and presented to engineers as another tool which they might choose for a given problem.

To gain this needed understanding of design methods, this investigation approached the problem through a series of mini-case studies. Site visits were conducted at several Lean Aerospace Initiative consortium members' facilities, where interviews were held with engineers working on a variety of programs at several levels, from program managers to detail design engineers. For reasons of confidentiality, no company or program names can be used in this thesis, though Table 3 provides a sense of the scope of this research effort.

**Table 3: Measuring the scope of this investigation.**

| | |
|---|---|
| Number of Sites Visited: | 9 |
| Sectors Represented: | Combat Aircraft, Missiles and Electronics, Space Systems |
| Total Number of Interviews Conducted: | 88 |
| Number of Interviewees with Titles of Manager, Director, Leader, or Chief Engineer: | 65 (74% of total) |
| Number of Interviewees with Title of Engineer: | 23 (26% of total) |

Interviews were typically conducted with individuals, though some interviews took place in small groups usually no larger than three. Prior to going to a site, a brief overview of the research (which essentially contained the material presented in the introduction to the thesis) was forwarded to the company, although not every individual who was interviewed had necessarily seen the summary. Each interview began in the same fashion, with a brief introduction to the research, its goals, an overview of SBCE theory, and a hypothetical example of how SBCE could be applied to an aerospace problem. This presentation is included in Appendix A.

Depending upon the person being interviewed and his or her prior degree of preparation, sessions usually followed one of two paths. If the person had had the opportunity to prepare material prior to the meeting, he or she would present that information, often tailoring it "on the fly" to more specifically address issues raised during the SBCE presentation. A discussion would then ensue, guided by reference to the questions listed in Table 4. If the person had not had the opportunity to prepare in advance, the session moved immediately to the interview questions. Answers varied, of course, depending upon the program on which the person worked, his or her role on that program, and the individual's own personal perceptions. Typically, however, responses to questions were fairly consistent across a company, even when engineers were working on different programs.

**Table 4: Interview questions.**

***Defining Requirements.***

How are requirements usually specified -- as single values, as ranges, or as combinations of both?

When are requirements set in the design process -- early, late, etc.?

Are tools such as quality function deployment used, and, if so, how?

***Number of alternative concepts considered.***

How many alternatives are considered during the design process?

How do these alternatives differ, i.e., are completely different systems being compared or are comparisons made between "variations on a theme"?

Are different alternatives considered at each level of design, i.e., complete systems, subsystems, small parts?

How long are the different alternatives considered, and are multiple concepts developed in parallel?

What means are used to eliminate designs from consideration?

***Iteration**.**

What are the major causes of iteration in design? Iteration defined in terms of trades between engineering specialties.

Are these tradeoffs considered in series or in parallel?

Do design schedules include plans for iteration?

***Use of prototypes**.**

Are prototypes regularly used in the development process?

If so, how many prototypes are developed?

Are these prototypes shown to the customer?

***Team organization and communication**.**

How are design teams organized?

What mechanisms are used to facilitate inter-team communication?

***Interface between design and manufacturing**.**

How is a part design communicated to manufacturing?

What constraints, tolerances or specifications are given on drawings prior to manufacture, and how are these requirements derived?

***Are there other features of the design process which these questions fail to capture?***

***What do you think works well about the way in which you do design?***

***What aspects of your design method do you think need improvement, if any?***

***Have there been instances in which you would have preferred to delay making a decision?***

*These questions were not addressed in every interview.

Notes from the interviews were then compiled into one or more summaries, each summary covering a specific topic. These summaries were then passed back to each company for their comment and approval. Once corrected and approved, the summaries were incorporated into the following sections.

### *5.3 Overview of Results*

As was summarized in the previous chapter, SBCE consists of two primary concepts: considering a large number of design alternatives and allowing specialty groups to consider alternatives independently. Thus, interview discussions tended to focus on topics related to demonstrating that set-based concepts were either followed or not followed, and why. As the next sections will demonstrate, *no company clearly practiced both of the core principles of SBCE*. Interestingly, almost every company visited was in the midst of reforming their design process in one way or another. Several of these reform efforts included tools aimed at allowing engineers to consider a greater number of design alternatives, while others focused on the development of standardized products which could be applied to several programs. Therefore, *evidence was found demonstrating that some aerospace companies do consider sets of concepts*. Furthermore, some engineers described how engineering specialists do consider several design options from their own perspective. However, these engineers also stated that specialists typically share only one idea -- their "best" idea -- with other engineers. Hence, *these practices fall short of ideal set-based techniques which call for the sharing of multiple ideas to establish regions of overlap*.

The following sections and chapters present eleven examples based on site visits. The first presents an overview of the design process at an airframe manufacturer, and this example is used to frame a discussion of common themes seen on many of the site visits. Later sections provide examples of more specific design strategies, followed by a chapter providing descriptions of new virtual product development tools being developed at two companies. Standardized product and process development initiatives at three companies are then discussed, as are the effects of reform efforts such as Cost as an Independent Variable (CAIV). The final chapter to this portion of the thesis closes by considering additional design process constraints which aerospace companies face and then by providing an overall assessment of the set-basedness of the industry, as represented by the companies visited for this study.

## 5.4 Common Themes: An Example

A description of the traditional product development cycle at an airframe manufacturer provides a good example of several themes common throughout the aerospace industry. As will be discussed, these shared traits include the nature of the conceptual development effort, the relationships between engineers, and the effects of the government contracting environment on the design process.

At this company, design work for a new system began during the proposal stage, which typically lasted for one year and culminated in a bid for a development contract. The company began design at this stage by defining basic parameters related to the airplane's configuration: number of engines, tail arrangement, wing layout, etc. An in-house-developed computer aided design system was used to assist configuration engineers to consider a variety of concepts by making rapid parametric design changes.

Once the configuration engineers selected an initial baseline design, this concept was passed onto the structural engineers, who designed the preliminary structural arrangement. As the structural engineers began their analysis, the configuration engineers often passed on one or two more designs, each of which was analyzed in turn. It is important to note, however, that the configuration team might have considered several other designs that were not passed on to the structural engineers.

Structural engineers began their work by considering different wing structural arrangements and means of attaching wings to the fuselage, as well as how many spars and bulkheads should be incorporated into the airframe. Next, a coarse-grid finite element model (FEM) was generated, which was used to develop an initial estimate of the loads in the airframe. Historical data were also used to supplement this analysis. The FEM and loads analyses were repeated several times, with each iteration in the analyses being used to refine the design.

One structural engineer said that his group typically began with a baseline design and conducted a first analysis. He suggested that in many cases, a structural engineering team was able to determine the "best" arrangement on their first attempt at a solution based on past experience.

When a significant amount of uncertainty was present in a design, alternative arrangements were developed which explored variations in the spacing of bulkheads, variations in web or skin thickness, and possibly some different joint designs. Sketches were often used to communicate these alternatives, rather than detailed engineering drawings. These variations in design, however, were only considered for the configuration being analyzed at that time -- different configurations had their own structural alternatives.

Manufacturing engineers were brought into the design process at this point in the system's development as well. "Working side-by-side" with the structural engineers, as described by one designer, these two groups developed subassemblies, designated structural breaks for economic manufacturing, and detailed other structural and manufacturing elements. This process was described as a give and take procedure, with one group proposing a design, the other group commenting, and the design then being revised.

One engineer also noted that design reviews were a source of concept exploration during this time. He described a process in which engineers explored "ranges" of designs, allowing different engineers from different specialties to understand one another's limits and constraints. This process has become more important as systems have become more tightly integrated, forcing closer coordination between specialties. In parallel with this design analysis, preliminary schedules for the entire program (i.e., concept design through production) were developed, along with an initial top level schedule for the release of drawings.

As the time approached to submit a bid, the design group converged on a single concept from the two or three which had been analyzed, as typically required in government-sponsored concept exploration contracts. Once the company won the proposal, the selected configuration progressed into conceptual design. At this time, it was common that there would be revisions to the system's requirements. These changes would prompt the design team to consider some variations in the design, and to further update the configuration. More detail was added to the design at this point, and wind tunnel models were constructed and tested.

During this stage of development, a three-step analysis cycle drove the design process: 1) an FEM analysis was conducted, 2) which was used to generate internal loads data for the design, and 3) this data was then used to update the structural design and layout of the vehicle. One such iteration typically lasted two to four months, and three of these cycles were usually executed during the conceptual design phase, thus requiring 6-12 months for the complete process. By the end of conceptual design, the basic concept for the vehicle had been finalized. The outside moldlines of the design were fixed, as were the major structural components.

Once conceptual design was completed, the configuration engineers left the program, and preliminary design then began, further refining the selected concept. Approximately three more cycles of the FEM-loads-design process were conducted during preliminary design. Data from on-going wind tunnel tests were also used to update the concept's configuration. In parallel, major suppliers were contacted, and long-lead items were ordered. Layouts released at the end of preliminary design included fasteners, clips, and brackets; all of which had been properly sized based on stress analyses.

While the design of the vehicle itself was being completed, tool designers were beginning their work. In conjunction with that effort, need dates for parts were established, and manufacturing engineers determined in what order drawings should be released to meet the manufacturing schedule.

Engineers described the next phase of the process, detail design, as "fast paced" and with "little time to make changes." A final stress analysis was released, though its results should not have been drastically different from the last analysis (this analysis was described as the "good copy"). Tool designs were released along with final parts and assembly drawings, and additional long lead items and other production materials were ordered.

The final stages in the development process were fabrication, assembly, and testing. The fabrication and assembly of the first few vehicles provided an opportunity to address manufacturing issues with operational hardware and suggestions were often made for how to improve the design for a smooth transition into production. Sometimes these changes would be incorporated "on the fly," otherwise they might be incorporated in later vehicles.

The design process "ended" as the first airframes were completed. Flight testing would begin, and this testing might result in modifications to the design. Overall, however, the vast majority of the design effort was completed at this point.

## 5.5 Discussion

As was noted, this example highlights several common themes which were seen during almost all of the site visits. Each of these recurrent ideas will be reviewed in turn, following the sequence of the design process described above.

### 5.5.1 Conceptual Design Processes

The conceptual design process at most companies took on one of two forms. The first form was similar to the one illustrated above: a small group of engineers, akin to the configuration engineers, would develop several different concepts, each of which would be analyzed by support engineering groups. Based on the results of these analyses, the design group would narrow the number of designs under consideration, until only one design remained. The second approach to conceptual design focused on developing one design, and then passing that single concept on for analysis. In the event that the design was slightly deficient, it was modified, and then reanalyzed. If the results of the analyses were completely unsatisfactory, a new design would be developed.

Of these two conceptual design techniques, the first -- considering several options and then gradually selecting one -- is clearly more set-based than the second. Engineers using this process readily accepted the ideas behind set-based concurrent engineering, and said that they practiced a similar approach. These designers stated that the purpose behind the consideration of a large number of alternatives was specifically to "understand the trade space," as one engineer described. By carrying out a large number of conceptual trade studies, engineers gained a better understanding of what design features contributed to meeting requirements and what design features could be used to help exceed requirements.

The second approach is not without merit, however. Engineers who practiced this technique -- of considering just one idea at a time -- typically cited one of two reasons. The first reason was budget constraints. To consider a greater number of concepts usually requires more engineers be assigned to a program. This greater staffing, obviously, requires more funding. In programs without large budgets, money is often not available to allow for more engineers. The small staff is then only able to consider one or two options at a time. In these cases, engineers often expressed a desire to consider a greater number of designs, but the budget realities of their programs prevented such approaches.

The other case in which the more point-based approach was often used was when engineers had a high degree of confidence that they knew the answer at the start of the program. Such foreknowledge was typical on programs which were closely based on previous projects (as opposed to a project requiring the invention of new technologies) or on programs when a specific technology was required by the customer. In these instances, an engineer's first guess would not be far from correct relative to the final answer. Thus, rather than considering and then discarding other options, engineers were comfortable "skipping ahead" to the one solution. A further discussion of this design approach is presented in Section 6.2.

While there were some variations in these approaches to generating a design, one issue which did not vary across all the companies was engineers' emphasis on establishing a baseline design. Engineers and managers at almost every company described the importance of having a single design point around which specialty groups could base their design trades and analyses. Depending upon the company, the amount of change expected in the baseline varied from "a little" to more than "eighty-five percent." Regardless of the degree of change, however, the purpose of the baseline was always the same: to ensure that all the engineers on a program were working on the same concept so that designers did not end up working on something irrelevant to the rest of the team. Many of the managers and engineers interviewed described programs awash in chaos when a baseline was either not established or when it was improperly communicated among a design team. The baseline clearly played an important coordination and integration role at almost every company which was visited.

### 5.5.2 The Departure of the Conceptual Designers

As noted in the example, the configuration engineers -- the conceptual designers for the company -- typically left a program by start of preliminary design. This tendency to have one group of engineers responsible solely for conceptual design was seen at every company studied during this investigation. Engineers described conceptual designers as being very good at taking amorphous notions and customer requirements and translating those concepts into a concrete design. These designers, however, often tended to show little interest in the later, more detailed phases of the design process. Consequently, these conceptual designers have developed specialized roles within programs, allowing them to apply their skills, and then to depart a program when these skills are no longer needed.

This practice contrasts sharply with those at Toyota. The role of the conceptual designer at Toyota appears to be shared by both the chief engineer and the stylists. Significantly, however, both the chief engineer and a program's lead stylist will stay with a project until its very end (Sobek, 1997). This continued presence by the stylist allows him to ensure that later changes, which must be made to accommodate engineering needs, do not violate the original styling intent. His continued presence also ensures that he has the opportunity to observe what difficulties were encountered downstream because of decisions which had been made by the stylists. The lead stylist will then be able to carry this knowledge with him to his next project, ensuring a feedback loop in the design process (Sobek, 1997).

This feedback loop does not appear as strong at many of the sites visited for this research. Some engineers, in fact, openly stated that they believed conceptual designers did not receive enough feedback from downstream groups. Instead, conceptual design engineers tended to move directly from one program to the next and, as one engineer pointed out, would design features on several programs, which caused downstream problems, before they received feedback information about these difficulties.

### 5.5.3 Relationships between Engineers

As was illustrated in the example, even very early stages of an aerospace product development effort require the involvement of a diverse array of engineers. Interestingly, regardless of the formal design process at a company, it appeared that the personal communications between engineers were almost always set-based in nature. In the process illustrated above, for example, a design engineer commented that he would often seek the advice of a manufacturing engineer when he encountered a specific problem. In the ensuing discussion, the two would often discuss a variety of options, explaining to each other the advantages of one design over another from their own perspective. The discussion ended when a design was developed that met both engineers' needs.

The frequency and extent of these kinds of conversations seemed to vary across companies -- they were described by virtually every engineer at the company used for the above example, while at another company such conversations seemed more rare. Regardless of the how often such conversations occurred, however, they were clearly quite set-based in character: Each engineer would consider the design from his own perspective, make suggestions, and then the two would modify their suggestions until a suitable degree of overlap was established. Thus, both tests of SBCE are met: considering sets and working from an independent perspective. While these conversations were often about very specific features of a design, and, therefore, extrapolating across an entire development process is difficult, it seems clear that at an interpersonal level, set-based methods are fairly common in the industry.

### 5.5.4 The Impact of the Contracting Environment

Another issue raised at many of the companies was the effect of the contracting environment on the design process. As shown in the example, designers selected a single concept for presentation during the bid process. This rapid narrowing of concepts prior to submitting a bid was described at many companies, across all of the sectors in which data was collected. Engineers stated that they believed presenting multiple options for a bid would lower their standing in a competitive selection process, so only one design was formally presented to the customer.

Therefore, *from the perspective of the contractors*, government practices seem to constrain aerospace companies' abilities to carry multiple concepts for an extended period of time[7]. As described by engineers at several companies, typically after a contract award, both the winning contractor and the government would review the design and the requirements, and often changes would be made to both. Despite this period of review, however, the contractor will have committed to a design very early in the process possibly with relatively little development work to substantiate the design.

Note that this development process contrasts dramatically with the relationship between Toyota and its top tier suppliers, which provide a good analogy for the aerospace environment. In the aerospace industry, the contractors might be viewed as analogous Toyota's suppliers, while the government can thought of as analogous to Toyota itself. While the real Toyota prefers to see many options from all of its suppliers during their "bidding" process, it appears that the aerospace industry operates using a slightly different model. Contractors might consider a large number of design options, but, unlike Toyota's suppliers, only one of these options is ultimately presented to the government.

Engineers did note that depending upon the specific program, government and customer representatives would review early design options, at which time multiple alternatives would be

[7] It is important for the reader to note that this research only addressed the design process from the contractor's perspective, and not the government's. Therefore, there likely exist important reasons for the government to insist on this narrowing which were not uncovered during this investigation.

shared. One program manager stated that he had worked on several programs in which the customer stipulated that multiple designs be considered for a specified period of time. Significantly, however, these requests for multiple design options were made during the pre-proposal phases of the program. Once it was time to bid for the contract, only one design was presented. It seems, therefore, that regardless of the preferences within a company, the external contracting environment can and does have a significant impact on the number of options which are considered and the length of time during which these options remain under consideration.

### 5.5.5 The Analysis Cycle as a Driver of the Design Process

As described in the example, in the case of aircraft design, the design process itself seems to revolve around the analysis of the loads inside the airframe. One of the first detailed analyses of a design -- the first FEM model -- begins to generate this data, and the loads are also some of the last data confirmed at the end of the design process. This emphasis on loads is for good reason -- the internal loads dictate the size of the internal structure, where certain subsystems may be placed, etc. In essence, the loads data can affect virtually every aspect of an aircraft's layout. Similar analysis cycles were seen in other aerospace sectors, such as thermal analyses in satellite design, and dynamics analyses in sensors and weapon systems.

What is interesting about these analysis cycles is the effect which they seem to have on the design process. Again, the degree of the effect varied across each company, but the nature of the effect itself was same: The presence of such a complex analysis problem -- one which could often drive changes to be made in to complete system design of the product -- was often a driver for finalizing a design as rapidly as possible. For example, in aircraft design, the longer designers wait to finalize the airframe's shape, the later in the process the loads are fully characterized. Depending on the results of the loads analysis, however, changes may need to be made to the design. If this data does not come back until very late in the process, little time may be available to make the needed adjustments. But the problem is yet more complicated still: to develop good loads data requires that the design be highly detailed. Analysis engineers must wait to complete their final analyses until a product's design is almost entirely complete. Consequently, when changes do have to be made, they can often be significant, affecting multiple aspects of a product's design. Thus, there is often pressure at the very beginning of the program to finalize the design as early as possible, leaving as much time as possible to make changes and modifications based on such analyses.

### 5.5.6 Long Lead Items as Drivers of the Design Process

Finally, and along similar lines as the analysis cycles, the last common issue raised at several sites was that of long lead items. Engineers and managers in every sector described a similar dilemma associated with such components and design decisions: while designers might want to delay a decision on a part's design or on which product to order, lead times often force an early decision. If the decision were delayed, the length of time associated with finally receiving the component would delay the entire program. Designers are often pressed, therefore, to make such decisions as quickly as possible. Clearly such decisions pose risks: if changes are made later in the process, a part that was ordered will need to be scrapped or sent back to the vendor, etc. But if no decision is made, the consequence -- delay -- is assured. Thus engineers described instances in which they had to commit to a design earlier than they might have liked. Interestingly, however, no instances of severe consequences were cited based on such decisions.

## *5.6 Common Themes: In Closing…*

Product development practices varied widely across companies. Despite their many differences, however, common trends were observed across almost all of them. To an extent these similarities are not surprising -- the complex nature of the products is similar between sectors, even if their technologies are not. At the same time, however, the degree of these similarities was somewhat surprising to the author. Regardless, the fact that many of the underlying notions were similar across the sectors implies that a general model for implementing SBCE is not unreasonable, and this model is presented in Chapter 11.

# 6. Examples of Design Strategies

## *6.1 Chapter Introduction*

The previous chapter presented on overview of an aerospace design cycle. The following sections present more focused, detailed accounts of specific approaches to design within several companies. Four examples are presented from different companies. The first two tend more toward the point-based model of design, while the second two illustrate more set-based approaches.

## *6.2 Example 1: Capitalizing on the Point Design*

### 6.2.1 Introduction and the Underlying Theme: Speed

One company's philosophy for design has focused on the development of "point designs," aerospace systems which excel at very specific missions. Their approach to the design process has evolved to match the development of these highly specialized systems. As will be seen, one underlying theme described by designers and managers was an emphasis on schedule. Engineers all stressed the need to develop the system in as short a time as possible. This focus on a short design cycle time, they believe, allows them to limit the cost of their development effort and to deliver operational systems to their customers in a timely fashion. Three additional key features of their design approach stand out from other companies' processes: requirements, design decision making, and people. Each of these features will be discussed in turn.

### 6.2.2 Approach to Requirements and the Limitation of the Method

To be successful, engineers emphasized that their development process demands a specific approach to requirements: the customer should specify *what* the system must do, but *cannot* specify *how* the system should do it. To facilitate this practice, the company prefers that the customer specify as few requirements as possible at the start of development. In addition, the customer should only specify top level system requirements -- the “what’s” and not the “how’s.” These system level requirements should also be verifiable, such that designers will be able to easily determine when their design has satisfied their customer’s needs.

Development of more detailed subsystem specifications must be left to the design team itself. As the design evolves, the team develops the appropriate subsystem requirements. If the customer had specified these detailed requirements at the start of development, engineers stated that their design space would be tightly constrained, limiting their ability to develop designs which would fully satisfy the customer’s primary needs. If these needs are truly the important ones, the company’s engineers argue, they should be allowed to sacrifice other aspects of the design to satisfy these most important requirements.

This approach to requirements definition also defines the primary limit of this design method. To be effective, designers at this company prefer that the customer only specify one or two characteristics that are the most important – speed, survivability, maneuverability, etc. Designers will then trade away performance in other areas to maximize performance relative to these one or two most important requirements. The products, therefore, tend to be point designs: they perform one or two mission exceptionally well, but have limited flexibility to perform other roles.

One engineer used the analogy of islands in the ocean to demonstrate this point. In this analogy, requirements can be viewed as the ocean’s height. Each individual island represent a different mission (Figure 19). These individual islands, however, might turn out to share a common base, which would become visible if the ocean was lower (Figure 20). In design, therefore, establishing demanding requirements relative to several missions might push the water level up very high, requiring designers to select one mission over any others.

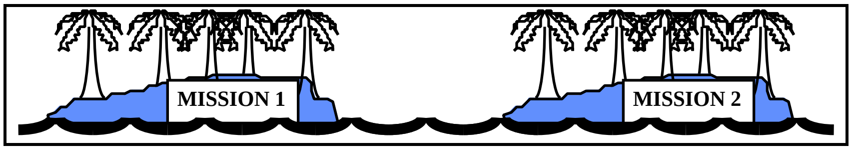



**Figure 19: Demanding requirements force choices.** By specifying very demanding requirements, a customer may force the designer to select one of two possible high-performance missions.

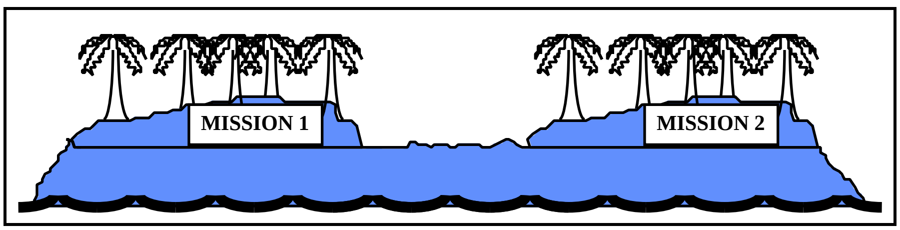



**Figure 20: Less demanding requirements allow fulfillment of multiple missions.** Reducing the requirements allows a reduction in performance relative to each mission, revealing areas of overlap in design or performance, so that both can filled.

This need to select only one island, i.e., mission, if the requirements are demanding is what drives the company to point designs. If the requirements were lowered, the system could perform more missions, but it would probably not perform any of them particularly well. This company's philosophy, however, is that it is better to perform a few missions very well rather than to perform several poorly.

One further risk which this approach to design runs, however, is that one lone island (i.e., a specialized design) may *not* share a common base with other islands (be adaptable to other missions). Were one to reduce performance requirements in an attempt to uncover other missions and roles, it might be discovered that the system is so specialized that it can do no missions beyond the one for which it was originally designed (Figure 21). Designers were able to describe several instances of this occurring. In one case, the product had been deployed and used very successfully. In an effort to sell more units, engineers attempted to modify the design so that it would be more flexible. While several studies were conducted, no additional systems were fielded: The studies revealed that in fact the design could not be readily modified to accomplish other missions. The product was a true point design. Engineers at the company believed, however, that such risks were worth running in order to develop exceptional designs.

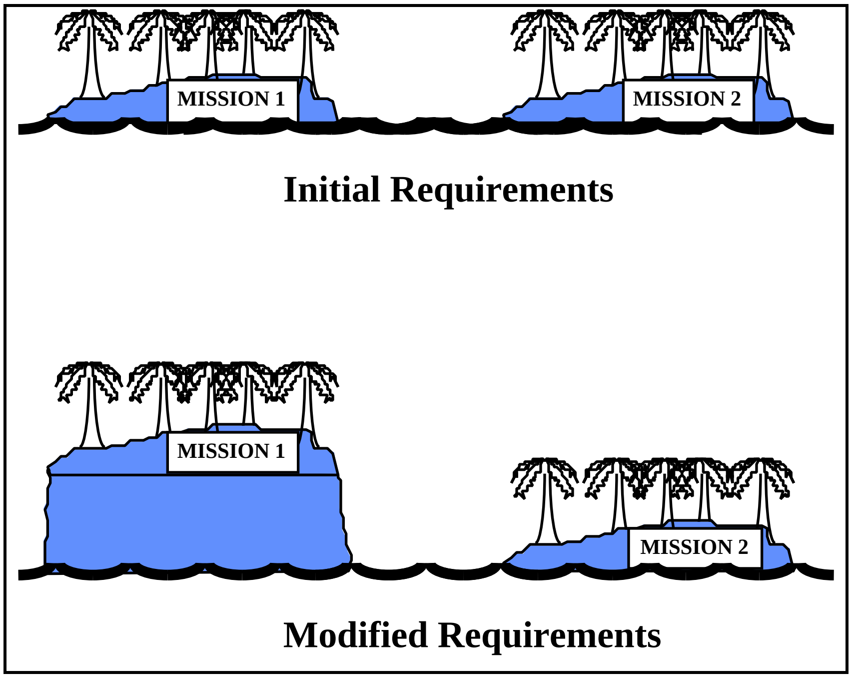



**Figure 21: The risk of a point design.** Even though requirements are reduced, the design still cannot satisfy both missions.

