Implementing Design for Six Sigma

Implementing Design for Six Sigma

A Leader's Guide—Getting the Most from Your Product Development Process

Georgette Belair
John O'Neill

ASQ Quality Press
Milwaukee, Wisconsin

American Society for Quality, Quality Press, Milwaukee 53203

12 11 10 09 08 07 06 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data

Belair, Georgette, 1969–
Implementing design for Six sigma : a leader's guide : getting the most from your product development process / Georgette Belair, John O'Neill. —1st ed.
p. cm.
Includes bibliographical references and index.
ISBN 0-87389-695-5 (alk. paper)
1. Six sigma (Quality control standard) 2. Total quality management.
I. O'Neill, John, 1956– II. Title.

TS156.B438 2006
658.5′7—dc22 2006016963

ISBN-10: 0-87389-695-5
ISBN-13: 978-0-87389-695-5

Publisher: William A. Tony
Acquisitions Editor: Annemieke Hytinen
Project Editor: Paul O'Mara
Production Administrator: Randall Benson

Printed on acid-free paper

ASQ
AMERICAN SOCIETY FOR QUALITY™

Quality Press
600 N. Plankinton Avenue
Milwaukee, Wisconsin 53203
Call toll free 800-248-1946
Fax 414-272-1734
www.asq.org
http://qualitypress.asq.org
http://standardsgroup.asq.org
E-mail: authors@asq.org

This book is dedicated to the many people who have touched our lives and to the organizations that have allowed us to work with them, have celebrated with us when we've "hit home runs," and have been patient with us when the inevitable mistakes have occurred.

Contents

Preface

The comedian Bill Cosby often recalls his time playing college football. In one of his stories, the scene is the Temple University locker room. The coach is pumping the team up for a game against their feared opponent, Hofstra. When the team has reached a fever pitch, the coach yells, *OK, let's get out there and fight, fight, fight*! The team tries to rush onto the field, but the door is locked!

As leader in your company's product development organization, do you find yourself in a similar situation? Has your company tried to pump you up to go *do Design for Six Sigma*? When you've tried to run out on the Design for Six Sigma (DFSS) field, do you also find that the door is locked?

Between the two of us authors, we've spent over 20 years either developing new products or helping others get better at product development. We were introduced to DFSS quite a few years ago, and it took us a while to *get* what DFSS is all about. One of our Master Black Belt friends freely admits that it took her three experiences to finally understand the DFSS process. She took General Electric's DFSS course, she ran a DFSS project, and then she taught the GE course—after that, she finally *got it*. So it's tough enough to figure out how to *do* DFSS.

Have you also been challenged to move your organization from its current design and development process to one that can reliably produce Six Sigma designs? Implementing DFSS is a cross-functional effort that can be an even tougher door to break down. This door may be locked, and marketing, R&D, and operations may have all their weight pressed against the other side. We've written this book mainly to help you break down that door. There are some very good how-to books on the mechanics of DFSS, such as Yang and El-Haik's *Design for Six Sigma: A Roadmap for Product Development*.

Our main goal, though, is to provide you with a *game plan* to help you move the ball down the field—from your current product development world to one where DFSS has been embraced and is a working part of your processes and culture. Whether the products you develop are made of metal and plastic, or of credit plans and mutual funds, this book will help you improve your development process, so that you may deliver better products and services to your customers. Whether you develop tangible products like cars or cough syrup, or you deliver service products like mortgages and retirement plans, DFSS can help you develop robust products that your customers will want and will want to pay for.

At a high level, there's no mystery to the approach we'll offer you. We want to help you understand how your current development process is performing, diagnose the current strengths and weaknesses of your new product development approach, and plan and implement changes that will improve your organization's ability to deliver Six Sigma designs. If your company has already adopted Six Sigma, you'll recognize that our approach is to apply the Define, Measure, Analyze, Improve, Control (DMAIC) improvement method to your new product development process.

HOW WE'LL GET THERE

In Chapter 1, we'll start in the locker room as your *coach*. Some of the questions we'll pose may seem harsh, but we've got to help you decide why you want to *do DFSS*.

Chapter 2 will provide the theory and rationale behind striving for Six Sigma designs. Just in case your company hasn't embraced Six Sigma as an improvement approach, we'll provide you with the necessary Six Sigma background. We'll start to build the case for DFSS in Chapter 3, to help you think about how to *sell* DFSS in your organization and build some momentum and desire to want to break down those doors.

Consider Chapter 4 your benchmarking visit. Here, we'll give you a picture of *what good looks like*—the phases, steps, tools, and deliverables of a mature DFSS process.

With the theory and groundwork laid, it's time for you to get to work. Chapter 5 will guide you through a gap and readiness analysis—what are the differences between your current development process and *what good looks like*. It's one thing to be aware of the gaps; it's quite another to be ready and motivated to take the necessary improvement actions. We'll focus on the change management aspect of DFSS here.

We're going to get tactical in Chapter 6. We'll take you through a step-by-step approach to planning and implementing the design process changes. We'll address process changes, training and skill building, infrastructure changes, and overall change management of your DFSS initiative.

So how do you know your changes have made a difference? Measuring results is addressed in Chapter 7. What kind of metrics should you have on your scoreboard, and how do you know you're winning the game? We'll continue this theme in Chapter 8—discussing how you will know your organization is ready to transition from DFSS implementation to sustaining mode.

It's understandable to want to rest after a long game, but there's a danger in slipping back to old habits if we rest too long. Chapter 9 will address how to regain your organization's energy and keep the DFSS drive alive.