#### 6.2.3 "Just Do It" Design: The 80/20 Rule, Select an Option Quickly, and Plan for Success

Although detailed approaches used by specific design teams vary across the company, one overarching attitude toward design seemed to emerge from all of them: "just do it," or, expressed another way, "go make it work." Managers and engineers at the company stress that their approach to design focuses on moving through the design process as quickly as possible. This philosophy seems to find expression in several aspects of their design approach: the 80/20 rule, rapid development of baseline concepts, and the rapid completion of trade studies.

Engineers and managers at this company consistently expressed a belief in the 80/20 rule. As described above, the company tries to work on programs which emphasize one or two requirements more than all others. Such a specific focus allows designers to subvert all other aspects of the project's performance to meet these most important requirements. Their design approach, therefore, prescribes that it is best to meet 80% of the customer's ideal goals with 20% of the effort and quickly field the system rather than struggling to fulfill the last 20% of the objectives with 80% of the effort. Instead of increasing the program's schedule and cost to achieve this last 20% of performance and optimize the design, managers and engineers preferred to deliver the system faster and cheaper. As one manager expressed, a design approach which attempts to find the "end all, be all solution" to a design problem represents the "antithesis" of this company's approach. Once a system is fielded, if the last 20% of performance is required it could be incorporated through modifications. The company's belief, however, is that it is better to add such performance later, rather than struggling to incorporate it early.

Within this framework of the 80/20 principle, the first aspect of the company's design approach is to develop a baseline concept as quickly as possible. The baseline establishes a common framework for a program, including its schedule, costs, design concepts, manufacturing issues, etc. One manager stated that the purpose of the baseline is simply to "get everyone marching to the same drum." Team members know that the baseline design will change , perhaps by as much as 85%, but the baseline is not intended to be a final solution. Instead, it is intended to be a communications tool, to be used to coordinate the efforts of the team members.

In addition, the development of the baseline provides an opportunity to understand "what's down the road," as expressed by a development manager. By going through the exercise of developing the baseline, engineers have the opportunity to preview what problems will be encountered later during the design process and what variables or design parameters will be the most important. Several engineers also pointed out that there are times when they are unsure of which questions they should even be asking at the start of the design process. Instead, they must "go down the line" to determine what issues must be addressed. The development of the baseline, therefore, provides designers with an opportunity to learn what issues will be important, improving the following design effort.

As the baseline is developed, specialty engineers and designers begin considering different design options and conduct trade studies to evaluate these options. The options are based on what one manager called a "technology basket." These baskets identify several technologies and estimate when those technologies will be operational. Technologies are then only considered during trade studies if their availability will meet the need date for the system under development. Programs will not necessarily use the term technology basket, but most do tend to follow the associated process.

As in other aspects of this company's design process, the emphasis when conducting these trade studies is to do the trades quickly. Managers suggested that their strategy for trade studies was to "do enough engineering to understand the trade space," but not to necessarily explore all of it. They also stressed that prior to the start of any trade study, trade study criteria must be established. These criteria are critical because they establish natural exit points for trade studies. Once a criterion has been met, the trade study can be stopped, since a design which meets the requirements has been found.

This trade study approach returns to the idea that designs should meet 80% of the customer's ideal goals, but should not necessarily be optimized. Rather than keeping the trade study going, managers prefer that their engineers select a design as quickly as possible. In this context, one manager suggested that his role was to first allow his engineers to study a problem, but then to "corral" them to select a solution.

If during the development of a system a better option is discovered, however, engineers and managers said they would not hesitate to incorporate it. One manager summarized the approach as "funnel [the design space] early," but then reexamine the decisions later, "just to be sure." When the design is reexamined, this manager said he has in fact made changes on several occasions. But, to be able to make such changes requires a firm baseline design, to which changes can be applied. If the baseline did not exist, changes might be made at random and for far too long.

These approaches to trade studies and incorporating improvements find further expression in the company's method for design problem solving. In this context, problem solving refers to how engineers and designers cope with design failures, such as wind tunnel data revealing an unexpected flutter problem or an FEM analysis revealing that a structure will fail under too light a load. Engineers described their strategy under such circumstances as planning "on success." Designers and engineers make their best "guess" (which is backed by analysis) and then go test the solution. They assume that the design concept will meet the requirements during the test. If the concept does falls short, engineers then go back and look at what can be done locally in the design space to fix the design.

In one development program, for example, wind tunnel data revealed a severe control problem for an air vehicle. Engineers did not reevaluate the entire vehicle configuration, but instead searched for solutions which could be incorporated into the existing design. Prior to developing these solutions, however, engineers established their trade study criteria, in this case based on the moment coefficient and weight (which would reflect changes in performance). Twenty-two different alternatives were developed, tested, and discussed. Several options satisfied the trade study criteria, and engineers then used their experience to select the one they believed most appropriate. Interestingly, engineers and managers in several programs referred to this type of process. In general, they all seemed to share a common belief: quickly develop a design, test it to uncover problems, and then fix the problems.

### 6.2.4 Making the Method Work: Robustness, People, and Small Teams

The engineers and program managers who described these processes cited a variety of factors which enabled the methods to work effectively. Three appear to be the most important: robustness , people, and small teams.

The concept of robust design was raised by conceptual designers at the company. Since the design process stresses the need to make decisions as soon as possible, the process does run the risk of a wrong decision being made early in a product's development. As just described, engineers focus on making specific modifications to a concept late in the development process to compensate for such possibilities. In conceptual design, therefore, it is important that a design be robust enough to accommodate such modifications without forcing a complete redesign. Note that in this case, then, that this use of robustness is similar to the one used in SBCE: It refers to the design's ability to adapt to modifications which might be instituted by other engineers.

Configurationists suggested that the best way to achieve such robustness is to "center designs" in the design space. Rather than pushing product concepts to perform better relative to some measures than to others, practice at this company tends to be to balance the performance between several measures. This design centering is manifested through the use of large margins for the design's performance measures, such as weight or power. Since the overall product concept is intended to perform best along one or two measures, however, the need for robustness applies primarily to other aspects of the design. In fact, these other features of the design must be robust, because, as discussed earlier, these features will be modified to achieve the primary design objectives.

In almost every interview, engineers and managers emphasized their reliance on people. As described, the company's design methods rely on rapid, but knowledgeable decision making. Therefore, the method depends upon the ability of engineers to make good decisions without the need for too much analysis, a skill which they develop with experience. Engineers at the company believe that their experience limits the likelihood that incorrect design paths will be chosen. In addition, this experience minimizes "perturbations" in the design process, such as being able to handle problems downstream during design.

To take advantage of this experience and design skill, the company organizes project personnel into small design teams, the final important aspect of their design strategy. Several managers described that the short lines of communication and the minimum number of interfaces facilitated by small teams reduces the likelihood of errors being introduced into the process. Small teams also mean that each engineer becomes responsible for a significant amount of the design effort. This responsibility then both empowers individuals to make design decisions, and, since they will be held accountable for the decisions, to make well-planned and timely decisions. Finally, the design teams are usually co-located, at least at the beginning of a project. Numerous engineers testified about the need to be able to gauge the responses and information provided by other engineers. Such personality intuition, they said, was often crucial in judging whether or not another engineer's design or analysis might change. Thus, at a fundamental level, this company's design strategy relies on nothing less complicated than the technical and communication skills of its engineers to facilitate a design process emphasizing a rapid build, test, fix cycle.

### *6.3 Discussion of Example 1*

This company's approach to design is clearly better described by the point-based model than the set-based model, although there are some elements of both. Consider first its similarity to the point-based approach. In this model of design, engineers brainstorm multiple options, then select the best one. If problems are found downstream with the selected option, changes are instituted locally, modifying specific features of the design to improve it relative to the need measures. This model appears to describe this company's approach quite well. Engineers conduct quick trade studies so that they can select a design as quickly as possible. Then, as was illustrated, when problems are encountered during testing, small changes are made to fix each specific problem. In its most basic features, this design method illustrates a point-based approach.

Note, however, that several additional aspects of the company's product development philosophy are well-suited to this design method. Perhaps most importantly, the company's designs tend to emphasize just one or two features above all others. Set-based methods, in contrast, are intended to help designers balance multiple, conflicting requirements. In the programs pursued by this company, such issues are less of a concern (although many of their designs do require tradeoffs between several parameters). Thus, it seems quite appropriate to avoid the expense attributable to considering large numbers of options.

Further, the company does make use of other set-based practices to help cope with some of the challenges raised by other aspects of its methods. Their emphasis on top-level requirements, for example, is similar to set-based approaches to requirements. Mangers and engineers noted the importance of having only the most important requirements explicitly stated by the customer so as to ensure design freedom downstream in the process. This line of reason is very similar to set-based approaches to requirements. In addition, engineers' references to robustness, which essentially matched the definition of conceptual robustness given in Chapter 3, illustrate a second set-based practice. By using margins around critical features of a design, upstream engineers help to ensure that changes made downstream do not invalidate the earlier work. Again, the concepts behind these actions are consistent with the basic concepts of SBCE.

In summary, it is clear that this company's approach is better aligned with the point-based model than with the set-based. Their approach to design, however, does make use of set-based concepts to help facilitate other point-based practices. Thus, this example helps to illustrate situations in which a point-based approach might work best: specifically, when there are a limited number of requirements, allowing engineers to trade other performance features to ensure success among just a select few. And, finally, this example also helps to point out an important lesson: that point-based and set-based methods are not mutually exclusive, but rather can be mutually supportive.

## 6.4 Example 2: Design by Constraint

### 6.4.1 Introduction

A program manager at one company has developed a flexible design approach intended to be adaptable to the particular requirements and goals of each product development effort. As he described, typical approaches to design focus on defining the *solution* -- what the system will look like, how it will execute the needed functions, etc. These solutions are developed to meet the requirements defined for the system. The goal, therefore, is to develop the highest performing system relative to those requirements.

This manager's approach, however, works from a slightly different perspective. Rather than initially working from requirements to define solutions, this method uses the requirements to define *constraints*. Since the requirements are defined by the customer, the constraints also represent the customer's needs and desires. These constraints then dictate how a design concept should evolve and how the design process itself should be adapted to each particular design problem. By modifying the design process, one ensures that the system design evolves as a direct reflection of the constraints, and, therefore, the customer's requirements. While steps A, B, and C might all be required to arrive at a final system design, the sequence in which these steps are executed may change. The goal, therefore, is to develop a balanced system meeting the needs of the customer.

### 6.4.2 A Multi-Tiered Approach

This manager's approach to design operates at two, tiered levels. At the higher level, the process is based around the concept of systems engineering. The second level focuses on sequence of the design process steps. Each of these tiers will be discussed in turn.

The manager described his approach to systems engineering as follows. Customer's needs are first defined and clarified, and these needs are then used to develop product concepts. Based on these concepts, the total system is decomposed into smaller subsystems. The approach emphasizes that it is better to select only one system-level concept, and then develop a variety of subsystem options and combinations, rather than attempting to manage the pursuit of multiple system-level designs. As concepts (both system and subsystem) are developed, they are compared to the performance and interface requirements, which are derived from the customer's needs. Later steps in the process focus on managing the configuration (by thoroughly documenting requirements, designs, and decisions) and balancing the system, (i.e., ensuring that decisions made at the subsystem level do not adversely affect the complete system's performance). Finally, the design must be verified and validated, to confirm that it does meet all requirements and specifications.

This process is then repeated for each subsystem, beginning again with the definition of requirements for that particular subsystem, and then following the other steps as described above. Since the second step in the process is to generate system concepts, subsystems can begin concurrent development as soon as their interfaces and functional requirements have been defined. During the initial development of these subsystem, designers are free to make their own design decisions, limiting the number of additional groups that must approve of changes. Once a subsystem has been released to the system, however, other design groups directly affected by the subsystem should review the design. Note, however, that is not necessary for all other subsystem design groups to participate in the review -- only those that are directly affected by the subsystem.

The decomposition process then allows for the second tier of the method to come into play, the sequencing of design activities. For a given product composed of numerous subsystems, all of the subsystems will have to be designed and developed. Depending upon the exact requirements for the product, however, the sequence in which these activities will occur may vary. Different customers will likely have slightly different requirements. These differences in requirements may change which product features or subsystems are more important. Product characteristics which are more important to the customer must then drive the design and development of the entire system, such that the system is an accurate reflection of the customer's desires.

To allow these customer-defined features to dominate the product's development, the *design process itself* must be adapted. Features or systems most important to the customer should be addressed first, so that these systems then establish constraints for other aspects of the design. If the design process is fixed without regard for the customer, earlier design decisions might limit the ability of later decisions to accurately reflect the customer's needs. By rearranging the process as needed, such arbitrary constraints can be eliminated, allowing the design to fully address the desires of the customer.

### 6.4.3 An Example

An example helps to illustrate these concepts. One design problem to which the method was applied was the conceptual design of a low Earth orbit (LEO) spacecraft. Aside from the specific mission to be performed by the spacecraft, the customer specified that low cost was the most important requirement.

As the program manager described, a more traditional approach to design might have begun by addressing system or subsystem details about the spacecraft, and how these details could be traded to arrive at a low-cost design. In such circumstances, specialty engineering groups would develop design alternatives for their respective subsystems and select the ones with the lowest cost. Such details, however, are not the only determinants of a spacecraft's cost.

Rather than starting from spacecraft design issues, *per se*, the manager first addressed the customer's highest priority: low cost. Cost, therefore, was the first constraint defined for the spacecraft. All other design decisions flowed down based on this top objective. Thus, the first question addressed was, what contributed to spacecraft cost?

Among other design issues, a major driver of the spacecraft cost was launch cost. Therefore, the first step in containing the cost of the design was to contain the launch costs for the system. This goal was historically accomplished by using a smaller, rather than a larger, launch vehicle. To use a smaller launch vehicle and meet system mission requirements, however, required that the designers maximize the use of available volume within the spacecraft to minimize its total size.

At this point, design on the vehicle itself could begin. Since volume was defined as a constraint (based on minimizing launch costs which was derived from minimizing the system cost), the first steps in the design process addressed how the vehicle's volume would be used. The design team began by considering the cross-section configuration for the spacecraft. Using this area efficiently would enable the overall volume of the spacecraft to be minimized. In the course of developing this cross-section, initial details of the spacecraft design were considered for the first time, such as stowage of solar arrays and mission antennas, stiffness of the spacecraft's structure, and the desired maximum spacecraft mass.

Once the cross-section had been determined, the next issue addressed was the spacecraft's overall length. Again, the goal was to minimize spacecraft size by maximizing the use of spacecraft volume. Since the cross-section had been established, fixing the length of the spacecraft determined the maximum total solar array area which would be available, as well as the total mission antenna area. Together with the launch vehicle fairing constraints, the requirements for the array and antenna areas determined the minimum length of the spacecraft.

Note that the mission antenna area was dictated by the customer's initial mission requirements. Thus, this point in the design process represented one of the first opportunities for the *customer* to understand what design trades his requirements forced. If, for example, the antenna could have been reduced in size by changing the mission requirements, the vehicle length or cross-section could have been reduced. Such a reduction could have potentially allowed for the use of an even smaller launcher, further lowering costs. On the other hand, if the customer wanted to change the mission requirements such that a larger antenna had been required, engineers could have quickly estimated the effects on the system's costs based on whether or not a larger launch vehicle would have been required. Thus communication between designers and customers was improved through the use of this documented, structured method.

The manner in which design trades on spacecraft systems were handled at this point also highlights how this method forced engineers to view the design from the customer's perspective. The design of the solar array provides an example. Had the responsible engineer been told to simply maximize the power available from the array, he likely would have designed the largest, most efficient array that was technically feasible. In this case, such an array would probably have cost too much for the customer and have required a very large launch vehicle. If the engineer had been told to design the lowest cost array possible, he might have re-used on old design which was cheap but had very poor performance. Such a design, therefore, would not have fulfilled the customer's mission requirements. The engineer could have been given both requirements -- maximize power and minimize cost -- but, again, the solution would likely have fallen short. Given both requirements, the engineer would have done everything he could to meet them, but he would have done so *based only on what he knew about solar array design, not the design of the total spacecraft system.*

Instead, as the design process was executed, the solar array designer understood that the constraints were not quite as simple as maximize power and minimize cost. Rather than having to consider these options from the solar array-perspective alone, the issues were set against the backdrop of *minimizing system cost* by *using the smallest possible launch vehicle* by *maximizing the use of a constrained volume*. Thus, the designer did not necessarily have to design small arrays, but he had to design arrays which could be stowed in a small area. At the same time, if he required slightly larger arrays he could influence the design by suggesting an increase in the spacecraft's length. In this manner, trades on subsystems could occur from a *system's perspective* and with a focus on the *customer's requirements*.

The rest of the conceptual design process is shown Figure 22. Note that as the design progressed, the issues addressed became less and less of a concern relative to the customer's mission requirements. Initially, the most important customer requirement was cost, which was addressed by minimizing the launch costs. Sizing of the spacecraft within that constraint was then dictated by the required size of the mission antenna and the power requirements for its associated subsystems. Later design decisions dealt with issues such as payload mounting, attitude control, and propulsion. As the design progressed through these steps, the customer became less interested in the details of the design. For example, the customer did not necessarily have a preferred propulsion system concept. He did, however, want to be guaranteed that the spacecraft would be reliably positioned in time and space for the lifetime of the system. These needs -- reliability, position accuracy, lifetime, etc. -- were the customer's concern, but such needs did not pre-specify a particular choice of propulsion system.

The final step shown in the design flow, the structural architecture, did not concern the customer at all, other than how this architecture met the other needs of the system. Note, however, that these last design decisions were also the most constrained. The size of the spacecraft had already been determined, as had the desired stiffness, center of mass, and the points of maximum loads (such as where the propulsion system had been mounted). The trade space of the structural designers, therefore, was more constrained than had been the case for any other system. Importantly, the design of the spacecraft's structure was also of virtually no concern on the part of the customer. While the customer would not have accepted compromises on items such as the spacecraft antenna (which directly affected mission performance), structural engineers could make as many compromises as they wished, so long as none of the previous design decisions were invalidated.

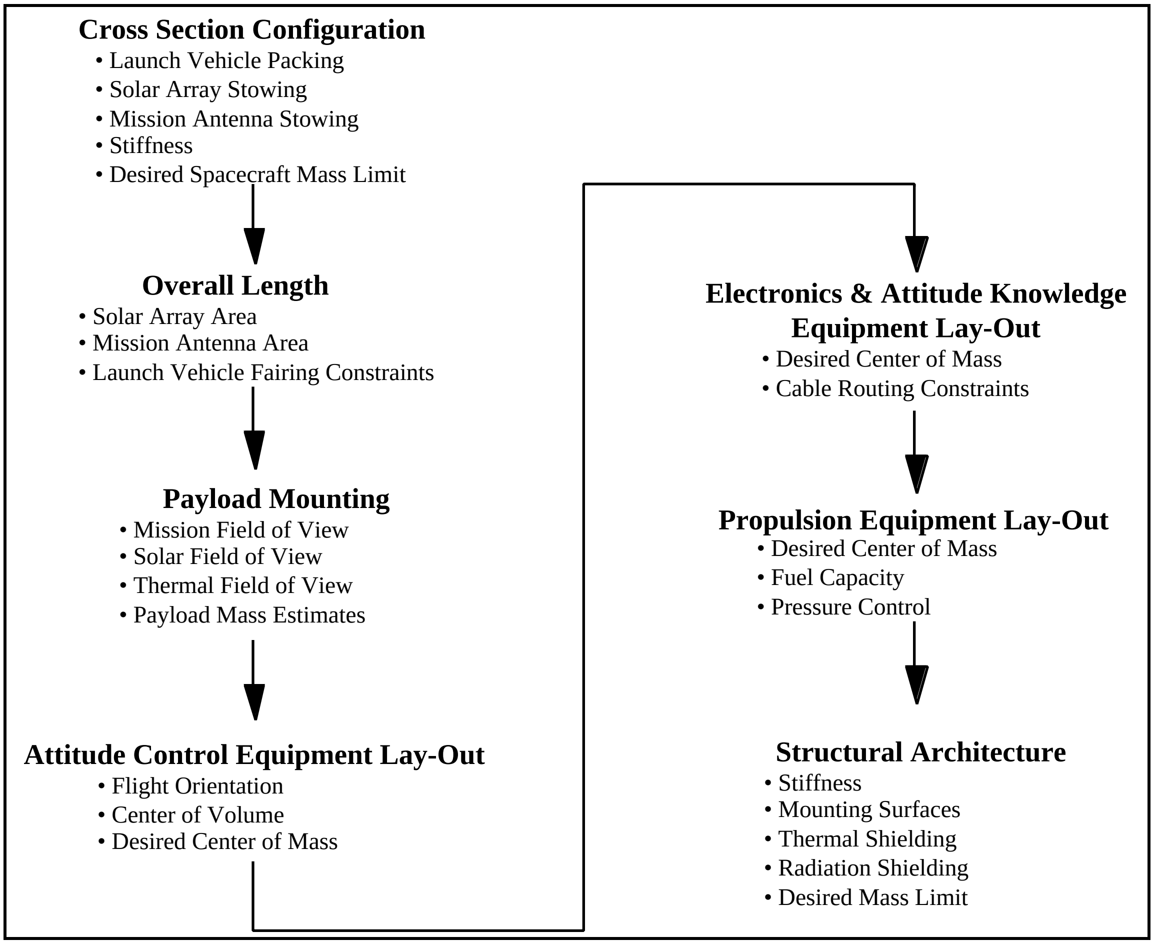



**Figure 22: Design process flow.** This flow diagram illustrates the primary steps and considerations for the conceptual design of a LEO spacecraft, given that the design was subjected to tight volume constraints as a result of the customer's low cost requirement.

### 6.4.4 Additional Issues

The manager who developed this design method also stressed three additional issues: documentation, iteration, and people. The importance of documenting design decisions can best be illustrated by reconsidering the structural design. Since the structural design was the most constrained, it also ran the highest risk of invalidating many other decisions if a design solution could not be found. To provide insurance in the event of such a problem, the program manager running the project emphasized the need to thoroughly document design decisions. In the event of a downstream problem, documentation would enable the engineers to understand what decisions they had made and why they had made the decisions. Perhaps more importantly, documentation would also allow them to select another option quickly, without having to repeat analyses which had already been completed. If such documentation were not available, engineers would have had a difficult time retracing their steps and would have run a high risk of redoing work that they had already done but not recorded. Documenting all analyses, design trades, and design alternatives, therefore, was an important theme within this approach to design.

Another important aspect emphasized by the manager was the nature of design iteration. In general, the manager did not support carrying multiple system-level alternatives throughout the design process. Doing so would increase the cost of the development effort and could potentially lead to confusion. Instead, he preferred that subsystem engineers consider several design options, conduct trade studies, select a preferred option, and then document everything related to the decision. Then, should a problem occur later in the design process, the trade space was not simply expanded at that point -- the entire process began again.

For example, should the propulsion group have conducted their analysis and determined that they could not meet the system requirements as then defined, the search for solutions would not have been constrained to simply evaluating esoteric propulsion concepts. Nor would each subsystem search frantically for what it could contribute to solving the problem. Instead, a new constraint would be placed on the design based on the propulsion limits, and the process restarted at the very beginning. As the manager stated, such problems tended not to be a result of never considering “the ideal solution.” Instead, that “ideal solution” was not chosen because the wrong criteria had been used to make a selection. Thus, refining the constraints placed on the system through iteration would modify the criteria used to select design options, usually leading to a valid solution being chosen. Since every design decision would have been well-documented, early steps could have been repeated much more quickly than they had the first time, limiting the loss in schedule caused by the rework. The fundamental objective is not to search until the “ideal solution” has been identified and justified, but rather to retain flexibility and objectivity, so that a valid solution is developed and accepted by the customer.

Finally, the manager stressed the importance of ensuring that the people working on a program understand the design process and what motivates it. Since this design process must be adapted to each specific design problem, it is not a method that can be memorized and repeated the same way every time. Program leaders and engineers must be trained to understand the overarching concepts of the method -- such as understanding the customer's needs, developing system alternatives, decomposing the system, etc. -- and they must also be taught how to adapt the details of these concepts to address a particular design problem. The program leaders must be aware of the sensitivity of the method to the initial customer requirements and to program and/or company imposed constraints so that the desired flexibility is retained. Furthermore, the inherent iteration process must be accepted by the program leaders and the reassessment of early decisions permitted without reproach. In addition, engineers must be trained to fully document their design work and decisions. If engineers do not understand these needs for flexibility and accountability, the process will flounder, preventing efficient accomplishment their goals.

### *6.5 Discussion of Example 2*

As was the case with Example 1, this method is more closely aligned with the point-based model, but, as was also true of Example 1, this approach also demonstrates a variety of set-based techniques. The approach is point-based due to its reliance on selecting a single design as quickly as possible and then using iteration to correct any deficiencies with the design. Note, however, that this method's use of iteration is different than the one used in Example 1. In Example 1, when a problem was uncovered in the design, engineers rapidly considered a large number of options to fix the problem without significantly changing the rest of the design. The method of Example 2, in contrast, uses a much more comprehensive approach to iteration: when a problem is uncovered, not only are immediate solutions considered, but the entire design process to that point is reviewed, and, if needed, redone. But within this more extensive iterative process, a similar philosophy to the 80/20 rule is applied: As was described, the goal is not an “ideal” solution, but one which simply satisfies the customer’s needs.

In the context of this drive to a single solution which is optimized through iteration, the Example 2 method also includes several set-based techniques. The first of these is that subsystem engineers are allowed to work independently until their design is “released” to the rest of the team. This relative autonomy of subsystem teams was not described at other sites, and is similar to set-based concepts of team independence.

The primary distinction between the set-based method and this one, however, is that in the set-based model the subsystem teams share multiple concepts with another, while in this approach the subsystem teams only share their one, released concept. As was noted above, the manager stated a very pragmatic motivation behind this limitation: limiting confusion. At times when subsystems had shared multiple design concepts, this manager has seen inter-team communication become confused and strained. In an effect to ensure easy communication, therefore, subsystem teams discuss only their best option. Note, however, that they retain their analyses for other design options, should the selected option not work as planned. Thus this approach to design combines some features of point-based and set-based approaches to achieve a flexible, but manageable design process.

Finally, Example 2 is unique among the others seen during this research in its complete subversion of the design process to the needs of the customer. The use of constraints to order the steps in the process and then limit the decisions available to engineers was not explicitly seen at any other site. This concept of design by constraint will be discussed again during the development of the SBCE model in Chapter 11.

## *6.6 Example 3: Combining Parallel Concept Development with Conceptual Robustness*

Engineers at one company described two approaches to design which they commonly use to help ensure that their final solution is a correct one: the parallel (i.e., simultaneous) development of alternative design concepts and conceptual robustness.

### 6.6.1 Development of Parallel Design Concepts to Explore the Design Space

Several engineers at the company described the importance of considering multiple design alternatives throughout the evolution of a product's design. While many companies conduct trade studies to determine a design concept's sensitivity to various parameters and requirements, this company follows a slightly different model. In their model, trade studies are not only conducted based upon one configuration to understand its sensitivities; instead, substantially different configurations are developed to explore specific regions of the design space.

The first step in this process is to establish a baseline for the system's design. The baseline is the result of extensive preliminary design studies and trades, and may quite closely represent the final product. This baseline serves as the primary focus of the program's effort, and it also serves as the starting point for further design studies.

Once the baseline is established, separate teams of engineers are assigned to the program to explore specific features of the product's design. These teams first select a particular requirement or performance measure. They then develop a new concept, rooted in the baseline design, which attempts to maximize performance relative to the selected parameter. For example, one concept might be developed to determine what would happen if the system were made as affordable as possible, while another concept might explore extreme performance features. Some of these studies result in only a few changes being made to the baseline design; they are used primarily educate the company about the benefits and sacrifices it has made in its baseline configuration.

Other studies, however, have revealed methods for improving the baseline design, and elements of these exploratory designs were then incorporated into the baseline.

### 6.6.2 Conceptual Robustness

Several engineers also discussed the importance of conceptual robustness in their design concepts. As it was defined previously, in this setting robustness refers to both the product's ability to function in a wide range of manufacturing and operating environments and to the design's ability to adapt to modifications which might be instituted later in its development. Engineers at this company stressed that since some design decisions must be made prior to their impacts on downstream design issues being completely understood, they attempt to make design choices which will be best suited for accommodating future alterations.

In a trade study during a past aircraft program, for example, three different tail-wing-flight control configurations were initially explored. When the analysis for the three arrangements were compared using the airframe's initial evaluation criteria (such as size, weight, and performance requirements) no advantage could be seen in one design over the others from a pure design standpoint. Considering the potential for future modifications, however, revealed that one design would accommodate changes (such as increases in weight) better than the other two. This concept, therefore, was selected for further development.

Another aspect of conceptual robustness described by engineers involved the use of an open architecture for many subsystems. Engineers recognized that an airframe will likely be in service over a period of time which sees the development of many subsystem improvements. The company manages this potential for change by allowing for growth in the subsystems and by planning for changes to be made in these subsystems. Such advanced planning reduces the risk that the complete system will become obsolete. Instead, individual subsystems can be replaced and updated, without forcing the development of an entirely new system.

### *6.7 Discussion of Example 3*

Example 3 demonstrates once again that aerospace firms do make use of set-based methods at various points in a product's development. In addition, the example reinforces ideas introduced in Example 1 about conceptual robustness.

As has been described in other examples, engineers in aerospace firms do tend to consider multiple design options throughout the development of a new product. This company's approach, however, is slightly different from methods seen elsewhere. Typically, multiple options are considered by other firms when problems are discovered in a design -- its aerodynamics are not good enough, it does not dissipate heat well enough, etc. In Example 3, on the other hand, alternatives are explored as part of the basic nature of the design's evolution, regardless of the existence or absence of other problems. Instead, the alternatives are explored to seek opportunities to improve the design relative to the needs of the customer -- to look for ways in which the design might exceed the customer's requirements.

In addition, the nature of this exploration process illustrates some of the set-based notion of specialty independence. In the complex world of modern engineering design, a "specialty" group can mean many things. Traditional specialties include engineering organizations, such as aerodynamics, thermal, or structures. More recently, specialty groups such as affordability and maintainability have taken on greater importance.

In Example 3, it is these more recent specialty groups which tend to dominate the development of alternative designs. Due to the complex nature of the product under development, these specialty groups cannot work in complete isolation -- they simply do not possess all of the knowledge required to develop an alternative product concept. Thus, they require the support of other engineers. A given alternative design, however, may be dominated by the concerns of one of the specialties. The results of this development exercise are then compared to the baseline, and improvements are made. While not exactly the same as the concepts used in set-based design, this approach does illustrate similar ideas: one group explores the design from their own perspective, generates alternatives, and then compares this set to another -- in this case, the baseline design -- to find regions of overlap.

Furthermore, the notion of conceptual robustness -- a term used by the engineers at the company -- does exactly match the concept as described in Chapter 3. Designers all described attempts to make decisions which remain valid, regardless of the choices made by downstream engineers. Thus these efforts match the idea of conceptual robustness described for SBCE. When taken together with the method of developing alternatives, practices at this company do illustrate some basic set-based principles, even if they do not represent set-based concurrent engineering in its purest form.

## *6.8 Example 4: Subsystem Installation*

### 6.8.1 Overview of the Design Dilemma

At one company, the subsystem installation group has traditionally not been brought into a development program until just prior to or just after contract award. At that time, a new system's configuration has already been analyzed through numerous trade studies and the design of major structural elements is nearly completed. In addition, all of the subsystems themselves have been sized and suppliers are being put under contract. The group's job, therefore, is to "make everything fit," clearly not a simple task.

As might be expected, a common occurrence is for two subsystem design engineers to want the same location in a vehicle for two different components. These location preferences are driven by issues such as tolerances to vibration, temperature, and noise. Since the installation group must eventually install all of the components, they will obtain the limits of these tolerances from each component or subsystem engineer. These sensitivities are often given as maximums or minimums, helping the installation engineer to understand the possible impacts of any changes that must be made. This process is further complicated, however, by the number of components and the degree to which the rest of the system's design has already been finalized.

### 6.8.2 New Developments

To help ease this process, the installation group is developing its own software package. Initially, this software tool will help the installation engineer design the routing paths for the wiring which runs between the subsystems. The user will input all of the black boxes which need to be incorporated into the system, where these boxes will be placed, and with which signals they are compatible and incompatible. In addition, each of the subsystem specialty groups will input their primary rules and constraints on where boxes can be placed. The program then determines the best routing schemes for all of the wiring and tubing throughout the system. Once this portion of the program is operational, the group plans to add a feature that will automatically place the boxes themselves in the system.

The desirability of a computer tool for this placement process was illustrated by way of example. Typically, a system developed by the company will have 300 to 400 electronic boxes. About 100 of these boxes might be constrained to a specific location for one reason or another. The other 200 to 300, however, can be placed all over the system. The number of possible combinations, therefore, rapidly becomes far too large to be manipulated by a human. It is anticipated that the computer system will be able to take advantage of this flexibility in the design possibilities, while at the same time reducing the time required to complete the design.

The engineer describing this computer tool specifically cited the desire to delay decisions as a motivation for the program. Currently, it requires approximately two years to plan the routing and to design the harnesses for the wiring. The computer tool is aimed at reducing this time to a few days, allowing the group to wait longer before beginning their work and then allowing them to complete the work more quickly.

### *6.9 Discussion of Example 4*

Example 4 is significant in that it illustrates several set-based concepts. First, because of its nature, the type of design problem presented in this example lends itself to set-based approaches: there are a large number of design variables, many possible combinations, and conflicting requirements (typically in the form of two boxes requiring the same location). So, the first lesson to take away from this example is when to use set-based techniques. (Contrast this problem, for example, to the type of problem described in Example 1.)

Next, this design group's approach to requirements fits the set-based model very closely, and also illustrates when this method of establishing requirements is important. As was described in Chapter 3, one form of set-based requirements are minimums/maximums. As illustrated in the example, this design group makes extensive use of such requirements for the environmental tolerances of the various black boxes. What is more important, however, is their motivation for using such requirements. To understand this motivation, one must again consider the nature of the design problem: These engineers are given locational preferences by other designers which are often in conflict. Each box, however, does have constraints regarding where it can be placed.

The subsystem installation engineers, therefore, need to know what the limits are regarding the placement of the boxes, so that they are free to make as many alternations in the locations of boxes as possible. If requirements were stated as point values, installation engineers would not have an indication of whether or not one location was necessarily better than another, only that it did not exceed a requirement. By stating the environmental tolerances as minimums or maximums, installation engineers can judge the quality of a location by how much margin remains between conditions at a location and the box's limits. Thus, this example clearly illustrates how minimum/maximum requirements can be used in set-based design practices.

The new computer tool being developed by the installation group also helps to demonstrate how set-based approaches can be used to delay design decisions. As noted earlier, an important goal of the tool is to enable installation engineers to do their job as late as possible, thereby giving the other design groups more time or reducing the total cycle time of the development effort. By using set-based approaches to requirements, the installation engineers give themselves some design freedom even though they are working very late in the design process. If very specific, point requirements were given for every box, the installation engineers would have little, if any, degree of flexibility in placing the boxes. Their set-based approach to requirements, however, provides the ability to relocate boxes -- within a given range of tolerances -- to facilitate changes late in the design process. By applying the computer tool, this flexibility can be used across a large number of variables, i.e., box locations, to facilitate a responsive and fast placement process.

Although this example focuses on a very narrow design problem, it does clearly illustrate several set-based techniques. Importantly, it reinforces the notion that such techniques are most appropriate for design problems with a large number of variables which are potentially in conflict.

### *6.10 Design Strategies: Point-based, with Hints of Sets*

In general, the examples presented in this chapter are dominated by point-based strategies of design. Interestingly, however, in every instance in which a point-based strategy might fail, set-based techniques are used to alleviate such problems. For instance, in Example 1, conceptual designers made use of conceptual robustness to limit downstream problems related to changes which are made to the design. Similarly, in Example 2, even though only one option was pursued at a time for each subsystem, engineers always kept records of other options in the event of downstream difficulties. Examples 3 and 4 then illustrated that some design methods actually do make use of the notion of sets, allowing engineers to consider a wide range of options before settling on a final solution.

What is missing from all of these examples, however, is SBCE's notion of independence. In none of the examples do any specialty groups spend significant amounts of time independently considering a design and then proposing sets of solutions to find the final answer. In this regard, no company's design strategy truly follows the SBCE model. However, lessons from the above examples will be incorporated into the model presented in Chapter 11.

# 7. Virtual Product Development Tools

## *7.1 Chapter Introduction*

As was noted, many companies are in the midst of reforming their product development processes. At many companies, this reform process involves the development of advanced product development tools. The following sections provide examples of two of those tools from two companies. The first example is based on a project which has already demonstrated significant elements of its software tools. The second example has demonstrated several of its elements, but is a bit larger in scope, and, therefore, a greater portion of it still remains under development.

## *7.2 Example 5: An Integrated Design and Analysis Package*

As noted, this first company seems to have progressed fairly far down the development process in their use of a new virtual product development tool. This computer-based tool has several components, and it appears the leverage offered by the tool is a direct result of the synergy between these components.

The first element of the tool is a database management system. A single, integrated database is established for each project or system under development. When a design or analysis engineer wants to work on the system, the tool automatically converts the system's dataset into the appropriate form for use with the engineer's discipline-specific tools. When the engineer has completed his work, the tool then re-converts this new data into the standard format, allowing other engineers to use the data.

This ability to translate between data formats allows engineers using different tools to be directly linked to one, common database. In the past, each tool required its own data format, and, therefore, its own database. The need to maintain these separate databases limited the ability of engineers to interact easily and in a timely fashion. Relatively minor differences, such as one analysis program using degrees Celsius while another used degree Fahrenheit, introduced time delays and opportunities for errors to enter into the system analysis. Working from a common database should eliminate most such problems.

Another element of this computer tool is an historical database of past system designs. This database is used by the tool to allow new designs to be analyzed by extrapolating from past systems' performance. This extrapolation process works fairly well, according to engineers familiar with the tool. They suggested that the tool works effectively when analyzing a system's overall performance, but that it fails to provide reliable data when analyzing boundary conditions or other details of a design.

The "center-piece" of the tool, according to engineers familiar with the project, is a mission simulator. This simulation program allows a small group of engineers to consider a large number of system designs and system trades in a very short period of time. The simulation begins with the designer developing a system-level design for a new vehicle. Using the historical database and system-level analysis tools, the simulator then projects the new design's performance. If the design is deficient in some respect relative to previous designs, the computer tool will recommend other designs to the engineer based on data from the historical database.

Engineers suggested that the tool is not quite "point and click," but that it is nearly so. For example, the design tool allows an engineer to hold several features of a design constant while allowing the engineer to then experiment with another element of the design. The simulation then allows the engineer to see the effects of these variations on the rest of the design.

Developers of the tool reported that during testing, this system has shown significant improvements over previous methods and tools. Results indicate that in the time that had been required to develop one to three designs, *thirty to forty* can now be considered. Not only are more designs being evaluated in a shorter period of time, however -- fewer engineers are required to develop these designs than in the past.

Engineers cited several advantages which were offered by this ability to consider so many designs so easily. The first was that engineers felt the quality of their designs would be improved. By investing more effort early in a program, engineers believe they will be able to reduce downstream problems encountered during the development of a design. One engineer stated that designers "want to avoid going down blind alleys." In the past, the limitations of design and analysis tools required engineers develop a design to a significant level of detail before good design analysis results could be developed. As was previously discussed, this meant that engineers typically had to take a best guess at the design and then move ahead. If later it was discovered that the design was deficient in some respect, a significant amount of work would have to be redone. Engineers hope that this new computer tool will limit such massive rework. Rather than having to pick a concept "blindly," the tool will help engineers conduct some preliminary analysis of new designs. In this way, they will have more and better information on which to base decisions as to which designs to continue to pursue.

Another advantage offered by the tool was the ability to examine common features between programs. This company, like others, is attempting to reduce the total development effort across several programs by designing common parts to use on several products (see Chapter 8). The capability of this tool to facilitate trades has greatly enhanced the company's ability to identify what elements of different products can be made common.

A final advantage to the tool discussed by engineers was its learning value. For example, a new engineer, recently entering the work force after leaving school, was given the opportunity to work with the tool to consider a variety of system designs. The new engineer reported that in the week he experimented with designs on the tool he had learned more about the design of that type of system than he had in several semesters of classes.

Engineers noted that this learning offers two advantages. First, designers will better understand the trade space of the particular design on which they working, improving the decisions they make about that design. Second, by learning more about system design on one program, engineers will be that much more experienced when they join another program. Their ability to make decisions relevant to this new project should , therefore, be that much more improved.

Overall, this company's preliminary experiences with its virtual development tool have shown the potential for the tool to improve communication between engineers, improve product quality by facilitating a more informed decision-making process, and improve engineer learning across projects. At the time of this writing, the company was completing the development of the tool and was about to apply it to a new program.

### *7.3 Discussion of Example 6*

This first example of a virtual development tool provides a good illustration of the application of the first core concept of SBCE -- sets -- in the aerospace environment. As was described, developers indicated that an important feature of the tool is its ability to allow fewer engineers to consider a greater number of design options in less time than was true in the past. Thus, in terms of SBCE, the purpose of the tool can be considered as enabling engineers to analyze sets of designs.

Importantly, the tool developers intuitively followed another SBCE concept related to the exploration of sets: the use of simple models to analyze designs. As was described in Chapter 3 tests should be simple and quick, just detailed enough to expose problems that are near the surface of the design problem. Rather than attempting to incorporate the most advanced and detailed analysis tools into the system, developers at this company used a simple analysis and modeling program. While this analysis is limited, it provides enough detail at the early stages of the design process. Design options which are clearly inferior will be revealed and eliminated, while those with potential can be retained, even if they will be eliminated by future, more detailed tests.

Finally, tests of this tool are beginning to confirm the learning potential associated with set-based techniques. Though still anecdotal, as was described, developers believe the learning effects that have been thusfar will continue once the system is in more widespread use. And, as they noted, these learning effects build across programs: what an engineer learns on one program, she will take with her to another, allowing that program -- its product and its other members -- to benefit from her increased experience.

## *7.4 Example 6: Using Computer Tools to Institute a Process-Based Design Method*

### 7.4.1 Introduction and Overview

One company's efforts to improve its design processes focus on the use of advanced computer tools and moving toward a process-based (rather than functionally- or IPT-based) process. Engineers characterized the development of these current plans as having evolved from a series of steps in integrated product design. The first step in this evolution took a "design-centric" approach, in which the company was organized along functional lines and communication was based on the "over-the-wall" model. More recently, companies have moved toward a "team-centric" approach based upon integrated product teams using cross-functional communication. The next step in this evolution, according to engineers at this firm, is a "process-centric" approach. While engineers are still organized in IPTs under this model, their communication is driven primarily by processes, rather than any reliance on a specific functional orientation. When asked what this process-view included, engineers answered "everything," from what the steps in the design process should be, to how parts should be procured, to how the vehicle should be manufactured and assembled.

The company's effort to improve its design process began in earnest in early 1995. Engineers specifically cited the support of their company's president as a key factor in helping the program succeed. The primary external motivation for the program was product affordability, so the program is intended to help consolidate best practices across the company's various divisions, with the goal of the system being fully implemented by the year 2000. The program hopes to achieve a 50% reduction in product development costs and cycle time, a 30% reduction in manufacturing and assembly costs and cycle time, and a 30% reduction in maintenance and support costs for fielded systems.

The group working to develop this system has identified several bottlenecks in the current design process. These bottlenecks were seen as major roadblocks to improving the process and were tightly wedded to the company's organization. Bottlenecks identified by the group included:

- unavailability of product and process information and difficulty in accessing "enterprise knowledge;"
- inability to reuse data (it is often easier for engineers to redo an analysis than to find old data);
- a loose association among engineering groups;
- too much time required to understand the impacts of design changes, too much time required to complete these changes, and an inability to control the changes themselves; and
- paper-based, "serial communication."

Specific outputs of the program are to include a variety of advanced, computer-based design tools. To aide in achieving a process-based approach to design, a simulation-based tool is under development. In addition, the company is attempting to implement a "virtual development environment." This environment is intended to integrate design and analysis tools and to consolidate the more than a dozen databases used at the company to one common, shared database. The company is also working to include as many commercial software products as possible in order to avoid becoming trapped in a closed architecture.

### 7.4.2 The Ideal Design Environment

Underlying these process improvement goals is a vision of how the company would like to be able to do design. Company documents suggest that an ideal design environment would be characterized by "concurrent communication" and "decomposition of iterative paths." In this environment, each engineering discipline "performs parametric analyses and then meets to determine where the optimal compromises are." These parallel parametric trades allow "each discipline process to be uncoupled from any iterative sequential path… thus providing a concurrent, integrated design space." To foster communication and design trades, preliminary analyses "will focus on the relative differences between configurations, rather than identifying the absolute answer." This change in focus will enable "early selection of the most promising configurations for further investigation and definition." These comparative studies will then be used to enable teams to "share sensitivities so that solutions can be optimized by considering several factors at a time."

One difficulty identified in implementing such goals is the realization that, as one engineer expressed, "integration thrives on the design cycle." Most often, engineers do not even realize that they will need to make changes in a design until systems are put together and integrated. For example, a structural engineer may be confident that he has designed the best bulkhead, only until a systems engineer informs him of the need to drill a hole through the structure for a wiring path. The goal, therefore, is to enable this communication earlier in the design process, thereby shortening the cycle of design.

To facilitate this goal, the aim of the software development effort is to allow engineers "to operate at the level of three dimensional objects." When an engineer needs to modify a design, he will simply need to make the necessary adjustments to the part's shape, and then the computer will automatically update the part's detailed features, dimensions, etc. In addition, the rapid ability of the computer to make such alternations will be used by engineers to vary design features to understand their impacts on affordability and performance. Ultimately, it is hoped that engineers will be able to rapidly develop "families of designs" which will allow them to assess the pros and cons of various aspects of a design concept.

### 7.4.3 Involving Downstream Activities

One important goal of the computer systems is to allow downstream design issues, such as manufacturability and supportability, to be addressed much earlier in the design process than is currently possible. For example, manufacturing engineers will be able to study how a certain part might be machined while the program is still in the conceptual design phase. Lessons learned from "making" these virtual parts will then be fed back to the design engineers to update and improve their design.

In addition to involving downstream engineers earlier in the process, their knowledge and skills are being embedded in "knowledge-based design advisors" integrated into design tools. These advisors will help to guide designers' decisions, and to ensure that design standards are used. In addition, the computer tools will note when standards are not used and will be able to assess the impact of such decisions on cost, performance, and cycle time.

As a consequence of this early involvement of downstream issues, engineers expect an increase in the number of changes that are made to a design early in its development. Since these changes are occurring early in the design cycle, however, there is still significant flexibility in the design, so that the cost per change is low. If such changes were made late in the design process, as has often been the case, the cost for the changes would be significantly higher.

### 7.4.4 Convergence Range and Decision Gates

In conjunction with the development of these design tools, the design process itself is being reformed. A major component of this reformation is the concept of a "convergence range," illustrated in Figure 23. Initially, a part design is developed, but only to a given level of detail, perhaps such that its dimensions are accurate to within 0.1 inches. Then, through the use of computer simulations, the part's details are refined to greater levels of fidelity, perhaps all the way to a dimensional accuracy of 0.001 inches. The design of the part is then varied at this very fine level of detail, allowing several alternatives to be considered in parallel. Analysis tools, which are coupled to the design tools, are then used to evaluate the designs, generating a dataset for the part. As the total vehicle concept progresses through conceptual design, increasing the level of detail in the concept, the number of part variations is reduced, until only one design remains.

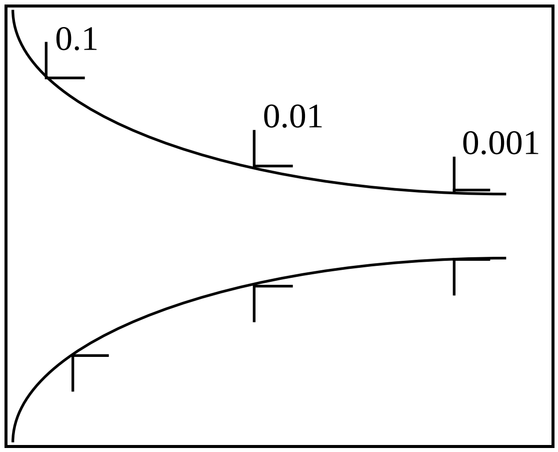



**Figure 23: Convergence range.** Initially, parts are only specified to 0.1 inches. Computer simulation is then used to refine the design to greater levels of fidelity.

An important issue in this convergence process is the development of the "gates" used to determine when to narrow the range of options being considered. Currently, the company is working on developing "globally agreed to convergence criteria." The intent is for these criteria to reflect a variety of perspectives, including engineering, operations, procurement, administration, data integrity, etc. These criteria then form the foundation used to judge where to place decision gates. Once data for a part or component are available to meet all of the convergence criteria, the part will progress to the next level of detail.

At the present time, the company has demonstrated this process on large components, such as bulkheads. Their goal, however, is to be able to conduct the required analysis, variation, and convergence on detail parts. The company sees this ability to consider a large number of detailed alternative part designs as a key element to enable the process decoupling described previously.

One engineer also described this process of considering design concept variations as critical to making good design decisions. As he put it, one "will never have enough information" to make a final decision, and it is also difficult to predict the downstream effects of a decision once it is made. The key, he suggested, is being able to understand the "sensitivities" of the design, i.e., how variations in one parameter will affect the total design. Consideration of these sensitivities therefore provides a guide to making decisions.

### 7.4.5 Database Management

Engineers emphasized that the key to implementing this approach to design has been database management. Historically, standards such as Initial Graphical Exchange Specification (IGES) have been used, and the company is now heavily involved in the development of the Standard for Exchange of Product Model Data (STEP). Since the convergence process relies heavily on generating and tracking a large amount of information for any given part, and then managing all such data for a number of parts, the effective use of database management systems has been a key aspect of the process development program.

As noted earlier, the company currently has a dozen database management programs, each of which supports a variety of computer aided design and analysis tools. These tools will now be connected via a product data manager, which will provide a common database for all of the computer tools. This increased ability to share the same data will eliminate the current efforts needed to re-define product data for each tool, thereby eliminating problems related to data translation and duplication such as inaccuracies, configuration control issues, and inefficiencies.

### 7.4.6 Pending Issues

Though the company has started to develop the exit criteria for the convergence gates discussed above, the process is still not perfected. Complicating their development is not only the varying levels of detail in analysis which some specialties can provide, but the critical nature of knowing some information before anything else. Structural loading data was given as an example. This information is critical to the entire design of the system's structure, and, therefore, must be available early in the design process.  Note that this problem is essentially the same as was described in Section 5.5.5.  Further complicating the issue is addressing the question of what happens if a design fails to meet all of the exit criteria for a given gate.  One engineer suggested that it will become essential to develop design options prior to entering a gate for a review to hedge against such failures.

While the approach to design described above is clearly the direction in which the company is moving, the engineers describing these changes emphasized the preliminary nature of their work: as noted, they expect to have the system fully functional by the year 2000.

### *7.5 Discussion of Example 6*

Even more than in Example 5, this example seems to be the first evidence of true set-based concurrent engineering thinking within the aerospace industry.  The company's model of its ideal design process, with its decoupling of iterative paths, parallel parametric studies, and sharing of sensitivities, is the only instance that was seen in this study in which engineers describe concepts similar to SBCE's notion of sets *and* independence.  As company documents described and discussions with engineers reinforced, the goal of the reformed design process is to allow different engineering specialties to consider a large number of concepts, and then compare these concepts to optimize a design.  This process is virtually identical to the ideal model proposed in Chapter 3.

Additional concepts presented in this example are also quite similar to notions of SBCE. The ideas behind the convergence rate and the decision gates, for example, match closely the ideas presented for the management of set-based processes.  In addition, the knowledge-based design advisors are similar to Toyota's manufacturing guidelines, and, therefore, are a means of providing constraints on the size of design sets.  As in the case of Toyota's guidelines, these computer-based assessment tools should help to ensure that design alternatives explored by different specialties do exceed the capabilities of other specialties.

Since this computer-based design system is still under development, one must cautious: the final form of the virtual development tool may not achieve the ideal environment which was described. It was informative to discover, however, that one company does see SBCE-type practices as potentially beneficial.

### *7.6 Conclusions: The Challenges in Implementing Virtual Product Development Tools*

It appears that given the complexity of aerospace systems, set-based practices are intimately tied to the development of advanced computer tools. The amount of information which must be generated to characterize even a small portion of an aerospace product rapidly begins to exceed to capabilities of the unaided human. In many regards, then, it is not surprising to see this link between set-based methods and computer tools.

Using computers to enable a virtual product development approach does not come problem free. One difficulty raised by several engineers was described as a Catch-22 associated with the power of computers. Though advanced computer aided design systems could be used to speed up the iterative cycle or to reduce the number of people needed to develop a design, they have instead been used to consider a greater number of designs in parallel. Though not a problem in and of itself, the ability to generate and analyze such a large set of concepts has made it difficult to know how many concepts to consider. As one engineer expressed, it is becoming even more difficult to know when it is "time to shoot the engineer and build the [system]."

Another problem associated with the computer analysis systems is that not all of the specialties can work to same level of detail. Structures, for example, has been a specialty which has benefited significantly from computer tools allowing numerous concepts to be developed rapidly. Controls, on the other hand, is still considered a "point-based" specialty, requiring a separate analysis for every design. Thus, while one specialty may be able to generate and analyze many concepts early in the design, the same may not be true for another. Coordinating these varying levels of design analysis fidelity poses a significant challenge to successfully implementing the process reforms.

As discussed by engineers at several companies, obtaining funds to develop advanced computer tools is also something of a Catch-22. Developers promise that the new systems will ultimately reduce the cost of design efforts, but initially a large amount of money is often needed to implement the computer tools. Managers who control the funds for such development naturally want proof of the system's capabilities before making large investments. To see such results, however, often requires that a significant amount of effort be devoted to developing the system. Thus, engineers ask for money to implement the system to save money, but managers first want proof of the savings. While the two companies cited in the above examples have overcome these challenges, engineers at several companies mentioned encountering such challenges.

Additional issues related to advanced computer tools will be addressed further in Chapter 11.

# 8. Standard Products and Practices

## *8.1 Chapter Introduction*

As was noted in Chapter 3, there is a close link between platform design approaches and set-based concurrent engineering. During this investigation, several examples were seen of platform strategies, and these examples highlighted the connection between SBCE and platform products. The following sections present these examples. Examples 7 and 8 illustrate companies which have modified their entire development practices to accommodate and facilitate platform strategies. The sections describing these examples, therefore, cover a range of topics related to the entire development process. Because of their similarities, a discussion of these examples is withheld until both have been presented. Examples 9 and 10, on the other hand, present much more limited instances of platform approaches. They are both instructive, however, for reasons not highlighted in Examples 7 and 8.

## *8.2 Example 7: A Company Beginning to Implement a Platform Strategy*

### 8.2.1 Background: Motivation; Basic Advantages and Disadvantages

Goals to reduce system cost and development cycle time have provided the impetus at one company to move toward standardized products and processes. The primary measure used by the company to track the implementation of standardization is non-recurring engineering costs. This measure directly reflects the amount of effort spent to engineer a part or component that can only be used for a specific program. Reusing designs between several programs means that engineering work completed for one system can be applied to another, thereby reducing the non-recurring engineering costs. The question then becomes, how can engineering work be most easily reused between programs?

The current answer is through product and process standardization. By standardizing a component's design, for example, engineers need only verify that the design will meet the operating conditions of the system in which it will be used, rather than having to completely design the component from a blank sheet of paper. Similarly, standardized processes mean that less time will be required to determine the best means for manufacturing a component -- simply select the standard process. By reducing the amount of effort required to generate a design, primarily by re-using past designs, significant savings can be achieved in both cost and time.

While such savings represent a significant advantage to standardized products, the drive toward standardization does come with a price. As one program manager described, "a common product is never an optimum product." Since a standardized design is sized to meet a wide range of conditions, components are often over-designed or under-designed for a particular application.

A follow-on trend, therefore, has been to develop products that are not only standardized, but that can also be tailored. Such products allow for two or three product parameters to be readily manipulated, usually in integer increments, such as adding or removing springs from a mechanism. While such tailoring improves a component's suitability for a given situation, the component will still not meet exact requirements; as one engineer pointed out, "tailoring makes it work, but doesn't make it pretty."

Aerospace products tend not to win contracts based on looks alone, however, and the advantages in time and cost offered by standardization have provided significant returns for the company. The following paragraphs present several "snap-shots" of various aspects of the company's efforts to standardize, highlighting the methods and approaches used to implement standardized products and processes.

### 8.2.2 Materials and Processes (M&P): Integrating Standard Materials and Manufacturing Processes into Design

The standardization effort within the Materials and Processes group (M&P) began with the development of a database characterizing all of the materials and processes used by the company. This database contains information on materials such as paints, adhesives, electronic materials, composites, metals, etc., and describes how the materials can be best used during manufacturing. The database also describes what operational or environmental constraints apply to a material or process.

When a new program begins, an M&P representative is sent to the program team. Based on the initial design studies for the system, the representative will develop a preliminary M&P selection list, which contains an account of what materials and processes might be applicable to this particular program. As the design develops, the representative will compare more specific needs of the system with this list. The representative will then recommend which material should be used for which aspect of the design, usually only making one recommendation for each component of the design (i.e., one material will be recommended for use on a structural support, not two or three options).

If a design team's requirements can not be filled by materials listed on the selection list, trade studies will be conducted to determine what material would fulfill the requirements. Once a material has been chosen, the M&P group will develop several prototypes (such as coupon tests or preliminary versions of system hardware) to qualify the material for the specific application. Once these qualification tests are completed, the material will be released for use on the program and will also be added to future selection lists so that it can be readily used again in the future.

Due to the evolution of material and process technology and the evolving needs of specific programs, the M&P group has established Parts, Materials, and Process Control Boards (PMPCBs). These Boards are forums for debating the development of new materials and processes, such as described above. PMPCBs also provide the oversight required to qualify a new material or process, and provide a mechanism for clearing materials for use on future spacecraft.

M&P engineers cited two primary advantages offered by this standardization approach. First, the selection lists provide an easy mechanism to provide design ideas to engineers working on a new program. Rather than having to start from scratch, engineers developing new designs can use the selection lists to see what materials have been used in the past and incorporate these materials into their work. Second, the lists provide data on materials and processes which have already been qualified for the environment in which they will be used. Many aerospace systems are required to operate in environments which far exceed conditions encountered by most consumer products, for example. Verifying that a material or process will in fact be able to survive in the required operating environment can be a costly and time consuming process. By maintaining a history of how a material has been used in the past, the selection lists allow a new program to bypass the time and expense of qualifying a material.

### 8.2.3 Electronics Packaging: Developing Evolving Standards

The electronics packaging design group provides a good example of how a standard product design can be initially established and then evolve to become more flexible and meet a greater variety of needs. This group is responsible for designing the boxes which protect and integrate mission electronics into a system. Their process of standardization has helped to reduce the cost of developing packaging concepts and has contributed to a design cycle time reduction of nearly fifty percent.

When a new system is being designed, a box designer first determines whether a standard design can be used. If the system is unique or has special requirements, the designer might be forced to develop a customized packaging concept. If such customization is not required, the designer will then select to use a standardized box design. The first version of this standard design came in fixed size increments. For example, boxes were available which could hold 5, 10, or 15 electronics cards. The obvious problem with such a standard design, however, was that systems would often require an intermediate number of cards, such as 3 or 7. In such cases, the box design would be larger and, therefore, heavier, than was necessary, since the box would have empty slots.

The next standard box design allowed for more flexibility. Since the 5 and 10 card boxes, for example, were essentially identical except for their lengths, the first evolution of the standard was to a design that could be “cut” to the hold exactly the required number of cards. The attachment points for the cards were set at a fixed interval, thereby allowing the box design to customized based on the number of cards required.

The next evolution of the standard is to break the box itself into smaller, standard elements. Rather than having a standard box, the next design will have standard endwalls, card attachment fixtures, and sidewalls. These standard elements can then be combined for a specific design, allowing not only the number of cards, but the card pitch (cards per unit length of box) to be customized for a given design. Since the design elements themselves are standardized, however, the design process will still be faster than if the box had to be designed from scratch, while at the same time achieving a greater level of customization than was possible with the original standard.

One program manager working to develop these standards also noted that new technologies can be incorporated directly into standard designs. In the case of the boxes, should a new composite material be developed with advantages over the current material, it could be integrated into the standard. The design itself would not change dramatically, but endwalls and supports could be made thinner, for example. Allowing for the inclusion of such developments further illustrates how standard designs can evolve over time.

### 8.2.4 Standard Designs and Decision Making

The evolution of standard product designs has impacting decision making during the design process in several ways. Two of the most interesting are the effect of standard products on when design decisions must be made, and how the existence of standard designs have affected the design decisions of various programs.

First, consider the impact of standard products on design decision timelines. While in many cases engineers emphasized the need to make decisions as early as possible, another group suggested that design standards have in fact allowed them to delay making design decisions. Several program managers described two ways in which this might happen. In the first case, typically true of small parts or mechanisms, the availability of standard designs means that detailed consideration of these parts can be delayed while more pressing or critical issues are addressed. In the past, such parts were often custom designed, requiring one or more engineers devoted to their design and analysis. In many cases, however, standard parts have now been developed which can be used with very little additional design or analysis work. Since the designs are already completed for these parts, no engineers need to be assigned to work on them until they are required late in detail design.

In the second case of delaying decisions, standard designs often allow for the "growth" of a system. Since standards are often developed around a scaleable architecture, engineers can initially choose to use the smallest equipment they think will be required. If, however, as the design progresses, engineers determine that their initial estimate was low, the next "size" can be used in the design. Two different examples illustrate this point.

In one instance, a system's structure was initially based around the smaller of two standardized designs. As details of several of the subsystems evolved, it became clear that the chosen structure would not accommodate these subsystems. Rather than having to start the design work from scratch again, however, engineers simply switched to the larger of the two common designs. While some interface issues were complicated to resolve, these problems were far less difficult than if the entire structure had to be redesigned.

In a second example, engineers sized the electronics for a control subsystem based on an initial estimate of the total system's performance. Again, as the system progressed into detail design, this estimate proved incorrect, requiring the addition of several more control mechanisms. In the past, the addition of these mechanisms would have required the control subsystem electronics to be redesigned. Standardized control subsystem electronics, however, allowed engineers to accommodate the growth in the control subsystem by simply adding more control cards, without having to redesign the entire subsystem.

Both of these examples illustrate how late changes to a design were accommodated through the existence of standard designs and products. Program managers stressed the value of this design flexibility offered by standardized product designs. In past programs, such changes would have resulted in a significant amount of design re-work. The availability of the next “size” in the standard products, however, limited this rework to less critical issues.

The existence of standard products has also had other, more direct effects on design decisions. Standard products are usually developed to accommodate some likely range of mission requirements. So long as a new system falls inside of that range, the system can make use of the standard products. That such an alignment will occur, however, is not necessarily assured.

For example, a program manager responsible for the development of a standard product described a program which had chosen a particular set of values for several mission parameters. These choices, however, pushed the system outside of the envelope for which a standard product was appropriate. The manager illustrated the situation with a sketch, reproduced in Figure 24. While the chosen values for the system were achievable with available technology, the design would have to be custom-developed, increasing the time and cost required to design the overall system.

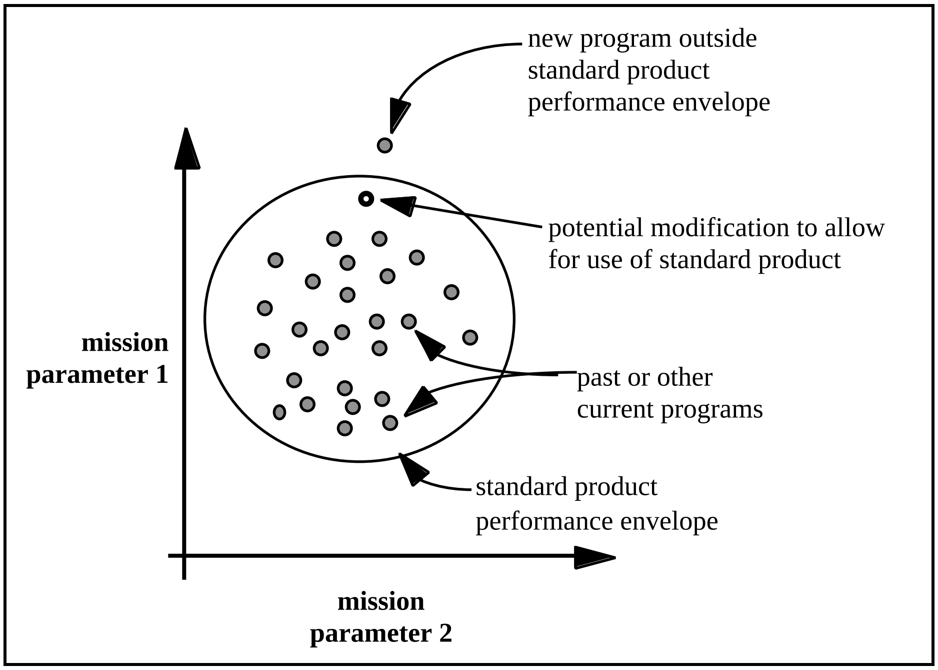



**Figure 24: Standard product performance envelope.** By modifying mission parameters, a program can take advantage of an existing standard product.

After meeting with design engineers on the program, the manager was able to convince them to modify their design such that the mission variables would fall within the realm of the standard product. Although making such a modification changed the system's performance, the system would still meet all of its requirements. In addition, it would fulfill these requirements at a lower cost than if customized products had been used.

This example highlights the potential influence which standard products can exert over a program's development. Because of their availability, standard products can be designed into a new system faster and at less cost than if new products were developed. As another program manager described, engineers do not select a particular bolt because that bolt is sized to exactly carry the needed load. The bolt is selected because it is available and cheap. Managers of standard product lines within the company have been portraying their products to development programs in a similar fashion. The product may not meet a design's initial requirements, but if these requirements are slightly modified, the standard part could be used. Applying the standard product then results in a significant savings in time and cost. At the moment, these benefits are beginning to outweigh the advantages of customized products.

### 8.2.5 Standardizing the Design Process

Several groups within the company have moved to not only standardize their products, but also their design processes.

For example, the electronics packaging group has generated a standard process for the design of their boxes. Though relatively general, this process flow defines the steps required to develop a box design, from the receipt of specifications through drawing release to manufacturing. The process flow not only identifies what must be done by the box design engineer, but also notes with whom this engineer must communicate to receive permission to continue to refine the design. Once a design concept has been finalized, for example, the design must be sent to the reliability, check, and analysis groups prior to the development of a detailed solids model of the packaging concept. The flow also clearly notes when iteration might occur in the design process. For instance, if in the example just described one of the groups has a concern about the design, the design might require modification. The potential for this iteration is clearly indicated in the design process flow.

Engineers in the packaging group and within other design groups stressed the importance of standardizing design methods. One program manager specifically cited the need for well-defined design practices, suggesting that standard methods "had to be used to be successful." He continued, saying that by following a structured process, engineers are less likely to overlook mistakes, and, therefore, are less likely to "have to go back and do things over."

One benefit of standard design processes cited by several engineers was its educational value. Program managers emphasized that having standard design methods simplified the training process for new engineers. These documented methods avoid telling new engineers, "do it long enough and you'll get a feel for it," and instead provide a concrete methodology in which new employees can be trained.

A similar benefit of standardized design processes was also described in terms of improving the dissemination of esoteric design knowledge. In one engineering group, a particular manufacturing process had proven difficult to understand, and, consequently, only a few engineers were very familiar with it. When these engineers would move on to other positions, however, they would take their knowledge with them. The development of a standardized design method for this process has since enabled the design group to rely far less on the skills or knowledge of any particular engineer. The standardized design methods now allow any engineer to be easily trained in the process, enabling new engineers to move up the learning curve far faster than was previously the case.

The electronics packaging group also illustrates the synergistic effects that can be achieved through standardizing both the design process and the design of a component itself. By ensuring that a predetermined number of reviews have been passed during the development of a box design, it is now possible to accelerate the process in its later stages. For example, if a standard box design is being applied to a new system, and the design has passed all of the needed reviews for this new application, manufacturing can begin to build the box prior to the formal release of the drawings. While the drawings go through the formal sign-off procedure required before they are "officially" released, manufacturing can already begin working on the first versions of the box. Revisions are incorporated into the drawings as needed as they receive the required signatures and are then formally released. In the past, all of the signatures would have been required *prior* to the start of manufacturing. Allowing for manufacturing to begin to build the box while these formal reviews are completed has contributed to the reduction in total design cycle time for the design of these boxes.

The push for standardization has also progressed into other elements of the design process, such as drawing notes. In the past, every engineer would write his own notes on a drawing, such as which dimensions were the most important, what tolerances applied where, etc. Though typically not problematic, the writing of these notes took away time that an engineer might instead use for design and also introduced the possibility for mistakes or confusion, particularly during the hand-off of the drawings from design to manufacturing.

To eliminate these problems completely, the design group has adopted standardized drawing notes. These are a numbered set of notes which are regularly used on the drawings generated by the group. When appropriate, the note is included. If a note does not apply to the drawing, the number is listed along with the label "NOT USED." Where needed, customized notes are still added, but since many designs are based on standard components, many notes are reused. Though a relatively simple measure to implement, standardized notes have increased engineers' time for design and reduced the possibility for miscommunication between other engineers and manufacturing.

Though standardized methods have lead to a significant number of advantages, engineers did cite some drawbacks. For example, in many cases, these methods are technology specific. If the technology involved in the component changes, the design method itself must change. In chip design, for instance, the size of the chips has been reduced over time, which has lead to an increase in chip complexity (since the functionality of a chip has either remained the same or has increased compared to past chips). As this evolution has occurred, design methods have had to be modified to handle the increased level of complexity. Whereas in the past only one engineer would have been needed to design a chip, six to seven are now required. This change has also forced a change in the way the chip themselves are developed. Previously, the chip was designed as one single entity. Due to the increases in complexity, however, chips are now partitioned into several blocks, one block then being assigned to an engineer. Clearly these changes have forced the design process itself to evolve.

Another potential drawback is that standard methods might eliminate some design options. Again, this consequence is directly related to the technology of the device being designed. Drawing another example from chip design, engineers described how many circuits can be designed for synchronous or asynchronous operations. In a synchronous design, commands are executed “to the beat of a clock,” while in an asynchronous design, commands are executed simply as needed. While neither design is necessarily “better” than the other, one design group has chosen to make all of its designs synchronous. Their standard design methods, therefore, are all built around the development of synchronous chips, thus limiting the possibility of developing asynchronous chips. While the design group still retains the knowledge to develop asynchronous chips, a program requesting such an effort must provide significant justification to convince the design group to deviate from its standard practices.

Despite these limitations, program managers who had implemented standardized design processes were confident that the advantages offered through standardization far outweighed their disadvantages. As one program manager explained, it is expensive and time-consuming to develop their systems, therefore they have been “driven to a process that would let [them] be successful the first time.” By building on and documenting the methods used to develop one successful design, several engineering groups have increased the quality and reliability of their designs, while at the same time reducing their costs and cycle time.

### 8.2.6 Standard Designs and Their Effects on Design Organizations

In addition to affecting the design process, standardized products have begun to influence changes to the organization of the company itself. A good example of these changes is provided by one of the mechanical design groups. This group has had responsibility for several elements of the systems built by the company, including the structure, mechanisms, and mission environment analysis. Each of these elements is then represented within the group by a smaller department. Historically, as a new development program began, engineers from each department would be assigned to the program and remain dedicated to the program for its entire length. The evolution of standard designs, however, has changed this process.

The mechanism design department, for example, has developed a wide array of standardized products. Currently, the department is working to make these standard products more customizable, such as spring mechanisms which can use more or fewer leaf springs to tailor a force for a given application. Such products have meant that the design department can serve several programs using the same design.

This increased breadth of applicability is now leading to new concepts for how this design department should be used in the company. Whereas in the past, mechanism design engineers were assigned to a specific *program*, they are now being assigned to a specific *product*. For instance, one engineer (or engineers) will become responsible for supplying all company programs with spring mechanisms. These product engineers will work to develop more customizable designs, while at the same time working with specific programs to ensure that the programs can best capitalize on existing standard products. Therefore, a major benefit of this product-aligned organization is that one engineer can now interface with several programs, rather than having one engineer per program. Engineers will be able to focus more directly on evolving their product designs, and will also be more aware of what needs are faced by multiple programs that could be met by common products. Engineers expect this organizational change to reinforce the other benefits of product standardization and consider these changes to be the next step in the evolution of their standardization effort.

## 8.3 Example 8: The Evolution of Standardized Products and Processes

### 8.3.1 The Development of a Gated Process

One company's recent success toward developing standard design methods provides a good illustration of what has occurred at several companies. Historically, most of the company's programs emphasized performance first, followed by schedule and cost. These priorities reflected the needs of the customers, who were typically most interested in performance. The company's design processes, therefore, were geared toward addressing these customer preferences. Schedule and cost constraints were often sacrificed in order to improve a system's performance. As one manager explained, engineers were "very prone to iterating continuously," because with each refinement, the design's performance would improve. A design concept would be generated, analyzed, and then redesigned based on this analysis to achieve ever greater levels of performance.

These practices, however, had several inherent problems. For example, each system under development consisted of numerous subsystems. Every subsystem, however, was continuously refining its design, so that at any given point in time, managers would have a difficult time assessing how close a subsystem was to completion. This fluidity in design also contributed to a fluidity in a product's requirements. Each time the design was modified, the requirements would also have to be adjusted. Ultimately, rather than the requirements being finalized prior to the design, both design work and requirements development would end at the same time. This simultaneity prevented a design concept from being directly compared to a firm list of requirements to verify that the design did in fact meet the customer's needs.

In addition, as one manager described, this ill-defined process lacked "any sense of closure." The process had no clear mechanism to determine when engineers should finalize a design and stop iterating -- there was no good definition of what made the design "better" as opposed to what would be good enough. Instead, the design would be refined until time or money ran out, but, as noted above, even these constraints were often relaxed to allow for performance improvements.

Over time, however, the company has significantly reformed its process. The pressure to make these reforms has come primarily from the customers, and largely from one requirement: lower costs. As previously described, the company's past practices largely disregarded cost and schedule issues relative to performance issues, an attitude which reflected the customer's desires as well. Even to the customer, higher performance was more important than remaining on time and under budget. The customers have changed in recent times, however, and so have their priorities. Customers now consider schedule and budget as important if not more important than performance. To stay responsive to customer needs and to remain competitive in the industry, the company has moved to adapt its processes to address these changed customer priorities.

The company's efforts began on the manufacturing floor. Every program in production was reviewed, and specific dates were assigned for the completion of each product and process. These dates were then communicated back to the design organizations. If the current design schedule failed to meet a date specified by manufacturing, the design organization had the responsibility to make changes to enable the program to conform to the schedule.

Once firm dates and schedules had been established between manufacturing and design, the company then began to establish timelines for the design process itself. The first step in this process was to thoroughly review and then understand what steps had to be accomplished to generate a product design. Important elements of this process included understanding how different groups, which were responsible for different elements of the system, depended upon one another for information and how the sequencing of this information impacted the development of the design.

Based on this data, the company established "process gates" to define a "gated process." The gates sequenced design tasks and were essentially a series of checklists which defined what design tasks had to be accomplished before the next series of tasks could be initiated. The gates also clearly defined who was involved in generating the data needed to exit the gate, as well as what organization was responsible for ensuring that the needed data had been collected and finalized. Thus, each gate was "owned" by one group within the company. For example, the first gate, which related to the proposal phase, was the responsibility of the business development manager for a new program. The intent of establishing such clearly defined process checkpoints and checkpoint ownership was to ensure that the right people were doing the right tasks, at the right time, and for the proper length of time. In addition, the gates also clearly defined which design activities could be completed in parallel, and which activities had be to executed serially.

Program managers emphasized several other important aspects of the gated process. One was the need to have clearly defined systems and subsystems. For example, a propulsion system might be defined by some engineers as the engine and the fuel system, while other engineers would only include the engine. Clarifying such definitions was essential in enabling the establishment of proper exit criteria at each gate.

In addition, common software tools or at least a common design database were critical to making the system successful. If design data had to be converted from one form to another as it passed through each gate, the process would become much more complex and would have increased the potential for errors. Such complexities can be eliminated, however, if different engineering groups use common design tools, or if their specialized tools can at least read data from the same database.

Finally, engineers emphasized the need to commit to keeping a gate closed once it had been passed. As noted earlier, past tendencies at the company had been to go back and refine a design whenever room for improvement was found. The gated process, on the other hand, stressed that once design decisions had been made and a gate closed, the decisions should only be changed if such changes were critical to meeting the customer's needs. This commitment to keeping a gate closed has helped to enforce the discipline required to bring the closure which had previously been lacking in the design process.

As acceptance of the standardized design approach has spread, the company has begun to enhance its coordination between programs. In the past, each program under design or manufacture had its own schedule. As the number of programs has expanded at the company, however, these independent schedules have lead to conflicts over resources, such as testing facilities. To address these issues, the company has implemented a company-wide scheduling system. This software tool enables the company to coordinate the schedules of all of the programs currently under development and manufacture, so that each program can then make realistic trades regarding its own schedule. For example, if a program wants to slip its schedule to allow for more design refinement, it might learn that such a slip would bring it into conflict with another program for use of a test facility. In the past, such a conflict might not be identified until both programs showed up at the facility at the same time. By coordinating the schedules of all programs, such conflicts can be eliminated.

### 8.3.2 Designing the Design Organization

Several program managers emphasized the importance of aligning the company's organization, processes, and products into a coherent system. This alignment occurs at several levels within the company. At the company-wide level, business units have been established with responsibility for the major elements of the products. One business unit is responsible for structures, another for electronic equipment, etc. These business units then assign specific engineers to address design issues for each program under development.

Within a program, the gated process helps to ensure that the organization of the program is properly matched to the product's development. The gates clearly identify who is responsible for what tasks and what data, thereby identifying what personnel should be working on the program at any given point in time. One program manager compared the evolution of a program's personnel to a lava lamp: The people involved in a program would not necessarily change, but the manner in which those people interacted might. As one group assumed responsibility for the next gate, their influence on the program would increase, for example. Once their work was completed, however, their influence might again decline. Engineers from the group might not, however, leave the program all together. As noted previously, the formalization of these changes provided by the gates helps to ensure that right people are involved in the program at the proper time.

### 8.3.3 Product Design

Along with the company's process and organization, its product has changed in recent times as well. Like other companies, this one has moved toward a product architecture built around standardized components. Standardized designs complement the gated process well, because the use of standard products limits the variability of the design process itself. For instance, one business unit has developed a standardized power amplifier. By adding or deleting elements on the circuit card, the output of the amplifier can be modified to match a specific program’s need.  This modification process is much more well-defined than if the amplifier had to be designed from a clean sheet of paper each time.

As with other companies, managers at this one saw both advantages and disadvantages to these standardized designs.  The largest disadvantage cited by engineers was that a product tended to be suboptimized for a given application.  Since standard components tend to come in integer increments, rarely does a standard product exactly match the requirements of a given program.

For this drawback, however, programs achieved several advantages.  The largest of these benefits are reduced delivery cycle times and lower costs.  As discussed above, these reductions are directly in line with the current demands of customers.  In addition, the standard parts tend to have a “design heritage,” i.e., they have been successfully used in various operational environments.  Such components, therefore, require less testing for a specific program and reduce the program’s risk.

Standardized products also allow for quick and easy trades to be considered for a system. The scaleable power amplifier described above, for example, enables designers to quickly consider several antenna designs for a particular application.  Since the amplifier can be readily modified over a fairly significant range of output levels, antenna engineers can reduce the size of antenna and use a higher output level from the amplifier without incurring any significant penalties in weight or cost.

Finally, as has occurred elsewhere, design groups at this company have been working to constantly improve their standard components.  As one example, engineers described how an electronics package that had weighed six pounds has been reduced to size of a pocket calculator. Such improvements in weight and size have allowed for the total product to become smaller and lighter or have allowed other elements of the system to become larger without increasing the total product’s size.

Establishing standard designs have allowed such advances to evolve off-line from any one program.  Since the components are needed by virtually every product, they can be developed on a path separate from a particular system, so long as the interfaces between the final product and the standard component do not change or are tracked carefully. Again, such practices have reduced the cost, cycle time, and development risk of any specific project, and have still allowed for significant advances in product technology and performance.

### 8.3.4 Taking Advantage of the World Wide Web

As different design organizations within the company have begun to standardize their processes, they have also begun to invest heavily in web-based process pages. Established on the company's intranet, each design group has begun to document their standardized processes on group home pages. The basic goal of developing these web pages has been to enable the sharing of standard practices both within and between design organizations.

A subassembly design process web page highlights many of the features under development throughout the company. The main page at the site contains an overview of the design process, highlighting steps such as Alignment, Convergence, Brassboard, Engineering Modeling and Analysis, and Manufacturing Readiness. Under each process step, users can select a variety of pages, each of which documents an aspect of the standardized design process.

Clicking on Alignment, for instance, reveals examples of flow diagrams and parts checklists from past programs. These documents can be used by new engineers to learn how to document such aspects of their designs. A standard process list describes the capabilities and limits of manufacturing processes, documents what information a designer must provide to manufacturing to allow for the use of a particular process, and has a form to enable the engineer to request the development of a new process. Under Convergence, engineers can call up standardized drawing forms and matrix templates which are used to help facilitate discussions between different specialty groups. By standardizing the forms used for such conversations, the company has moved toward a more common frame of reference and language between specialties, improving the communication between them. Other pages under the other process steps contain additional examples and forms which have been standardized for the design process.

An important standardized form on the web pages is the design notebook. Clicking on the link for the notebook allows an engineer to download a standardized document template to his desktop computer or workstation. The notebook is intended to be used to track the design and development of a product -- to provide a history of the product's evolution. Providing engineers with a standardized template for such documentation increases the likelihood that they will use it. At the same time, the standardized form will enable the next step in the development of the web pages: placing past design notebooks on the web. The intent is to allow a new engineer to view the notebooks of past programs to provide access to lessons learned on past programs. Use of the notebooks already helps current programs when they encounter difficulties in the design process. By having documented their decisions, engineers can more easily review their past work to see where changes can be made. When past notebooks are placed on the web, future programs will be able to avoid pursuing dead-end design paths in the first place.

### 8.3.5 Influencing Requirements

The development of these standard components and practices has also impacted the manner in which the company attempts to influence its customer's requirements. In the past, the company would accept the customer's requirements and design a system which exactly met those requirements, regardless of cost or schedule. As customers have become more sensitive to such factors, however, the company has been working to modify their relationships with customers early in program. Rather than taking a customer's requirements as cast in stone, program managers are now returning to their customers and offering a choice of options. On one hand, they will offer to build exactly what the customer requested. Moving down such a path, however, often means that standard products can not be used to a very large extent. In order to provide the customer with a product in less time and at a lower cost, they will also offer to develop a system which capitalizes on standard products to meet most of the customer's requirements, but not all of them. Under many circumstances, customers have been willing to back off on several requirements to take advantage of the faster cycle times and reduced costs offered by a standardized product.

Engineers have emphasized that such communication with the customer and influence over system-level requirements have been critical to realizing the benefits of standard components. If every customer's specific requirements were met to the letter, much of the progress which had been made in standardizing components would be worthless -- each product would have to be custom-built at every level of design. By working with the customer, however, the company has been able to reduce the cost and delivery time of its systems, and still satisfy its customers' most important needs. Note that as reforms in this customer-producer relationship take hold, the company will have successfully changed its entire development process, which began on the manufacturing floor and has progressed back into the product design process, the product's architecture, and, finally, the relationship with the customer.

### *8.4 Discussion of Examples 7 and 8*

Examples 7 and 8 both illustrate companies which are pursuing platform product strategies. Their design processes are typically more point-based than set-based, but as was noted in Section 3.6, there are important connections between platform design and SBCE.

As shown earlier in Figure 24, platform design clearly has a set-based character: the needs of a set of programs are compared in order to develop a single product which can be used by all of the programs. Looking for such regions of overlap across programs is similar in many respects to the logic of SBCE. Set-based concurrent engineering, however, looks across the needs of several functional groups *within a program,* rather than across multiple programs. Thus the philosophies are similar, but the execution is quite different.

Other features of the Examples 7 and 8 illustrate additional set-based concepts. The M&P selection lists described in Example 7, for example, are similar to Toyota's design guidelines. These selection lists provide a good illustration of the independence concepts of SBCE. As was described, when a material and process engineer is assigned to a program, he first reviews that program's requirements. He then returns to the company's material and process database, and uses the requirements to select an initial list of materials and processes for the program. In effect, the M&P engineer is independently exploring the design space for the program. This initial list is then narrowed as the design progresses and more details are revealed. In many respects, therefore, this process is a good example of SBCE. Where it falls short, however, is in the limited number of options that are shared between the M&P engineer and other members of the team. As was discussed, typically the M&P engineer only recommends one material for a given application, although she may have considered several. If the engineer were to share more options with the other engineers on the team the process would truly fit the SBCE model.

Example 8 also demonstrated another concept similar to those in SBCE: once the design space has been narrowed, avoid expanding it again. As was described, once a gate is closed in the design process, engineers at the company described in Example 8 are instructed not to attempt to improve the design later in the process unless absolutely necessarily. This instruction is akin to SBCE's notion of staying within a set once committed to it.

In summary then, both Examples 7 and 8 provide good illustrations of the advantages and disadvantages of platform-based design strategies. In addition to the issues raised in this discussion, several other features of these examples will again be explored in Chapter 11.

### *8.5 Example 9: Design Re-Use*

In an attempt to reduce costs and cycle time, one company has embarked upon a major effort at design re-use. The basic aim of this effort is to allow design work completed for one project to be used on another, and in many regards is similar to efforts at product and process standardization. One engineer illustrated this process and compared it to past processes in a sketch which has been reproduced in Figure 25. As described by the engineer, in the past, programs were considered one at a time, in isolation from one another. Each program, therefore, essentially required that all elements of a system be designed from scratch. The basic problem with this approach, however, was that other projects may have already developed systems or components which would be applicable to a new design. Similarly, since new programs were self-contained entities, each program was developing its own system elements, even if all of these programs might have started at the same time and might have shared some design features. In either case, a significant amount of design work was duplicated between the programs.

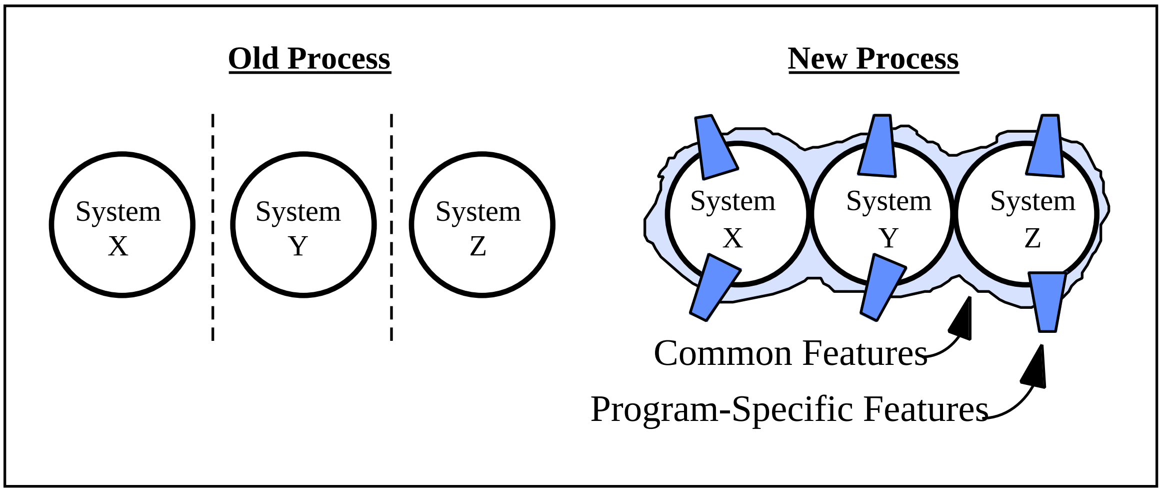



**Figure 25: Moving to design re-use.** In the old process, Programs X, Y, and Z were each considered in isolation. In the new process, the three programs are considered together, so that common features across the programs may be identified.

In the company's new approach to design, several programs will be considered simultaneously. This simultaneous consideration allows engineers to look for elements of overlap in the designs. Though each program will require design work which is specific to that system, the total design effort across all of the programs will have been reduced.

Engineers described a multi-step process which facilitates this type of design re-use. First, the requirements for several programs are reviewed. These reviews allow engineers to understand the basic performance needs of each system. The requirements for the programs are then compared, establishing areas of overlap in the designs. Design engineers are then allowed to develop baseline components which can be used by all of the programs. These baseline designs are then passed onto each program, which in turn develop derivatives of the common designs to meet the specific needs of their particular program. This process has reduced the total amount of design work required across the programs and has consolidated similar design efforts across several product lines.

This approach to design is also facilitating a major reform effort in the company's manufacturing strategy. In the past, each new product developed by the company was manufactured in a new factory built specifically for that program. Moving to a design process which emphasizes commonality across programs has enabled the company to stop this practice. The company is currently in the process of developing a common production facility. Several projects, all of which share some design features, are to be produced in this single facility. As in the case of the design re-use effort, this change in approach to manufacturing is expected to reduce system costs and cycle time.

### *8.6 Discussion of Example 9*

Though similar to the strategies of standardization presented in Examples 7 and 8, this company's approach is slightly different. In the earlier examples, the emphasis was placed on modifying a new design such that it could make use of old component designs. In this example, however, a slightly different pattern is revealed.

Rather than forcing a new system to simply make use of an old product, this approach compares several new systems and then develops a new product which can be shared between them. Furthermore, each system does share exactly the same product -- they share only those features which are clearly common between them.

Perhaps what is most interesting is the design process used to generate these shared elements. One design group is charged with the task of developing the shared baseline components of the product, while engineers assigned to each system are then allowed to modify and complete the design to meet their specific needs. This approach is interesting in that it suggests something of a narrowing process. A baseline design which is widely applicable is first developed. Further engineering of the design adds more features and details, and gradually tailors it to a specific application, i.e., *narrows* the applicability of the final design. In addition, this narrowing process is accomplished by different groups of engineers. One group develops a baseline design, while subsequent, downstream groups refine the design. This progressive narrowing is a notion which will be seen again in the model presented in Chapter 11.

### 8.7 Summarizing Standardized Products and Processes

Because a company makes use of standardized products does not mean that it also uses SBCE. However, companies which follow such strategies clearly make use of several set-based practices.

# 9. Set-Based Approaches and the Influence of Cost

## *9.1 Chapter Introduction*

Many of the previous examples described the impact of cost on the design process. Cost has motivated many companies efforts in design process reform, from the development of virtual product design tools to the implementation of standardized products and processes. As the following sections illustrate, however, cost has also lead some programs to consider *more* design alternatives during a development program. Two examples are presented, followed by discussion.

## *9.2 Example 10: The Cost of Considering Alternatives*

One design manager provided a vivid example of the need, benefits, and justification for carrying multiple design options in parallel. Many of his programs are significantly constrained by time. These time constraints limit the ability of a given number of engineers to consider design alternatives. They simply do not have enough time to consider one concept and then consider another.

In order to evaluate more than one design option, therefore, the manager increases the size of his engineering staff for a project. These additional engineers can then work in parallel with the initial team, evaluating an alternative design concept. Bringing these additional engineers onto the program, however, increases the program's cost, since the hours spent by these engineers will now be charged to the program. It was necessary to bring these engineers into the program to meet the schedule constraints, but the program also faces tight budgetary constraints. How, then, can the extra expense of these engineers be justified?

The answer, in many of the experiences of this manager, is directly related to the cost of the system delivered by the program. As shown in Figure 26, four engineers were initially assigned to the project. When a second design option became a possibility, two additional engineers were added. These additional engineers were with the program for about four months, until the team was required to select a preferred concept. The extra cost associated with the additional engineers is represented by the shaded region in Figure 26.

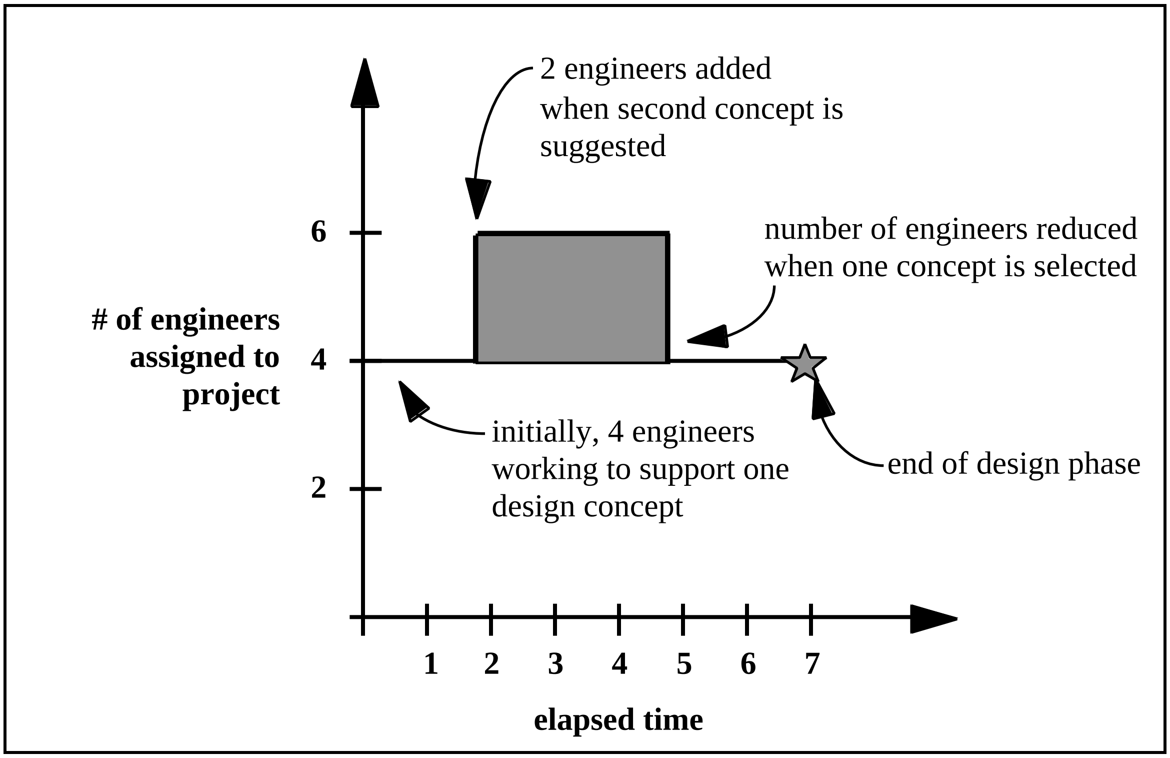



**Figure 26: The cost of carrying multiple designs in parallel.** When multiplied by the monthly salary of the engineers, the shaded area represents the approximate cost of carrying a second design in parallel with the first.

As explained by the design manager, this extra expense can be justified in one of two ways. The first is in the event that the initial concept proves unfeasible. By carrying the second option, the program's schedule can still be met. In addition, although one design proved unfeasible, exploring that region of the design space has educated the engineers on the program, and they will know not pursue similar paths in the future.

The second means of justifying the expense is directly related to the alternative design. If this alternative design can reduce the system's cost by an amount equal to or greater than the cost of the additional engineers, then the investment was worthwhile. By spending more money to evaluate an alternative concept, the schedule deadlines will still be met, while a lower cost system will be delivered. This second case is the one which the manager has encountered more often. Faced with program budget and time constraints, he has used extra engineers to develop designs that would reduce the operational system's cost, thereby recovering the cost of the additional engineers and improving the system's design relative to its expected cost. Thus, the need to remain within a constrained schedule while at the same time meeting a fixed program budget forced the manager to consider multiple design options in parallel.

## *9.3 Example 11: The Impact of Cost as an Independent Variable*

### 9.3.1 Introduction

Cost as an Independent Variable (CAIV) is a Department of Defense initiative aimed at lowering and controlling the costs of weapon systems. As described by one company, the DOD has not issued detailed directives to contractors, but has instead allowed companies to develop and then demonstrate their own methods for implementing CAIV. The methods used by one company illustrate how this new emphasis on cost has affected the design and development of aerospace products.

Traditionally, the primary focus of engineering trade studies was to identify the design alternative with the best performance. In such studies, cost was usually only a "peripheral issue." This company's approach to CAIV, however, has altered the nature of these trade studies. As described in a company document, rather than only considering performance issues, two additional principles must be obeyed during all studies:

"1) Cost is imposed in every trade study as an absolute constraint. If a design option increase the cost of the baseline design, the option is rejected.

2) Continuous trade studies are conducted during the entire design cycle for the sole purpose of identifying means of cutting costs without adversely affecting performance and schedule."

In addition, design engineers have received additional training related to cost issues, so that they might be able to implement these principles directly. Their approaches, described in the following paragraphs, clearly illustrate how an emphasis on cost has forced companies to consider larger portions of a product design space than was necessarily true in the past.

### 9.3.2 Implementation

Engineers described a five-step process which they have used to implement Cost as an Independent Variable trades. First, the minimum needs of the warfighter were established. These minimums defined the lowest level of acceptable performance, thereby constraining the design space at one end, while allowing freedom in another direction. The second step of the process was to quantify cost-performance trades, i.e., explore the design space. The goal of this step was to establish a "CAIV curve," showing cost along the horizontal axis and performance along the vertical axis (Figure 27).

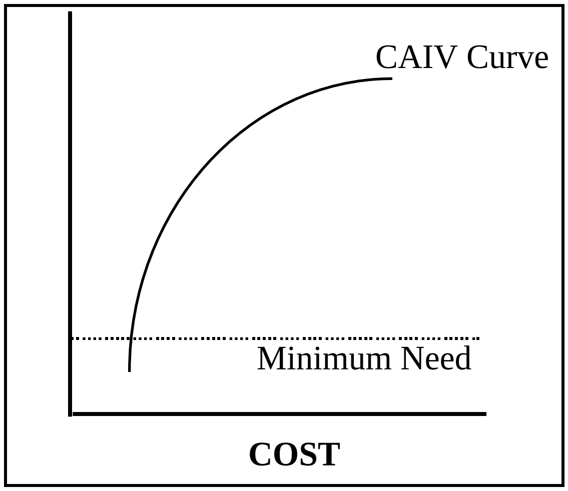



**Figure 27: A CAIV curve.** The minimum needs of the warfighter, i.e., the customer, are first indicated on the graph. The CAIV curve is then drawn to illustrate various levels of performance and the cost associated with each performance level.

Once the curve itself was defined, the next step was to set “aggressive and reasonable” cost goals. The intent of these goals was to define some level of performance greater than the minimum need, at a cost that would be affordable to the customer. After the goals were established, the next step in the process was to manage program risk such that the cost goals were met during development. Engineers noted that the development process might result in decreasing the system’s performance in order to meet cost goals. Finally, once the system was fielded, efforts were made to continue to reduce the system’s cost.

Engineers demonstrated the utility of the CAIV curve in analyzing design trades. Figure 28 illustrates the basic relationships which can be depicted on the plane -- more capable (at higher cost), less capable (at less cost), less risk (less performance at greater cost), and higher risk (more performance at less cost). These basic relations can then be used to help formulate design trade strategies.

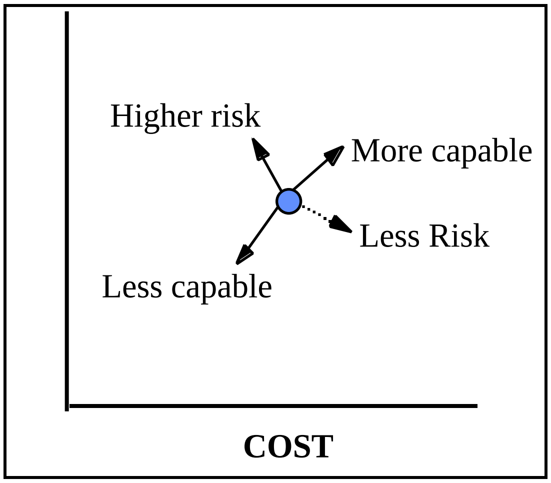



**Figure 28: Categories of cost performance.** The Cost-Performance plane can be used to represent a variety of potential trades.

This process is illustrated in Figure 29. In Figure 29, design options A through F are compared to a baseline design (indicated by the diamond on the curve). As can be seen in the figure, the curve suggests how trades can be made. For instance, design feature C has a greater slope than design feature B, meaning that C has a greater performance-to-cost ratio than B. For a given dollar amount, therefore, the design would benefit more when feature C is added than when feature B is added. Similarly, to reduce the system's cost, feature D should be removed first, since it's removal will decrease costs to a larger extent than it will decrease performance. The removal of feature F, on the other hand, would allow some cost savings, but such a decision would also result in a significant decrease in performance.

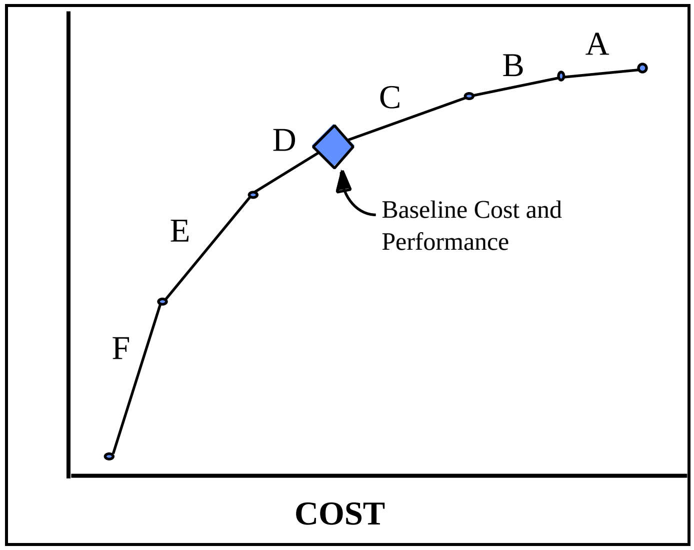



**Figure 29: Using a CAIV curve to guide design trade strategies.** Each segment on the curve indicates how technologies could be added or removed from a design to increase or decrease the design's performance and cost.

Engineers described two possible approaches to design to cost (DTC), which are central to implementing such CAIV strategies in "daily life." The first possible DTC method is to design a system, analyze it in terms of cost and performance, and, if the system costs too much, redesign it. A second approach is to allocate the costs throughout all of the systems on the vehicle, design a system, and then reallocate the costs as needed based on cost and performance trades. The company has been following this second approach, allocating costs via integrated product teams (IPTs). Each team, which is responsible for a given element of the system, is allotted a given amount of "cost" which they must remain below. If they exceed this value, the cost limits of other IPTs might be reduced to open up more "cost space" for the IPT which requires it. If such a reallocation is not feasible or practical, the IPT might then be forced to redesign its system element to reduce the cost, potentially sacrificing performance to do so.

While CAIV curves have been useful in showing which areas of the design space have greater "leverage" for improving performance-cost ratios, they are not intended to make decisions for an IPT. One engineer suggested that the greatest utility was in using the curves to begin discussions and negotiations over design decisions, rather than using the results of the curves alone to drive the decisions.

### 9.4 Discussion of Examples 10 and 11

Examples 10 and 11 both demonstrate several interesting effects. First, consider the similar consequence which they both illustrate: an emphasis on cost can force a program to consider more options. The manager who described the process described in Example 10 presented the example in terms of two alternative technologies; Example 11, taken from a larger program, shows how the CAIV curve is used to consider a wide range of possible design options. In each case, the motivation for presenting the choices was to enable the designer and the customer to select the most cost effective solution. Rather than considering just one design at a given cost, engineers in both cases instead reviewed a number of possibilities.

These various designs presented differing levels of performance and, more importantly to the customer, differing costs. In essence, both examples lead to the creation of sets of designs, each set consisting of a group of designs with various levels of performance and various costs. By comparing the options in the sets, the customers and the designers were able to meet the minimum requirements while doing so at the lowest possible cost. Consideration of only one option might have sufficed to meet the customer's requirements, but in both cases would have done so at a greater cost. Therefore, despite the increase in cost associated with considering a number of design alternatives, such an approach may minimize the overall cost of a program by allowing for the discovery of a lower cost final product design.

Example 11 also demonstrated several additional set-based principles, most notably the use of the minimum constraints and the use of goals. Rather than establishing a specific value for the customer's performance requirements, the CAIV methods illustrated above instead focused on defining the customer's minimum needs. So long as these most basic levels of performance were satisfied, any design would be acceptable. This flexibility then allowed engineers to explore a range of design possibilities, which were ultimately captured on the CAIV curve. Both the designers and the customer were then able to make informed judgments as to which technologies to include, based on how much performance was gained relative to the minimum needs and at what cost.

The use of the minimum constraint on performance was then supported by the use of goals for the costs of specific systems. As described, the goals allowed for some flexibility, enabling designers from one system to “trade” with another system to increase its cost limit. Together with the minimum constraints, these goals helped to facilitate a design process which was flexible yet which ensured that the customer’s most important requirement -- cost -- was satisfied. Both techniques are also good examples of set-based methods, enabling engineers to explore multiple design options while always keeping the customer’s needs in mind.

# 10. An Industry Perspective: Constraints, the Assessment, and a Summary

## 10.1 Chapter Introduction

The previous chapters have presented specific examples of design practices throughout the industry. Some of these practices have matched the set-based model closely, while others were more akin to the point-based model. This chapter summarizes the results presented in the preceding five. First, constraints on aerospace industry design processes are summarized, followed by an overall assessment of the set-basedness of the aerospace industry. The chapter then concludes with a summary of the lessons learned from the previous eleven examples.

## 10.2 Constraints on the Design Process

### 10.2.1 Suppliers

One issue touched upon in Section 5.5.6 was raised by engineers at many companies as a significant constraint in their design processes: the lead time associated with parts that must be purchased from suppliers. Because of these lead times, designs must be finalized as early as possible, so that orders can be placed with suppliers. The later design decisions are made, the later these orders are placed, and the longer the development effort will take.

For several companies, these lead times are complicated by the fact that the aerospace company may not be the supplier's largest customer. Rather than being able to place large orders with the supplier, and, therefore, have a significant amount of influence over the supplier's delivery schedule, some aerospace companies are forced to "wait in line." Though companies have worked to establish long-term relationships with suppliers, and, as one manger expressed, communicate with them "incessantly," the reality is that these companies can not place orders large enough to "drive" the decision making of some suppliers.

Companies have tried several tactics to work with suppliers to overcome these limitations. A good example of these efforts is one design group's program to qualify families of parts prior to their being needed by a particular project.

As explained by a manager in this group, many components purchased from suppliers are the types of parts that will be used repeatedly on a variety of systems. In the past, managers had waited to order these parts until a specific program required them. The problem with doing so recently, however, has been the pressure to reduce cycle time. Waiting to order the parts until a program specifically requests them means that the program will be delayed while waiting for the parts to be delivered. The issue is further complicated because many of the parts must be qualified specifically for the aerospace environment. Few facilities exist in the country which can be used for such qualification testing, potentially adding six months or more to the lead time for a part.

To overcome these problems, this design group has begun to qualify "technology families" prior to having a part requested by a specific program. The design group is in constant communication with its suppliers. When a supplier develops a new component that is likely to be used by future programs, the group will order several advanced examples of the part. Suppliers typically develop several similar versions of a part, thus leading to the manager's term of technology families. The design group will then qualify the various part versions, giving the data back to the supplier as part of the agreement for being given advanced access to the parts. When a new program then requires the part, the qualification process can be bypassed, shortening the lead time for the part.

Another possible way to shorten the lead time for parts in the face of little authority over suppliers is to carry inventory. Several divisions at various companies have begun this practice, but they all agree it will likely be short-lived. By carrying inventory, parts which are likely to be used by future programs can be ordered in advanced. The parts can be delivered, inspected, and cleared for use prior to any program needing them.

While this practice can reduce the lead time associated with a part, easing pressures to make design decisions, inventory comes with a cost. Many design groups are under severe pressure to limit costs, and, therefore, are pressured to reduce any inventory which they might carry. If the inventory is reduced for key components, however, cycle time for development projects will increase due to increased lead times. While the resolution of these issues is beyond the scope of this research, the effects of such dilemmas on design decision making are important to recognize.

### 10.2.2 The Limits of Parametric Models

Engineers at several companies described some of the difficulties they have encountered in conducting large numbers of trade studies supporting conceptual design efforts. One issue discussed at length was the difficulty in moving in the design space defined for a parametric study. A parametric relationship might exist from historical data on how different engine sizes affect an aircraft's maneuverability and acceleration, for example. Based on this historical data, it is relatively easy to develop an initial point for conducting the study. In addition, small changes from that point can also be fairly well predicted. Major deviations from the initial point, however, may result in performance changes that can not be simply extrapolated from the historical data. While significant effort has been devoted to quantifying these changes by resorting to basic physics, conducting large numbers of trade studies have nonetheless proven challenging to several programs.

These inherent limits to parametric models ultimately limit the number of possible design options which can be considered. Analysis programs which rely on basic physics -- such as computational fluid dynamics codes or finite element models -- typically require a large effort to develop the detailed product models needed to use the program. Such tools are, therefore, expensive and time consuming, and, consequently, are not well suited to considering a large number of designs which might be discarded.

Some of the tools currently under development, such as the ones presented in Examples 5 and 6 may overcome many of these limitations. Since the tools are still under development, however, such claims cannot be made with absolute certainty.

### 10.2.3 Communicating with the Customer

Engineers working on an airborne weapons platform described an interesting quandry in their relations with the customer concerning the use of estimates for system performance and features. While not a constraint *per se,* the story does illustrate that idealized design models might encounter unexpected difficulties in real life.

Their example was based upon early weight estimates for the aircraft. At one point, the contractor had estimated the weight of the airframe to within 1000 pounds. The customer then requested that an additional 300 pounds of equipment be added to the vehicle. Since the weight estimate was only accurate to 1000 pounds, said the customer, even adding the 300 pounds of material might still mean the vehicle was 700 pounds *below* the required weight. The contractor, on the other hand, also recognized the possibility of the aircraft ultimately being 1300 pounds *over* the specified weight.

The use of preliminary estimates is clearly a set-based practice. At the same time, however, by stating to the customer that the weight values were only estimates, the manufacturer opened itself up to additional requirements from the customer. The lesson from this example seems clear: improving the design process not only means reforming how the designer operates, but also how the customer operates. If a company were to attempt to implement SBCE without educating the customer, problems such as the one just described would likely occur with alarming frequency. Therefore, it is important that not only the designer but that the customer also have a firm understanding of a new design approach and how to behave given that approach.

## *10.3 How Set-Based is the Aerospace Industry?*

### 10.3.1 Three Answers

As was presented in the introduction to this thesis, one of the primary aims of this thesis was to assess the set-basedness of the aerospace industry, i.e., to determine if and to what extent aerospace firms already practiced set-based methods. Perhaps not surprisingly, there is no simple answer to this critical question. Instead, three answers should be given.

The simplest of the three answers is based on the definition. As was stated in Section 3.7, SBCE consists of two primary concepts:

1. Consider a large number of design alternatives, i.e., *sets of designs*; and
2. Allow specialists to consider a design from their own perspective, using the overlap (intersection) between individual sets to optimize a design.

Using this strict definition, the answer to the question, "does the aerospace industry practice SBCE?" is, "*No, the aerospace industry does not practice SBCE.*" The primary reason for this negative result is that very few industry practices followed the second principle of SBCE, that of allowing specialists to consider designs from their own perspectives. Further, while many aerospace design teams would consider multiple options, they would not carry the options for an extended period of time, and would instead select the "best" option as quickly as possible. Though a "set" may have been developed, its use was much more closely related to the point-based model than the set-based. Therefore, based on the obversations made during this research, the aerospace industry does not currently practice SBCE.

The second answer, however, is, "*Yes, the aerospace industry does practice some set-based techniques.*" While this answer is in the affirmative, note that it is a qualified yes: Many of the examples presented in the preceding chapters illustrate set-based methods, but none of the examples coherently demonstrated both of the basic SBCE principles. Techniques such as conceptual robustness, prolonged consideration of multiple options, minimum or maximum constraints, and delayed decision making were described by engineers at almost every company which was visited. In several of these cases, these practices were an integral part of well-developed, systematic design methods. But again, no site consistently demonstrated *both* primary principles of SBCE.

Finally, the third answer is also a qualified yes: *"At the level of interpersonal communication between engineers, the aerospace industry does practice set-based techniques."* This result was described in Chapter 5. Engineers stated that when they talked to other specialists, the nature of the conversation would often fit the definition of SBCE. Each engineer would present a set of possible options and would explain the advantages of these options from his or her own perspective. The engineers would then work to arrive at a solution which met both of their needs. Because each engineer presented a set of options, and because this set was based upon each engineer's independent assessment of the problem, such a dialog does meet the definition of SBCE. As noted earlier, however, the extent and frequency of such conversations varied dramatically between companies. Furthermore, where such discussions did occur, they were often related to very specific issues. Though a given conversation may have been SBCE-like, the overall pattern of development usually fit a more point-based model.

### 10.3.2 Implications of the Answers

The fact that at the present time no company practices SBCE in its ideal form does not necessarily invalidate the theory. As Ward *et. al.* noted, Toyota was alone in the auto industry in their consistent use of sets. While the existence of an aerospace firm which did practice SBCE would clearly have reinforced the theory, the absence of such an example should not be interpreted as a failure of the basic concepts.

The presence of multiple set-based practices within the aerospace industry, on the other hand, does support many of the key tenants of SBCE, if only in isolation rather than as part of an integrated design strategy. The apparent implication of the somewhat ambiguous assessment provided by this study is that further research is required. To that end, the following chapters present a model for the implementation of a complete SBCE approach to design, as well as what additional efforts can be made by and with the government to further encourage set-based practices.

### 10.4 Lessons from the Examples

The eleven examples presented in the preceding chapters all provide a variety of lessons which will be used to help develop a model for implementing SBCE. Since references will be made to these lessons, they have been summarized in Table 5 which appears on the following pages.

**Table 5: A summary of the lessons from the industry examples.**

| EXAMPLE | LESSONS |
|---|---|
| 1: Capitalizing on the Point Design | · Use a point-based approach when there are a limited number of requirements.<br>· Aim to satisfy most of the customer's requirements, but do not attempt to satisfy every requirement at the expense of time or cost. |
| 2: Design by Constraint | · Keep detailed notes of different design options.<br>· Develop designs that satisfy the customer's needs, rather than ones that are "ideal."<br>· Allow subsystem engineers to work independently until their designs are released to the rest of the team.<br>· Subvert the entire design process to the customer's primary needs to ensure that these needs are addressed first. |
| 3: Combining Parallel Concept Development & Conceptual Robustness | · Use a baseline design as a point of departure for the development of sets of design alternatives.<br>· In the case of complex systems, allow specialty groups to guide the development of design alternatives explored by teams of engineers.<br>· Use conceptual robustness to guard against downstream decisions invalidating previously made choices. |
| 4: Subsystem Installation | · Apply set-based techniques when a design problem includes a large number of design variables, many possible combinations of these variables, and conflicting requirements.<br>· Use minimum/maximum constraints to allow flexibility in the design process and to aide designers when selecting between several options.<br>· Use computer tools to speed up a design process, once the process itself is well-defined. |
| 5: An Integrated Design and Analysis Package | · Use computer tools to allow fewer engineers to consider a greater number of design options.<br>· Use simple computer models, such as historical databases, to provide quick analyses of design alternatives.<br>· Enhance individual and corporate learning by allowing individual engineers to consider large numbers of design alternatives during every program. |
| 6: Using Computer Tools to Institute Process-Based Design | · Use computer analysis tools sharing a common database to help decouple design problems.<br>· Perform parametric trades from multiple engineering and design perspectives; determine optimal system compromises by comparing these trade studies.<br>· Use early trade studies to identify relative differences between design options, rather than to identify the best option.<br>· Define convergence ranges to guide the narrowing process.<br>· Establish decision gates to clearly indicate when options should be eliminated. |
| 7: A Company Beginning to Implement a Platform Strategy | · Standardize design processes to shorten development cycle times and reduce mistakes.<br>· Look for regions of overlap both between specialties within a program and between programs.<br>· Share selection lists between engineering specialists to allow engineers to understand what options are available to other members of a design team.<br>· Arrange the design organization based upon the product's design, so that engineers can support multiple programs. |

**Table 5, continued.**

| | |
|---|---|
| 8: The Evolution of Standardized Products and Processes | · Begin product development reforms on the manufacturing floor. Then work backwards through the entire development process.<br>· Assign clear responsibilities and authorities to the appropriate elements of a design team throughout the development cycle.<br>· Once committed to a design, do not make changes to it unless absolutely necessary.<br>· Arrange the design organization based upon the product's design, so that engineers can support multiple programs.<br>· Align the product, the manufacturing process, the design process, and the organization to complement each other.<br>· Use the World Wide Web and corporate intranets to share design knowledge across the company. |
| 9: Design Re-Use | · Allow one group of engineers to develop a product design to a given point, and then allow another group to further refine the design.<br>· Align the product, the manufacturing process, the design process, and the organization to complement each other. |
| 10: The Cost of Considering Alternatives | · Consider multiple design options when additional designs have the potential to lower a system's cost.<br>· Consider multiple designs in parallel when schedule constraints do not allow for design alternatives to be considered one at a time.<br>· The cost of considering multiple options can be justified if the final design, a result of considering the additional options, lowers the total system's cost. |
| 11: The Impact of Cost as an Independent Variable | · Consider multiple design options when additional designs have the potential to lower a system's cost.<br>· Use minimum constraints to allow flexibility in the design process and to aide designers when selecting between several options.<br>· Establish cost (or performance) goals to allow flexibility in the design process and to avoid establishing firm requirements too early in the development process. |

# 11. A Model for Lean Set-Based Concurrent Engineering

## *11.1 Chapter Introduction*

Thusfar, the first two goals of this research have been addressed: SBCE has been defined and a preliminary assessment was made of the set-basedness of the aerospace industry. The only remaining goal is to recommend how SBCE could be applied to aerospace product development projects. That is the intent of this chapter.

To that end, the following sections will detail how to implement a product development and design process using set-based techniques. One lesson which was revealed during the course of this research, however, was that to consider one feature of a design process without considering the process as a whole limits the amount of improvement one can achieve. Therefore, the following sections expand the scope of this investigation somewhat, to provide a much more overarching view of product development processes. This broadened view is not intended to be authoritative or absolutely complete. It is intended, however, to help place set-based techniques within the framework of an integrated design process.

The model is developed in the following sections through several steps. First, a basic question is answered: when should set-based techniques be applied? Once this answer is developed, the model is introduced initially by way of analogy to manufacturing processes. Finally, the complete model itself is presented, beginning with the concept of lean and concluding with individual and corporate learning.

### *11.2 When Should SBCE Be Used?*

The most important question a program manager should consider prior to attempting to use SBCE is whether or not set-based methods are in fact appropriate for his or her development project. As the examples of the previous chapters demonstrated, many design processes are quite successful without ever formally applying set-based methods. Therefore, the first step in applying SBCE is to ask, Is SBCE the right approach for a given project?

The examples provided in the previous chapter point to program attributes which can be used to determine the applicability of set-based practices. Example 1, for instance, seemed to indicate that when a program faces a limited number of requirements which can not be traded away, point-based techniques can be very successful. On the other hand, Example 4, which made use of many set-based methods, was characterized by a large number of conflicting requirements and variables. Importantly, however, in this second example many of the requirements could be adjusted, allowing for a variety of design trades to be considered. More generally, programs which were characterized by flexibility in requirements and customer needs tended to make more extensive use of set-based techniques than programs which faced very tight constraints.

These results are consistent with the theories presented in the SBCE literature. Sobek (1997) for example, suggests that point-based approaches are best suited to projects in which there is a mandate to use a specific technology (p. 239). In Example 1, the nature of many project requirements did in fact force the company to select from a very limited number of technologies. This example, therefore, seems to support Sobek's conclusion. Furthermore, point-based approaches are also appropriate for projects in which systems are not tightly coupled and where the problem and the technologies are well understood (Sobek, 1997, p. 14).

In terms of set-based approaches, Sobek (1997) proposes that these methods "seem most suited to contexts with tightly coupled complex systems, where rapid learning is critical and when the problem is not well understood" (p. 13). Again, these conclusions match closely to observed trends, particularly in the case of Example 4. When there are a large number of connected variables, the ability to consider and then compare multiple configurations in parallel allows engineers to gain a better understanding of the problem and, therefore, make better design decisions.

Table 6 summarizes these conclusions and provides a method to determine which approach -- point-based or set-based -- is most appropriate for a given design problem. Note, however, that this choice in design method is made at the highest level of the design process. At very detailed levels, engineers will likely always make use of both set-based and point-based design techniques. The guidelines in Table 6 are intended to help managers determine when the *overall* approach to a design problem should be point-based or set-based. Regardless of the choice that is made, managers and engineers should recognize that any design strategy will ultimately use methods and strategies from both design philosophies.

**Table 6: Criteria for selecting between set-based and point-based design strategies.**

| ***If the development project is characterized by:*** | ***Then apply:*** |
|---|---|
| · A large number of design variables<br>· Tight coupling between design variables<br>· Conflicting requirements<br>· Flexibility in requirements allowing for design trades<br>· Technologies and design problems which are not well understood, and, consequently, require rapid learning | Set-based techniques |
| · Requirements for specific technologies<br>· Requirements to optimize the design along only one or two dimensions or parameters<br>· Well-understood technologies or design problems | Point-based techniques |

## *11.3 An Analogy: How Design is Like Manufacturing*

To help motivate the model presented in the following sections, it is instructive to begin by way of analogy. As was noted in Chapter 2, the goal of lean product development is to have a design process which flows, gradually refining a concept until it meets all of the customer's requirements and can be easily manufactured. Since these concepts are borrowed from the world of manufacturing, it is worthwhile to explore the analogy in slightly greater detail, as a means of motivating the model to follow.

In manufacturing, one typically starts with a large piece of material which will be gradually machined away. The initial size of this piece is important. If it is too large, significant time and effort will be wasted making clearing cuts, simply machining off material that will never be used. On the flip side, if the piece is too small, material for later cuts will not be available, meaning that the part cannot be made.

Design can be viewed in a similar manner. In place of the starting block of material, one instead has the design space, an abstract region which "contains" all of the solutions for a given design problem. As in machining, the initial size of this region has important consequences. If too many design options are considered, engineers will waste time analyzing designs which will never be produced. If too few concepts are developed, engineers may not be able to satisfy all of a customer's needs.

This first piece of the analogy, therefore, provides some initial guidance as to the use of sets: The initial set of design options should be large enough to ensure that it contains a feasible final solution but not so large as to require excessive initial refinement.

Continuing with the analogy, in machining, the material is passed from one cutting tool to the next, each tool specially designed to trim away more and more of the material. In a lean manufacturing environment, these tools are typically arranged in U-shaped cells to facilitate easy transfer from one machine to the next (Black).

In design, therefore, one can conceive of a process in which a product concept is refined as it is passed from one engineering group to the next. Adapting the analogy, specialty engineers and their analysis techniques can be thought of as the "machine tools" of the design process. One must be careful with using the analogy at this point, however. In manufacturing, once a part is removed from a machine, that machine need no longer be involved in the manufacture of the part. In design, however, engineers often must participate in the design process, then observe changes which are made by other engineers, and then possibly make adjustments to the design.

To understand how a design process might avoid these needs for adjustments and corrections, consider again the manufacturing process. In machining, an upstream machine only removes as much material as is absolutely necessary, leaving other chunks of the material for removal by downstream machines. So long as the upstream machines do not remove too much material, downstream machines can operate without communicating directly with the upstream ones. Furthermore, as long as both upstream and downstream machines are aware of the needs of the final product, their cutting processes can be designed to ensure that a part need not be passed back to upstream machines for rework.

To apply these concepts to design, one must again return to the notion of sets. Upstream engineering groups are now responsible for narrowing a set to a given degree, such that all of the remaining design solutions are acceptable from their perspective. These remaining options are then passed on to the downstream groups, who can pick and choose from among the narrowed set those designs which best suit their needs. Since any of these designs have been cleared as acceptable by the upstream groups, there is little need for these groups to make adjustments once the downstream groups have made their selections. In this fashion, set-based methods can help to eliminate the backflow associated with point-based methods.

These concepts will be explored further in the coming sections. The basic notion of the analogy should be kept in mind, however: A set of designs is like the starting piece of material in a manufacturing process. Engineering groups are like cutting tools. By properly managing sets, a design can be gradually narrowed to a final solution as each engineering group eliminates design concepts in turn, just as a series of machine tools are used to cut away a block of material into a final product.

### 11.4 The Model for Lean Set-Based Concurrent Engineering

As was noted in the introduction to this chapter, to present set-based methods in isolation from the total product development process limits the benefits which can be realized by applying such techniques. The following sections, therefore, provide a step-by-step approach to improving design processes and then incorporating set-based approaches within those improved processes. Again, the intent is not to provide a final solution to the design of the development process, but rather to help put set-based techniques in context, to better enable engineers to apply the theories.

### 11.4.1 Why is the Concept of Lean Important to SBCE?

As the title of this chapter indicates, many of the important aspects of the following model will be based around concepts of "leanness". A fair question, therefore, is why is this concept important for the implementation of SBCE? The answer is best understood by considering the concept of leanness at its most fundamental level. When one strips away all references to production systems, product development techniques, or other processes, the concept of lean becomes simply *the elimination and minimization of waste* (Womack and Jones; Shingo; Black).

Set-based techniques require that engineers be able to consider a large number of design options. If the development process which they use to conceive of and then review these concepts is filled with wasteful practices, engineers will have a difficult time actually analyzing all of the designs. If the process is lean -- that is, if waste has been eliminated from the process -- engineers will be able to work through multiple design options very rapidly. Therefore, *leanness should be considered a prerequisite for implementing set-based techniques*. The first steps presented in the following sections help to clarify and define these concepts.

### 11.4.2 Definitions: Operations versus Processes

Prior to developing this design model, it is important to distinguish between *processes* and *operations*. Using the definitions developed by Shingo, "[p]rocesses transform materials into products," while "[o]perations are the actions that accomplish those transformations" (p. 4). Thus, a *design process* transforms information -- knowledge about user needs and about parts required for assembly technologies, technical information, and information about competitors -- into a completed product design (Sekine and Arai, p. 25). A *design operation* could consist of a great variety of actions, from requirements analyses to CFD modeling to detailing part dimensions on a CAD rendering. Typically, the images generated at the mention of "design" -- of engineers working at computer terminals, of engineering drawings, etc. -- are design operations. Taken together, these operations compose the design process.

### 11.4.3 Improve Processes, then Operations

The first important aspect of design process reform is that processes should be improved before operations (Shingo, p. 5). To comprehend the reasoning behind this statement, one need only consider a series of operations within a process. If the process as a whole is slow, it makes little sense to speed up a single operation; instead, "[w]hat counts is the average velocity" (Womack and Jones, p. 178). These concepts are similar to ideas espoused in the theory of constraints. As Goldratt describes, a typical wrong hypothesis about cost is that the only way to achieve good cost performance is through good local performance everywhere. In reality, however, the best way to achieve good performance is to ensure good performance at the system level first, and then, once the system is properly arranged, to improve local performance.

The problems associated with attempting to improve local performance before improving system performance can be illustrated most easily using a manufacturing system[8]. Suppose one machine, in the middle of the system, is dramatically improved, allowing it to process many more pieces than had previously been the case. The rest of the manufacturing system, however, is not changed. The results are relatively obvious: The one machine will rapidly process a piece and then wait as the machines ahead of and behind it complete their work. Since the system's speed has not been changed, no real advantages were realized by improving just the one machine. Thus, the importance of the first statement in this section: improve the process first, then improve the activities which make up that process.

These concepts can and should be applied directly to design processes. Investments in an advanced computer analysis tool, for example, are efforts to improve *operations*. Reorganizing a design team such that its structure mirrors the product which it is developing, on the other hand, is an example of improving *process*. Importantly, the company discussed in Example 6, Using Computer Tools to Institute a Process-Based Design Method, illustrates the need to reform processes and then operations. The company began by first defining an ideal design environment -- by defining an ideal process -- and then moved to develop tools to realize this environment -- to develop operations to support the process.

Another example of this issue relates to the differing abilities of some analysis groups to complete their work. Many of the engineers interviewed during this investigation noted that some specialties can complete their analyses very quickly, while others can require days or weeks to complete the review of a single design. When working to improve design operations, therefore, reformers should take such issues into account. Efforts to speed up the slower moving functions will ultimately prove more beneficial then efforts to accelerate already rapid analysis tools.

Furthermore, the need to improve the processes is paramount when implementing SBCE. Since design tasks will have to be completed numerous times in the course of a single development project (in order to analyze multiple designs), process inefficiencies will be revealed quickly, and these inefficiencies will degrade the problem solving effort.

### 11.4.4 Start in the Factory

An interesting trend noted at several companies which have successfully implemented design process reforms was that these efforts began on the *factory floor*, not in the engineering offices. Example 8, the Evolution of Standardized Products and Processes, provides a good illustration of this practice. Process designers first organized assembly and testing schedules in the factory and then used these schedules to determine the engineering and design timelines.

---

[8] This example is adapted from Goldratt.

Similar practices can be used with the products of the design effort, beyond just its schedule. In effect, the purpose of design is nothing more than to develop the plans needed to actually build the product. As one manager explained, he in fact dislikes the term "design process;" to his way of thinking, design is an element of the manufacturing process, and should be referred to as such. Therefore, the needs of the factory should be used as guidelines to determine the needs of the design process. Any design operations which do not directly support the needs of the factory should be considered for elimination.

### 11.4.5 Understand Value

As Womack and Jones note, the first step in becoming lean is to define value. Importantly, value "can only be defined by the ultimate customer," although it is the producer who *creates* value (Womack and Jones, p. 16). Furthermore, value must be defined for a specific product, with specific capabilities, at a specific price, and for a specific customer (Womack and Jones, p. 19).

Thus, the first step in any development effort, whether it is point-based or set-based, is to define the value provided to the customer by the final product. Typically, engineers in the aerospace industry work under a contract which spells out how the customer will measure value: the cost of the system, the range of an aircraft, the coverage area of a satellite, the accuracy of a missile, etc. Depending upon the importance placed on such requirements, designers can gain a good sense of which product features are most valued by the customer.

The mission of engineers and managers, therefore, becomes arranging the design process to generate a product which provides the value requested by the customer. As was shown in Example 2, Design by Constraint, once value is viewed from the customer's perspective, the sequence of the design process itself might change. In general, this sequence should first address the customer's most important needs, when the design is still the most flexible. Later steps in the design process should be used to decide upon features which are less valued by the customer. In this manner, the design process itself can be tailored to ensure that engineers develop a system which meets the customer's definition of value.

### 11.4.6 Eliminate Waste

Concepts of leanness are based primarily on the Toyota Production System (TPS). Thus, to understand lean techniques, one can turn to this manufacturing system. As Shingo states, the Toyota Production System is, above all else, "a system for the absolute elimination of waste" (p. 67). As was noted earlier, eliminating waste -- becoming lean -- is an important first step to implementing set-based practices.

Shingo defines waste as "any activity that does not contribute to operations" (p. 76). Returning to the definition of operations and design, waste, therefore, is any activity which does not accomplish the transformation of knowledge into a product concept(s). Womack and Jones broaden Shingo's theories to define six types of waste which can be grouped into two categories of operations. The six examples of waste are (Womack and Jones, p. 20):

1. Mistakes which require rectification.
2. Production of items which no one wants, leading to the build up of inventories.
3. Processing steps which are not actually needed.
4. Movement of employees and transfer of goods from one place to another without any purpose.
5. Groups of people in a downstream activity waiting because an upstream activity has not completed its work on time.
6. Goods and services which do not meet the needs of the customer.

The two types of operations are (Womack and Jones, p. 38):

Type One: Steps that unambiguously create value.
Steps that do not create value but which are currently unavoidable.

Type Two: Steps that create no value and can be eliminated immediately.

Adapting these examples to the design environment is quite easy. For instance, "production of items which no one wants" results when engineers design a subsystem based on an incorrect understanding of the baseline design. "Processing steps which are not actually needed" include practices such as conducting detailed computer analyses on a very preliminary design that is likely to change. Analyzing a design process and then identifying activities as either Type One or Type Two allows engineers and managers to "lean out" a development process.

One important caveat must be noted relative to waste in a design process. At first glance, the consideration of many design options which are ultimately not used may appear to be waste. The designs are not included in the final product, and, consequently, appear to have added no value to the process. But upon further reflection, the opposite conclusion is probably more correct: Considering multiple options *adds* value to the development process. When engineers analyze a variety of design possibilities, they learn more about the trades which they must make and what features best satisfy a customer's needs. Thus, by considering options, engineers create knowledge about a design problem. As has been noted throughout this thesis, one of the primary objectives of a design process is to create knowledge. Therefore, considering multiple options adds value to a design effort, rather than creating waste.

Further, in many instances observed during this research, the consideration of new options enabled engineers to improve their designs. Take, for instance, Examples 10 and 11. These examples demonstrated that by considering a variety of design options, engineers were able to deliver a product which met the customer's basic needs at a minimal cost. In these cases, value was added to the process not simply in the form of additional knowledge, but specifically in terms of the customer's definition of value: The designs exceeded a minimum level of performance while remaining under a given cost. Again, the conclusion seems clear: Considering a variety of alternative concepts adds value to the design process.

### 11.4.7 Organize Design Teams to Mirror the Product Architecture

Complex products will often have to be decomposed into smaller elements in order to ease the product development problem. Thus, the nature of this product decomposition "implies a specific pattern of organizational communication" (Gulati and Eppinger, p. 14). The product's decomposition scheme is defined by its architecture -- "the set of technical decisions... for the layout of the product, its modules, and for the interactions between the modules" (Gulati and Eppinger, p. 5). The product's architecture, therefore, can be used as template on which to base the design of the product development team (Anderson; Gulati and Eppinger). This relationship is illustrated in Figure 30.

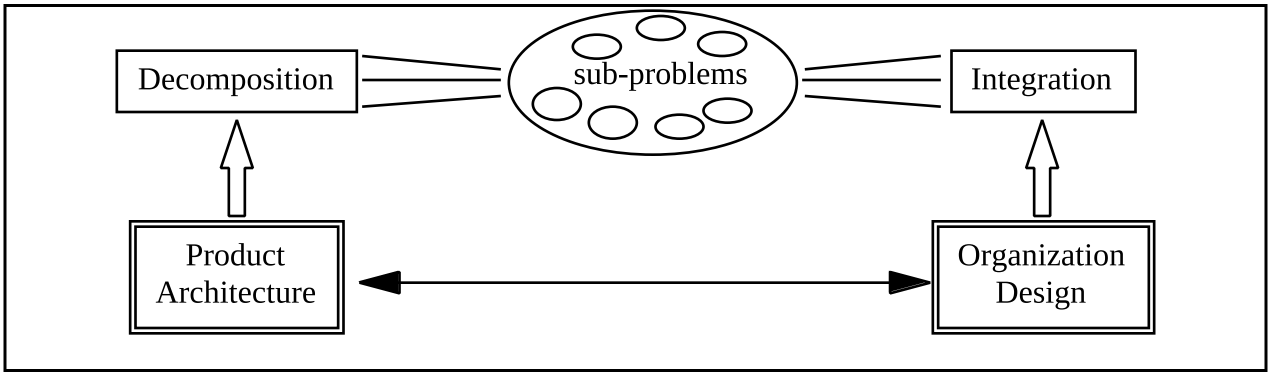



**Figure 30: The link between product architecture and organization design.** A product's architecture is used to decompose a complex design problem into several simpler sub-problems. Proper organizational design then helps to integrate the solutions to these sub-problems into a coherent final solution. (Adapted from Gulati and Eppinger, p. 26)

Note that this link between product architecture and design organization is used by Toyota. As described in Chapter 4, Toyota's design teams are organized along "functional" lines. These functions, however, mirror the subsystems of the cars which they are responsible for designing. Body, chassis, and powertrain, for instance, are all referred to as functional groups, but these titles also match the major subsystems of the products. Toyota's design teams, therefore, are organized by the architecture of their products.

This method of organizing a design team has implications for both the team's performance (in terms of delivering a product) and the product's performance (in terms of how well it satisfies its customers). Von Hippel (1990), for example, notes that "one of the primary difficulties many firms encounter in their attempts to respond to innovation [i.e., design] problems of a novel type is their continued reliance on historically derived but now suboptimal divisions of the problem-solving task" (p. 409). Rather than considering how design tasks might be interdependent, firms tend to rely on "assumed economies of specialization" (von Hippel, 1990, p. 410). Instead, firms should consider not only how to design the product's architecture, but also how this architecture should influence the team's organization.

In the case of an unprecedented product, designers may not know in advance how the product will be decomposed (Browning, 1997, p. 87). In such circumstances, therefore, the organization must be flexible so that as the product is partitioned and decomposed and its architecture defined, the organization evolves to match (Browning, 1997, p. 87; von Hippel, 1990, p. 407). In following such a pattern, a firm's design team organization will be a function of the specific product under design. Studies suggest that "fixed organizational structures generate products whose architectures remain fairly rigid" (Gulati and Eppinger, p. 17), so by altering a design team's organization for each new development project, a firm increases the likelihood that the team will develop a novel design solution. This adaptability can then have an impact on the firm's ability to compete against other companies as well as potentially increasing customer satisfaction with the product.

Following such a strategy for organizational design has distinct advantages when implementing SBCE. If a design team is organized along the product's architecture, each "specialty group" in effect becomes an element of the product itself. In the case of Toyota, for example, body engineering was responsible for considering all of the possible structural designs. The sets developed by each group, then, represent different product configurations. In a functional organization, by way of contrast, the responsibility for developing design alternatives would be spread out among several groups. A product-based design team, therefore, helps to organize and track the evolution of the design sets.

### 11.4.8 Standardize Design Operations

Reinertsen notes that "[i]t is very easy to treat product development as a black art rather than a science, because so many of its elements are unpredictable" (p. 2). Yet, while "many product designers liken themselves to the fine artist," in reality, "[d]esigning a product is not like creating an oil painting" (Sekine and Arai, p. xiii). As Sekine and Arai explain, "in this age of diversification it is not possible to handle the myriad of products while treating the design process as a fine art -- rather, it is necessary to stratify it" (p. xiv). Hence the need to standardize design process.

As described in Section 4.2.3, an important element of the Toyota design process is its use of standardized design methods. These efforts have had several benefits for Toyota, which can be realized by other companies who choose to follow a similar approach. One of the most important of these advantages is that standardized practices provide a basis of comparison for design process improvement efforts. Womack and Jones note that "[j]ust as activities that can't be measured can't be properly managed, the activities necessary to create... a specific product which can't be precisely identified, analyzed, and linked together cannot be challenged or improved (or eliminated altogether), and, eventually perfected" (p. 37). By adopting a standardized design method, engineers can "accurately measure throughput time to continually improve the design methodology itself" (Womack and Jones, p. 54).

Once the individual operations within a design process are standardized, new possibilities for the design process are also created. Example 2, Design by Constraint, suggested that a design process should be tailored to specifically address the needs of a particular customer. This notion was reinforced when the concept of value was considered. Thus, an ideal design process would be one that was highly flexible, allowing it to be easily tailored to each new development challenge.

The desire for such flexibility suggests that a design process should be built out of modules (i.e., operations) which can be arranged and tailored to meet the exact needs of each design problem. Therefore, managers and engineers should strive "to create standardized building blocks that are defined primarily at their interfaces rather than by their internal procedures" (Reinertsen, p. 121). Standardized interfaces between design operations would allow the individual steps to be reconfigured to address a given customer's requirements in the needed sequence. In this fashion, standardized operations facilitate a highly adaptable design process.

Standardization is also important specifically for implementing SBCE for the development of complex products, such as those encountered routinely in the aerospace industry. Example 10, the Cost of Considering Alternatives, demonstrated that to develop multiple design options typically requires that a small team of engineers be assigned to each alternative concept. Standardized practices then become essential to ensure that the products of these multiple teams are directly comparable. If each team conducted its trade studies in a different manner, for example, the results of those studies could not be compared. Thus, *whenever different teams are used to develop design alternatives which will then be compared, the design processes of these teams must be standardized.*

### 11.4.9 "Right-Size" Design Tools

Another important concept borrowed from lean manufacturing systems is referred to as "right-sizing" by Womack and Jones. This concept states that a tool should be sized to match the size of the operation in which it will be used (Womack and Jones). Right-sizing is best understood by considering tools which do not adhere to this idea. Womack and Jones label such tools as "monuments:" "any machine which is too big to be moved and whose scale requires operating in a batch mode" (p. 175). In contrast, a tool that has been right-sized is flexible and can be used with small batches.

In design processes, monuments are most likely to appear in the form of complex, highly sophisticated computer tools. As described in Chapter 3, set-based methods work best when used with simple, quick models and analysis techniques. Example 5, An Integrated Design and Analysis Package, demonstrated that such approaches are not incompatible with aerospace design problems. The lesson, then, is that companies should avoid investing large sums of money to develop highly detailed computer simulation tools to be used early in the design process (Sobek, 1997, p. 239). Such tools precisely fit Womack's and Jones' definition of a monument. Instead, computer analysis tools used early in a product's design must be adaptable so they can keep pace with the many changes that are likely to occur at the beginning of a product's evolution. Design tools used late in the design process, when changes are made at a much slower rate and a high level of fidelity is required in analyses, can be complex and sophisticated. Efforts to employ such systems earlier in the effort, however, are likely to result in few benefits.

### 11.4.10 Use Sets to Make the Design Process Transparent

Womack and Jones assert that an important feature of lean systems is that they are transparent: everyone in the system "can see everything, and so it's easy to discover better ways to create value" (p. 26). Point-based design methods, in which engineers only share their best design options, are not at all transparent. In fact, these methods can be thought of as opaque: engineers prevent other members of a design team from viewing the options which they considered since they present only their best idea.

Set-based practices, in contrast, are very transparent. By communicating about sets, engineers share all of their ideas and constraints with one another. As Womack and Jones suggest, this sharing allows engineers to work together to develop the best overall design, i.e., to discover the best way to create value.

Beyond simply improving the product, transparency is also intended to ensure that problems are discovered early, so that they can be eliminated (Womack and Jones). As described in Chapter 3, sets are a useful tool for tracking the progress of a design. An element of a design whose set is larger than others late in a development project is an indication that engineers are still uncertain about the solution -- it is a signal that there might be a problem. When a set becomes small, on the other hand, engineers are indicating that they are confident that they are near a final solution. By openly indicating the size of a set and thereby creating a transparent process, design problems can be quickly identified and the appropriate resources devoted to their resolution.

### 11.4.11 Educate the Customer

As described in Section 10.2.3, if the customer is not well-informed about the use of ranges or tolerances in a design process, the customer is likely to inadvertently increase the risk of the design process. Similarly, as demonstrated by Toyota's relationships with its suppliers, a mutual understanding of goals and constraints is essential to facilitating set-based techniques.

Thus, as a company moves to implement set-based concurrent engineering, it must educate its customers about their role in the process. Three points should be conveyed. First, customers must be trained in the use of set-based requirements, such as ranges and minimum/maximum constraints. Second, as described in Chapter 10, designers must help customers to understand that once requirements are established, the customer should not try to change them, even when he sees large margins early in a concept's development. Instead, the customer must trust the designer to develop the best design possible. If the customer attempts to raise a performance requirement because preliminary design estimates indicate some margin, for example, the customer will increase the risk of the development process. Perhaps more importantly, such actions by the customer may invalidate other decisions which have made in an attempt to optimize the design. Finally, customers must trust that the designer will indeed deliver the best possible product. Such trust must, nonetheless, be earned by the designer.

Note that a company which is first attempting to implement set-based practices might require a "leap of faith" on the part of customer. The designer must say, "Trust me," and the customer must grant this trust, even though neither side may have a past history of such relationships. If the designer does deliver as promised, the customer will be reassured that the process can work, and will be inclined to follow such practices again.

### 11.4.12 The Concept of Flow

Once a design process has been improved by defining value from the customer's perspective and by eliminating waste, the next step is to make the process flow. When a process flows, a product “is worked on continuously from raw material to finished good” (Womack and Jones, p. 22). The primary aim of flow is to eliminate the waste associated with downstream groups waiting for products from upstream groups. As with other lean concepts, the theory of flow also comes from the world of manufacturing. In manufacturing system design, a major source of waste is when stock develops ahead of or behind a machine because other machines are not working at the same rate (Shingo; Black). To prevent such a build up, production systems are designed such they are leveled, that is, equal amounts of product are manufactured by each step in the process (Shingo, p. 27). The *kanban* system so commonly identified with the Toyota Production System is in fact the mechanism used to help control the flow of a leveled process (Shingo).

The goal of a flowing system is that all backflow, scraps, and stoppages are eliminated (Womack and Jones, p. 52). As Sekine and Arai state, the point is “to create an uninterrupted flow in the design process, one that is free from ‘stagnant pools’” (p. 31). Implementing such a system within a design process is challenging. As was described in Chapter 2, the typical solution is to implement “concurrent development.” The fact that one design decision depends upon the outcome of another, however, often means that some sort of sequence must be established for making decisions. Concurrent development, therefore, reverts to a serial process with an emphasis on short, quick feedback loops and rapid iterations. Such a process, however, limits the degree to which the overall design effort can flow: serial, point-based techniques require that engineers accept and in fact plan for iteration and backflow. As was described in the analogy presented earlier in this chapter, the use of sets can aide in fully realizing this desire for flow. But prior to explaining how this flow can be achieved, another important concept must be introduced: pull.

### 11.4.13 Controlling Flow with Pull

The notion of pull can be expressed in a variety of ways, but the basic idea is that an upstream process should not produce anything until it is requested by a downstream process

(Womack and Jones). As was noted in the previous section, the *kanban* system was invented by Toyota to control its production process, and that technique is a pull-based system. When a downstream process requires a part, a *kanban* card is sent back to an upstream process, instructing the upstream process to produce the part (Shingo). The downstream process, therefore, "pulls" the upstream process to produce an item.

Thus, pull is intimately linked to flow: Pull ensures that the production rate of upstream processes is matched to the needs of downstream processes. Abstracted to a slightly greater degree, pull states that information should flow in the direction opposite to the primary flow of the process (Cochran). *Kanban* cards, for example flow from downstream processes to upstream ones, while the production process itself flows from the upstream processes to the downstream processes. As will be explained, the same concepts can be used to help improve design processes.

### 11.4.14 Using Sets to Achieve Flow

Assuming that the preceding steps have been implemented, a design process has been organized around the customer's definition of value, all forms of waste have been eliminated, the design team has been organized to mirror the product, design processes and operations have been standardized, tools have been right-sized, and the process has been made transparent. Furthermore, the customer has been educated in set-based methods, so that the customer can properly participate in the development process. SBCE can now be implemented by combining the notions of flow and pull with sets, and linking these ideas to the lean development process. First, consider the application of flow to sets.

Using the customer's requirements as a starting point, each specialty group initially considers the design problem independently. Each group develops a range of possible options, some of which may exceed their specific needs, others which might not meet all of their requirements. Specialty groups then compare their sets, looking for regions of overlap.

At the beginning of this process, all of the sets will be large. So long as the sets are large enough, a significant region of overlap should also exist. The next step in the process, therefore, is to shrink this area of overlap until only one, globally optimized solution remains. To accomplish this narrowing, SBCE turns to the notion of flow.

Whereas point-based strategies often fail to fully achieve a flowing process free of backflow, set-based techniques offer the potential to attain continuous flow. Backflow occurs in point-based methods because decisions made by a downstream designer can invalidate previous choices made by an upstream engineer. The upstream engineer, therefore, will have to reevaluate the design and make adjustments to the product concept -- this is the essence of backflow in design.

A set-based approach achieves flow through the narrowing process. Upstream and downstream designers both participate in the initial development of design sets. Rather than selecting only one concept to be passed to the downstream groups, however, in SBCE upstream engineers pass down several design options. Each of these options must satisfy the minimum needs of the upstream groups, though some designs will perform better than others. Yet, as long as the minimum needs are met by all of the designs, the upstream groups will have satisfied their requirements.

Downstream groups are then free to select from any of the designs which they received from the upstream groups. They will, of course, select the designs which best suit their own needs. Importantly, though, these downstream groups now have some flexibility in what choices they can make, while at the same time limiting the extent of any changes which might be made to upstream decisions. The likelihood for backflow has thus been reduced, moving the design process towards the ideal concept of flow.

There are several important implications of this flowing design process. The first is that the hand-off of design concepts from the upstream groups to the downstream groups is *not* a return to the over-the-wall model of design. Upstream engineers continue to participate actively in the design effort, helping to inform downstream decisions, primarily by indicating which of the remaining design options are better from upstream perspectives. In addition, as will described in the next section, the hand-off between the groups is not blind: by applying concepts of pull, the upstream group can be well aware of downstream needs.

The second implication of the using sets to achieve flow is that staffing levels can be reduced for upstream groups as the set narrows. In order to initially develop as large a set as possible, staffing levels for a project will need to be at their greatest at the very beginning. As upstream groups select designs and eliminate others, however, the staffing within these groups can be reduced. Managers should be prepared, therefore, for the need to budget for the largest staff at the start of the project, but then, as the set narrows, can expect to see the staffing needs and budgets decrease.

### 11.4.15 Using Pull to Control the Flow of Sets

As noted above, the hand-off from the upstream groups to the downstream groups is not blind, but is instead a well-informed transfer of design options. To achieve this understanding between upstream and downstream groups requires the use of pull. As described in Section 11.4.13, pull can be defined as information flowing in the direction opposite to the primary process flow.

To implement pull in design, therefore, two questions must be answered: What is the primary flow and what information should move backwards through the process? The first question was already answered in the previous section: As just described, in a set-based environment, the primary flow is defined by the narrowing of the set of design options.

Pull can then be used to regulate this flow by allowing *constraints* from downstream groups to flow backwards to upstream groups. A key aspect of SBCE is its concurrency: both upstream and downstream groups consider the design problem simultaneously. This simultaneity enables downstream groups to have a preview of the decisions that will be made by upstream groups, while also providing the downstream groups an opportunity to express their preferences for certain choices. The expression of these preferences then helps to guide upstream decision-making processes, increasing the probability that upstream decisions will be compatible with downstream needs.

The relationship between Toyota's stylists and body engineers, as described in Chapter 4, provides a good example of these ideas[9]. Initially, styling (the upstream group) considers a wide array of potential car designs. While still under development, these designs are reviewed by body engineering (the downstream group). Body engineering then releases the *kentouzu*, the study drawings, back to styling. These drawings indicate how designs might be improved to better accommodate the needs of the body engineers. The *kentouzu* are, therefore, a tool which the engineers use to communicate their constraints to the stylists. In essence, the drawings have moved downstream information back to the upstream group, to influence, or pull, the upstream group's design decisions in a particular direction.

#### 11.4.16 Apply the Principles of SBCE to Ensure a Globally Optimal Solution

Once the initial set of designs has been identified, engineers and designers can begin to narrow the number of possible alternatives. As they do so, the basic principles of SBCE should be followed to ensure that the final solution is a globally optimal design. These principles include (Sobek *et al.*; and see Chapter 3 for further details):

- Define feasible regions. Search to understand what is possible, rather than what is best.
- Seek conceptual robustness. Attempt to make design decisions which will remain valid regardless of the choices made by other engineers.
- Explore tradeoffs by designing multiple alternatives.

---

[9] As before, these concepts are based on information presented in Sobek (1997).

- Look for intersections between sets. Designs which fall within these regions of overlap will represent solutions which are optimal from a system perspective, i.e., they will be the globally optimal solutions.
- Narrow sets gradually, while increasing the fidelity of analyses. Ensure that a design is feasible prior to committing to it, and use simple tests to eliminate inferior designs.
- Stay within a set once committed to it. Attempt to continuously narrow the field of solutions. Enlarge the set of designs only if absolutely necessary.
- Control the narrowing process by establishing process gates at which integration activities take place, such as the construction of prototypes.

### 11.4.17 Ensuring that Learning Occurs throughout the Process

Wheelwright and Clark note that organizations which excel at product development typically also excel at learning (p. 53). They state that learning from each development effort is essential to strengthening "the foundation for the next iteration of the development strategy" (Wheelwright and Clark, p. 52).

In the context of engineering design, "learning from experience means learning from development projects" (Wheelwright and Clark, p. 284). Set-based methods present the opportunity for engineers to learn more than they might if point-based design methods were used. By exploring sets of options, engineers increase the number of designs on which they work for a given project (Sobek, 1997, p. 235). Each design developed and analyzed by an engineer presents her with an opportunity to better understand the limits and constraints which she faces.

The challenge for a company, however, is to transfer the learning that "goes on in the heads of individuals" to the development team as a whole (Wheelwright and Clark, p. 293). This issue is one to which large sums of research have been devoted, and this thesis will not attempt to address the issue in great detail. Suffice it to say, companies must ensure that records are kept of the various designs which are explored and then eliminated or further pursued by a design team.

Example 8, the Evolution of Standardized Products and Processes, demonstrated how technologies such as the World Wide Web can be combined with standardized design methods and record-keeping techniques (in the form of standardized design notebooks) to capture such learning. Whatever methods are pursued by a company, engineers must ensure that lessons learned while exploring the limits of one design are passed on to future designers working on similar problems.

### *11.5 Summarizing the Model*

Clearly it is important to implement SBCE as part of a broader effort to improve a product development process. The steps recommended to achieve such improvements are listed below.

1. Overall, first improve processes, then improve the operations which make up those processes.
2. Start in the factory. Assess the needs of the factory, and use these needs as guidelines for improving design operations.
3. Define value from the perspective of the customer.
4. Eliminate all waste from the development process.
5. Organize design teams to mirror the product architecture.
6. Standardize design operations. Arrange the operations so that a customer's most important requirements are satisfied first.
7. Right-size design tools.
8. Communicate using sets to make the design process transparent.
9. Educate the customer about its role in a lean, set-based process.
10. Use sets to achieve continuous flow through the development process.
11. Use the constraints of downstream groups to implement a pull-based control system.
12. Apply the principles of SBCE to ensure that the final solution is a globally optimal solution.
13. Ensure that lessons learned while exploring the possibilities and limits for one product are passed on to engineers working on similar problems in the future.

# 12. Concluding Thoughts

## 12.1 Policy Issues and Implications

The aerospace industry is one in which the primary customer is usually the government. Efforts at reforming corporate design processes, therefore, often require complementary efforts to modify government practices. Throughout this research, the impact of customer routines were seen on company development practices. Where appropriate, this research has also sought to indicate how customer behaviors can be modified to best accommodate set-based strategies. Section 11.4.11 in particular demonstrated some ways in which customer behavior can be used to reinforce set-based practices within a design firm. Companies attempting to implement SBCE are urged to do so in cooperation with government program offices. As noted previously, the first time that a company attempts to implement set-based techniques may require that the government make a leap of faith. When deciding how risky such a leap might be, the government has every right to consider a company's past performance using traditional development practices. A company which has proven itself trustworthy under such conditions is probably more likely to continue to demonstrate such behavior using new techniques.

Earlier in this thesis the author emphasized that this research was conducted solely at company sites -- no effort was made to visit government contracting offices. This omission represents a limitation to this research: Views on the government's role represented solely contractor perceptions, without the benefit of the corresponding insights of the government offices themselves. Despite this shortcoming, the observations shared by engineers should be considered carefully by the government. This research presents contract offices with an opportunity to understand the opinions and perceptions of the companies which work for them, regardless of the biases which might be present in those views.

Specific actions which contracting offices could take to support the use of set-based techniques include:

- Using performance requirements which specify *what* a system must do, but which do *not* specify how the system should do it.
- Stating requirements either as ranges or as minimum/maximum constraints. Then, as the design evolves, work with the engineers to narrow the requirements toward specific values.
- Avoiding changing requirements, even if initial product designs appear to have significant performance margins.
- Encouraging companies to develop multiple alternative designs and to share these designs with the customer. Allow the customer to comment on the designs in order to assist the engineers to narrow the set of concepts.

Note that in general these actions increase the flexibility of the designers. The intent of this increased flexibility is to help facilitate the delayed decision-making for which SBCE strives, to ensure that the most informed design decisions are made. Contracting offices, however, have every right to constrain this flexibility to some degree. In particular, customers should rigorously enforce minimum performance requirements. So long as a contractor satisfies these minimums, the designers should be free to make any decisions they wish. Customers should also be willing to consider changing these minimums, however, if the designers can demonstrate significant reasons for doing so. SBCE intends that the customer and designer work together to develop the best possible product, and such cooperation will inevitably require some give and take on both sides.

### 12.2 Final Thoughts on SBCE and the Aerospace Industry

As noted in Chapter 10, no company seemed to fully demonstrate both of the principles required to qualify a design process as set-based concurrent engineering. Many industry practices, however, contained set-based concepts or ideas. Based on the examples presented in Chapters 5 to 9, it does appear that set-based concurrent engineering could be applied to some aerospace design problems.

Companies interested in attempting to implement SBCE should consider first conducting a small-scale, controlled experiment. To conduct such a test, a company should first ensure that its design operations are standardized. A control test should first be conducted, in which engineers use the standardized operations with a traditional, point-based approach. The same product should then be designed using the same standardized operations, but this time engineers should consider sets of design options. The results of the two design efforts could then be compared to determine if set-based methods in fact resulted in an improved product, a faster development process, a less costly process, or all three. Future decisions about the use of sets could then supported or rejected based on these results.

### 12.3 Recommendations for Further Research

As with any research project, this one concluded by presenting more questions than were initially considered at the start of the effort. Some of the issues which the author believes to be the most worthy of further pursuit are listed below.

- Further exploration of the role of the government in set-based development efforts. Specifically, research should consider how to adapt Toyota's model of supplier relations to the contractor-government relationship.
- Similarly, additional research should be conducted to determine how aerospace companies could use the Toyota model of supplier involvement with their own suppliers.

- Further research into the role of computer tools in set-based development processes. Some of the most vivid examples of set-based techniques were seen in company efforts to develop advanced computer tools. These efforts, therefore, seem to be a fertile ground for the refinement of set-based methods within the aerospace industry.
- Research to develop "set-supporting" CAD programs. One issue that was discussed by several engineers was that typical CAD/CAE/CAM tools are best suited to considering just one design at a time. Such limited capabilities to rapidly evaluate multiple options inhibit the application of set-based techniques in complex product development environments. A useful avenue of research, therefore, would be to develop the tools required to adapt computer systems to support set-based concurrent engineering.
- Finally, a very important area of further research is the Toyota development process. At the time of this writing, only one detailed study existed of this design procedure. Research is needed to broaden and deepen academia's understanding of this company's unique practices.

# Works Cited


Adler, Paul S. and Robert E. Cole. "Designed for Learning: A Tale of Two Auto Plants." *Sloan Management Review*, Spring 1993. pp. 85-94.

Anderson, David M. *Agile Product Development for Mass Customization*. Chicago: Irwin Professional Publishing, 1997.

Black, J. T. *The Design of the Factory with a Future*. New York: McGraw-Hill, Inc., 1991.

Browing (1997), Tyson R. "Exploring Integrative Mechanisms with a View towards Design for Integration." *Advances in Concurrent Engineering -- CE97*, Aug. 20-22, 1997. Fourth ISPE International Conference on Concurrent Engineering: Research and Applications. pp. 83-90.

Browing (1996), Tyson R. "Systematic IPT Integration in Lean Development Programs." Cambirdge: Massachusetts Institute of Technology, Master of Science Thesis, 1996.

Carbone, James. "At Toyota, Supplier Targets and Specs Go Together." *Purchasing*, November 23, 1995. pp. 15-21.

Chang, Tzyy-Shuh, Allen C. Ward, and Jinkoo Lee. "Conceptual Robustness in Simultaneous Engineering: An Extension of Taguchi's Paramater Design." *Research in Engineering Design*, Vol. 6, 1994. pp. 211-222.

Cochran, David. Lecture notes for 2.812, The Design and Control of Manufacturing Systems. Cambridge: Massachusetts Institute of Technology, 1998.

Dixit, Avinask K. and Robert Pindyck. "The Options Approach to Capital Investment." *Havard Business Review*, May-June 1995. pp. 105-115.

Finch, William, and Allen C. Ward. "A Set-Based System for Eliminating Infeasible Designs in Engineering Problems Dominated by Uncertainty." *Proceedings of the 1997 ASME Design Engineering Technical Conferences*. Paper #DETC97/DTM-3886. Sept. 14-17, 1997. Sacramento, California.

Funk, Jeffrey. "Japanese Product-Development Strategies: A Summary and Propositions about Their Implementation." *IEEE Transactions on Engineering Management*, Vol. 40, No. 3, August 1993. pp. 224-236.

Goldratt, Eliyahu M. *Critical Chain.* Great Barrington, MA: The North Rivers Press, 1997.

Gulati, Rosaline K. and Steven D. Eppinger. "The Coupling of Product Architecture and Organizational Structure Decisions." MIT Sloan School of Management Working Paper, Number 3906, May 28, 1996.

Hauptman, Oscar and Karim Hirji. "The Influence of Process Concurrency on Project Outcomes in Product Development: An Empirical Study of Cross-Functional Teeam." *IEEE Transactions on Engineering Management*, Vol. 43, No. 2, May 1996. pp. 153-164.

Hehs, Eric. "F-22 Design Evolution, Part I." *Code One*, Lockheed Martin Tactical Aircraft Systems, Vol. 13, No. 2, April 1998. pp. 2-17.

Kalyanarum, G. and V. Krishnan. "Deliberate Product Definition: Customizing the Product Definition Process." *Journal of Marketing Research*, May 1997. pp. 276-285.

Khurana, Anil and Stephen R. Rosenthal. "Integrating the Fuzzy Front End of New Product Development." *Sloan Management Review*, Winter 1997. pp. 103-120.

Klein, Janice A. and Gerald I. Sussman. "Lean Aircraft Initiative Organization & Human Resources (O&HR) Survey Feedback -- Integrated Product Teams (IPTs)." Massachusetts Insititute of Technology Case Study, LEAN #95-03, April 7, 1995.

Krishnan (1997), V., S.D. Eppinger, D.E. Whitney. "Simplifying Iterations in Cross-Functional Design Decision Making." *ASME Transactions: Journal of Mechanical Design*, Vol. 119, December 1997. pp. 485-493.

Krishnan (1996), Viswanathan. "Managing the Simultaneous Execution of Coupled Phases in Concurrent Engineering." *IEEE Transactions on Engineering Management*, Vol. 43, No. 2, May 1996. pp. 210-217.

Krishnan (1992), Visawanathan, Steven D. Eppinger, and Daniel E. Whitney. "Ordering Cross-Functional Decision Making in Product Development." Massachusetts Institute of Technology Sloan School of Management Working Paper #3299-91-MS, Revised, October 1992.

Krishnan (1991), Visawanathan, Steven D. Eppinger, and Daniel E. Whitney. "Towards a Cooperative Design Methodology: Analysis of Sequential Decision Strategies." Massachusetts Institute of Technology Sloan School of Management Working Paper #3299-91-MS, May 1991.

Liker, Jeffrey K., Durward K. Sobek, II, Allen C. Ward, and John J. Cristiano. "Involving Suppliers in Product Development in the United States and Japan: Evidence for Set-Based Concurrent Engineering." *IEEE Transactions on Engineering Management*, Vol. 43, No.2 (May, 1996). pp. 165-178.

Meyer, Marc H. and James M. Utterback. "The Product Family and the Dynamics of Core Capability." *Sloan Management Review*, Spring 1993. pp. 29-47.

Nobeoka, Kentaro and Michael A. Cusumano. "Multiproject Strategy, Design Transfer, and Project Performance: A Survey of Automobile Development Projects in the US and Japan." *IEEE Transactions on Engineering Management*, Vol. 42, No. 4, November 1995. pp. 397-409.

Otto, Kevin N and Erik K. Antonsson. "Trade-Off Strategies in Engineering Design." *Research in Engineering Design*, Vol. 3, 1991. pp. 87-109.

Pomponi, Renata A., "Organizational Structures for Technology Transition: Rethinking Information Flow in the Integrated Product Team," Massachusetts Institute of Technology (Technology, Management and Policy Program), Unpublished Doctoral Dissertation, June 1998.

Reinertsen, Donald G. *Managing the Design Factory: A Product Developer's Toolkit*. New York: The Free Press, 1997.

Sekine, Kenichi and Keisuke Arai. *Design Team Revolution: How to Cut Lead Times in Half and Double Your Productivity*. Translated by Warren Smith. Portland: Productivity Press, 1994.

Shingo, Shigeo. *A Study of the Toyota Production System*. Translated by Andrew P. Dillon. Portland: Productivity Press, 1989.

Smith, Robert P. "The Historical Roots of Concurrent Engineering Fundamentals." *IEEE Transactions on Engineering Management*, Vol. 44, No. 1, February 1997. pp. 67-78.

Sobek, Durward K., II, Allen C. Ward, and Jeffrey K. Liker. "Principles from Toyota's Set-Based Concurrent Engineering Process." White Paper, Dept. of Mechanical and Industrial Engineering, Montana State University, Bozeman, MT., 1997.

Sobek (1997), Durward K., II. *Principles that Shape Product Development Systems: A Toyota-Chrysler Comparison*. PhD. Dissertation, Industrial and Operations Engineering, University of Michigan, 1997.

Sobek (1996), Durward K., II. "A Set-Based Model of Design." *Mechanical Engineering*, Vol. 118, No. 7, July 1996. pp. 78-81.

Suh, Nam P. *The Principles of Design*. New York: Oxford University Press, 1990.

Taguchi, Genichi. *Taguchi on Robust Technology Development: Bringing Quality Engineering Upstream*. Translated by Shih-Chung Tsai. New York: ASME Press, 1993.

VanDyke Parunak, H., Mitch Fleisher, John Sauter, and Al Ward. "A Marketplace of Design Agents for Distributed Concurrent Set-Based Design." *Advances in Concurrent Engineering -- CE97*, Fourth ISPE International Conference on Concurrent Engineering: Research and Applications. Oakland University, August 20-22, 1997. pp. 287-293.

von Hippel (1995), Eric. "'Sticky Information' and New Marketing Research Methods." Sloan Working Paper #3753-95. Cambridge: Massachusetts Institute of Technology Sloan School of Management, January, 1995.

von Hippel (1990), Eric. "Task Partitioning: An Innovation Process Variable." *Research Policy*, Vol. 19, pp. 407-418.

Ward, Allen. "Toyota, Termites, and Zero Risk System Development." Presentation to LAI Plenary Workshop, October 9, 1997.

Ward, Allen, Jeffrey K. Liker, John J. Cristiano, and Durward K. Sobek, II. "The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster." *Sloan Management Review*. Vol. 36, Iss. 3, Spring 1995. pp. 43-61.

Ward, Allen. "A Recursive Model for Managing the Design Process." DTM, 1990. pp. 47-52.

Wheelwright, Steven C. and Kim B. Clark. *Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality*. New York: The Free Press, 1992.

Womack, James P. and Daniel T. Jones. *Lean Thinking.* New York: Simon & Schuster, 1996.

# Appendix A

The following pages contain the slides used to introduce interviewees to the basic concepts of SBCE. The brief presentation includes an example of how to apply set-based concurrent engineering, based on the author's own experience as a program manager for a student-designed unmanned aerial vehicle.

# Design Methods in the Aerospace Industry

## LAI Product Development Focus Team

**Joshua Bernstein**
jbern@mit.edu
(617) 258-7984
fax (617) 258-7845

LEAN AEROSPACE INITIATIVE

## *Overview*

- Research Goals
- Literature Review Summary
  - Introduction to Set-Based Concurrent Engineering/Set-Based Design (SBCE/SBD)
  - Advantages of SBCE
  - Design Method Spectrum
- A Quick and Dirty Example
- Discussion...

LEAN AEROSPACE INITIATIVE

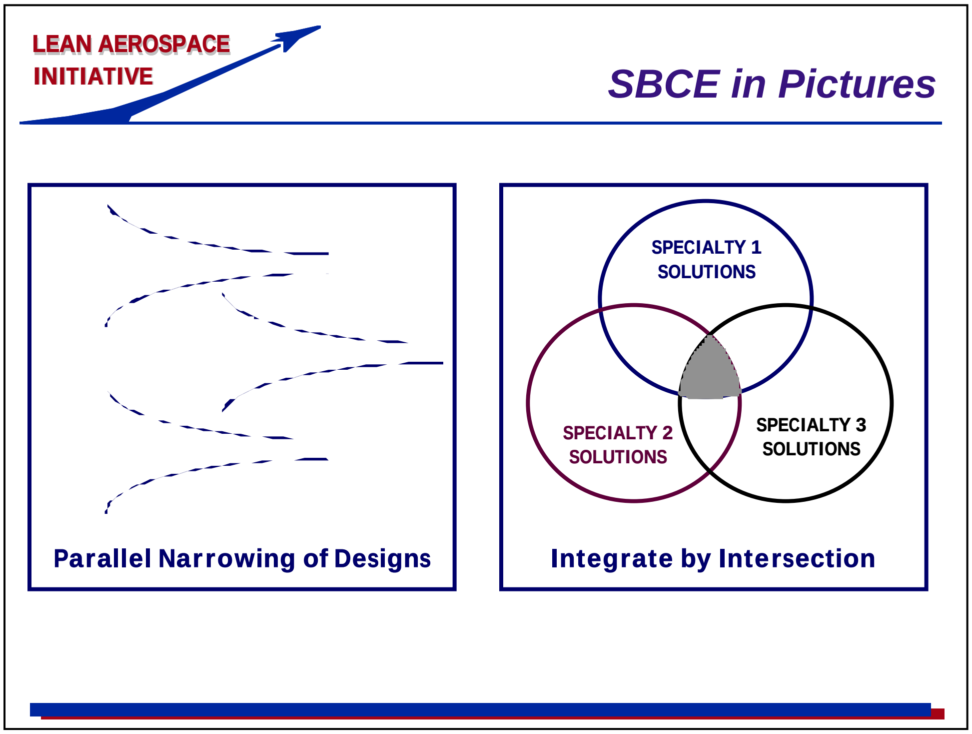
## *Research Goals*

- **Define set-based concurrent engineering/design**
  - **What are the major aspects of this method?**
  - **How does it differ from "traditional" methods and those currently used in practice?**
- ***What elements of SBCE already exist in the aerospace industry?***
  - **Assess the "set-basedness" of companies**
  - **Looking for histories of the design process**
- **Could more advanced implementation of SBCE offer advantages to the aerospace industry?**
  - **What obstacles exist that might inhibit such changes?**

LEAN AEROSPACE
INITIATIVE

## *Introduction to SBCE*

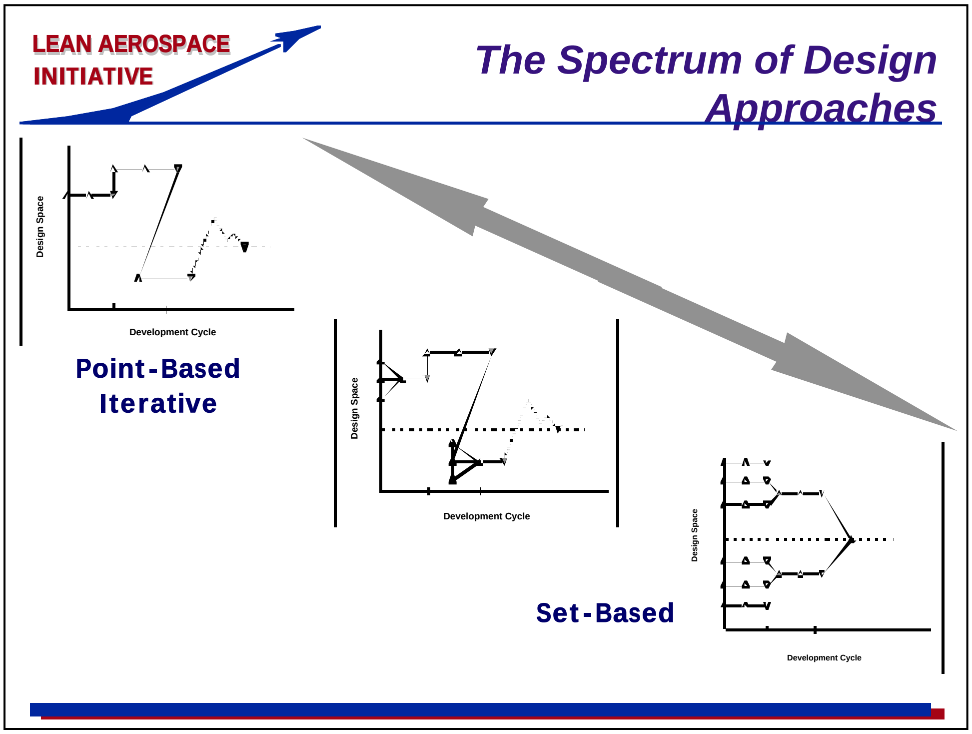
- **Delay setting requirements -- use ranges rather than point values**
- **Initially consider a large number of design concepts, i.e., *sets* of concepts**
- **Delay selecting a single concept… Carry several concepts throughout the process "*narrowing*" the set of concepts considered as the design problem evolves**
- **Communicate about sets of designs rather than individual concepts**
- **Each functional group is pursuing its own design for a given system… then groups look for *intersections* to optimize the final design**


LEAN AEROSPACE
INITIATIVE
SBCE in Pictures
SPECIALTY 1
SOLUTIONS
SPECIALTY 2
SOLUTIONS
SPECIALTY 3
SOLUTIONS
Parallel Narrowing of Designs
Integrate by Intersection

LEAN AEROSPACE INITIATIVE

## *Advantages of SBCE*

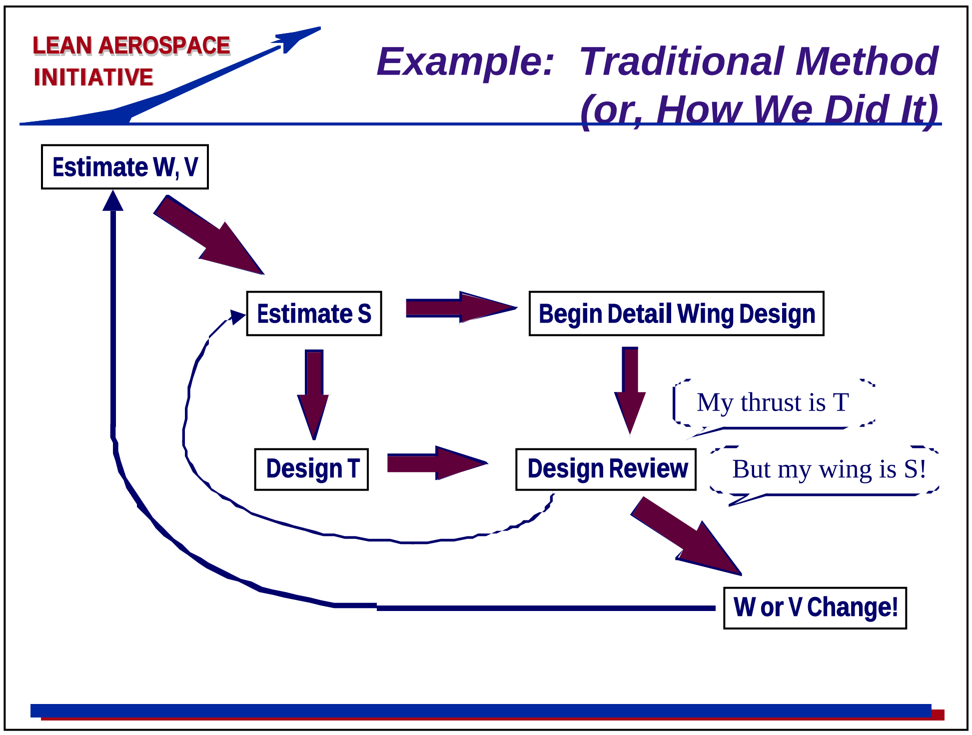
- **Improved communication between engineers**
  - **Communicate only what you know**
  - **Communicating about sets allows different specialties to understand the constraints faced by others**
- **Decisions must be valid for the entire set**
  - **Allows specialties to work independently to a given level of detail while improving coordination between specialties**
  - **Helps to prevent downstream decisions from invalidating upstream decisions**
- **Designers know to what level of detail to work based on how narrow a set has become**
- **Allows for a thorough exploration of the design space**

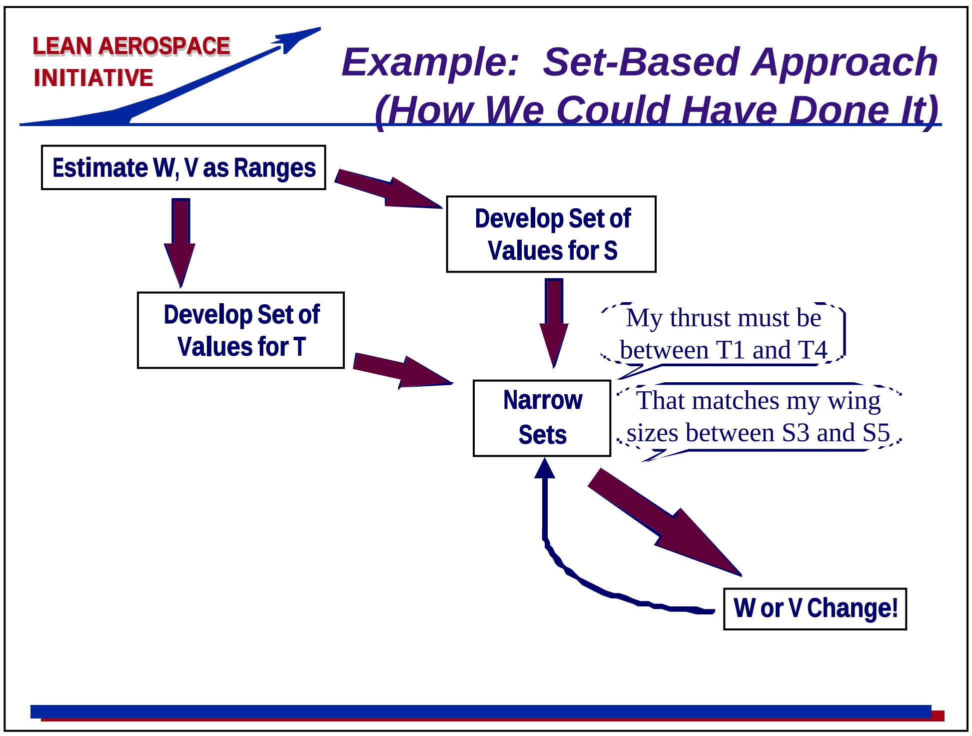

LEAN AEROSPACE INITIATIVE
The Spectrum of Design Approaches
Design Space
Development Cycle
Point-Based Iterative
Design Space
Development Cycle
Design Space
Development Cycle
Set-Based

LEAN AEROSPACE INITIATIVE

## Quick and Dirty Example: UAV for Draper Labs

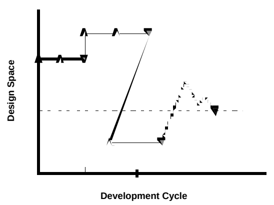
- **Consider the design of the engine and wings:**
  - volume available to store wings = V
  - weight of vehicle = W
  - wing area = S = S (W, T, V)
  - engine thrust = T = T(S, W)
  - Unique requirements environment: Goal was to maximize loiter, but no specific value was given -- just make it as large as possible

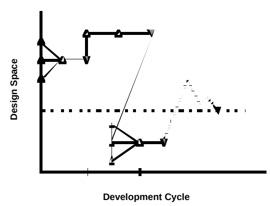

LEAN AEROSPACE
INITIATIVE
Example: Traditional Method
(or, How We Did It)
Estimate W, V
Estimate S
Begin Detail Wing Design
Design T
Design Review
My thrust is T
But my wing is S!
W or V Change!

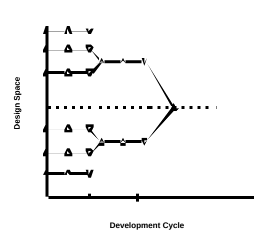

LEAN AEROSPACE INITIATIVE
Example: Set-Based Approach (How We Could Have Done It)
Estimate W, V as Ranges
Develop Set of Values for S
Develop Set of Values for T
My thrust must be between T1 and T4
Narrow Sets
That matches my wing sizes between S3 and S5
W or V Change!

LEAN AEROSPACE INITIATIVE

## *Assessing Industry Design Practices*




# Where do you fit... and why?

LEAN AEROSPACE INITIATIVE

## *Discussion*

- Your Design Methods
  - Requirements Definition
  - Number of alternatives considered
  - Iteration -- Tradeoffs between elements of the design and between functional specialties
  - Use of prototypes
  - (Team organization and communication)
  - (Communication between design and manufacturing)
  - Other aspects...
- Has there ever been an instance when you wished you could have delayed making a decision?
- Reaction to SBCE
  - Implementing SBCE in the aerospace industry

LEAN AEROSPACE INITIATIVE

## References


Finch, William, and Allen C. Ward. "A Set-Based System for Eliminating Infeasible Designs in Engineering Problems Dominated by Uncertainty." Proceedings of the 1997 ASME Design Engineering Technical Conferences. Paper #DETC97/DTM-3886. Sept. 14-17, 1997. Sacramento, California.

Liker, Jeffrey K., Durward K. Sobek, II, Allen C. Ward, and John J. Cristiano. "Involving Suppliers in Product Development in the United States and Japan: Evidence for Set-Based Concurrent Engineering." IEEE Transactions on Engineering Management, Vol. 43, No.2 (May, 1996). pp. 165-178.

Sobek, Durward K., II, Allen C. Ward, and Jeffrey K. Liker. "Principles from Toyota's Set-Based Concurrent Engineering Process." White Paper, Dept. of Mechanical and Industrial Engineering, Montana State University, Bozeman, MT.

Sobek, Durward K., II. "A Set-Based Model of Design." *Mechanical Engineering*, Vol. 18, No. 7, July 1996. pp. 78-81.

Sobek, Durward K., II. *Principles that Shape Product Development Systems: A Toyota-Chrysler Comparison*. PhD. Dissertation, Industrial and Operations Engineering, University of Michigan, 1997.

Ward, Allen, Jeffrey K. Liker, John J. Cristiano, and Durward K. Sobek, II. "The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster." *Sloan Management Review*. Vol. 36, Iss. 3, Spring 1995. pp. 43-61.

Ward, Allen. "A Recursive Model for Managing the Design Process." DTM, 1990. pp. 47-52.

Ward, Allen. "Toyota, Termites, and Zero Risk System Development." Presentation to LAI Plenary Workshop, October 9, 1997.