We've alluded to your work of implementing DFSS as a game. We'll close by suggesting some future *plays* in Chapter 10—directions you may take to continue to improve your development process.

Finally, even though it's not the main focus of this book, in researching this book we heard over and over that a DFSS case study would be very useful. Have fun reading our historical DFSS example in Chapter 11!

We can't promise you that DFSS will cause your stock's price to double in the next year. We can only promise that, if you dig deeply into your new product development process and follow the guidelines in this book, you can implement major improvements to this important process. We've played on the DFSS field ourselves and have been fortunate to be asked to help others do the same. We've seen the results DFSS can bring. We're hoping you'll join us. OK, let's get out there and fight, fight, fight!

A FEW NOTES

- We've included a number of references to other DFSS books and information. Some of this information is referenced via Web site URLs. As time goes by, we can't guarantee that the website will exist when you go to look for it. If you can't find it where we did, Google it!
- Although it is a common business term today, we recognize that Six Sigma is a registered service mark and trademark of Motorola, Inc. Similarly, Minitab is a software package owned by Minitab, Inc., and Crystal Ball is a software package owned by Decisioneering, Inc.

Acknowledgments

From Georgette: To my brothers and sisters, mom and dad, and my dear friends Karl, Megan, Marlene, and Jona. You have encouraged me to ask questions, to seek new paths, and to keep an open mind as I navigate through this world. To the great mentors I have had the pleasure of working with: David Perry, Mark Pomeroy, Kathy Vigue, Lucia Buehler, and Ed Kopkowski. Thank you for sharing your passion for excellence!

From John: To Larry Pabst and Mario Fedele—my *engineering fathers*, to Dr. Kazuyuki Suzuki—my *reliability big brother*, to Bill Hensler for that fateful DFSS call, to Eric Mattenson and the late Bill Lindenfelder for the chance to help shape GE's DFSS program, to Liz Iversen for the chance to help shape J&J's Design Excellence initiative, and, of course, to my wife, Nancy, and children, Mary and Michael, for allowing Daddy the time to devote to this book!

1

Introduction

IS YOUR PRODUCT DEVELOPMENT PROCESS HELPING YOU WIN THE GAME TODAY?

Think about the products your company has recently launched. Have these products achieved what you hoped for—rapid customer uptake, high sales volume and revenue/profit, high Sigma levels for critical-to-quality requirements, and low complaint rates and return/warranty costs? How are your costs compared to your competitors'?

Is R&D still trying to stabilize products with its manufacturing partners? Are you overwhelmed with the number of parts you have to manage to produce your products? If you have a Six Sigma initiative, how many of your Black and Green Belts are working on reducing product defects whose causes lie in design decisions? Are your product development efforts mostly producing line extensions of existing products?

Now think about the products that your company does consider successful. How much bigger could the product have been? How much higher could sales/revenue/profit be? How about planning the next generations of your products? As these products mature in the market, is your company actively developing/acquiring the technologies you'll need to meet the market demands of tomorrow?

Now think about what is going on inside your product development process. Do R&D teams have clear goals for what they are trying to develop? Do they have a clear understanding of what the customer wants in the product? Does product development take too long because your engineers are still mired in a build, test, fix, "design" mode? Do the two monsters of scope and feature creep roam your hallways? Does manufacturing complain about designs that can't be built? There is a corollary to this question: Does R&D

understand the capabilities of manufacturing processes and how this variation will impact what the customer sees?

Do you still *throw things over the wall* (marketing requirements to R&D, R&D specifications to manufacturing, parts requirements to suppliers)? How many of your development projects are *science projects,* where the team is trying to invent a technology and where schedule and budget are way over projections?

If these questions seem a bit brutal and personal, don't feel too bad. You and your company are not alone. While companies have spent millions of dollars on systems to better manage their manufacturing and supply chain operations, only a few have really focused on improving their *Archimedes' lever*—the design and development process.

One of us recently tried to purchase a radio-alarm clock—a simple, mature product—and returned two brands back to the store. Ah, you say, manufacturing must have screwed up! No, both were design issues.

We had a hard time finding our favorite rock station because the first clock's tuning dial would allow only coarse adjustments (we're just glad we didn't buy that clock as a gift for an elderly person with arthritis). We took that clock back to the store and purchased another brand. Ah, now the sweet sounds of heavy metal in the morning! The second brand, however, had an annoying habit of running fast, about an hour a month. We could have developed a *work-around* and reset the clock once a week, but that would reward the company for its incompetence. Back this one went, too.

Clearly there were design issues with both of these products. But why should this occur? In the 21st century, why would any company release a product that will inevitably annoy and disappoint customers? How can we even imagine it happening on mature technology like dial radios and alarm clocks? What are we missing?

DO YOU EVEN HAVE A PRODUCT DEVELOPMENT PROCESS?

Recently, we were talking with a young mechanical engineer just hired by a company who has been on a several-year DFSS journey. We asked why he had decided to change companies. He was quick to respond, "At my old company, we had a chaotic approach to product development. We didn't do market research, there was no discipline to product development, and there was a lot of conflict between the development staff and production." He went on to say that he made his job move because he was eager to learn "what a good development process can look like and how you get from bad to good!" In a few words, he had captured one of our basic learnings and a key hypothesis of this book: (1) design is a process and (2) the design process can be improved.

When we see problems in a manufacturing process, it's usually pretty easy to track those back to the *6Ms:* the method, the man, the material, the machine, the measurement system, and Mother Nature. Although sometimes they are not quite as obvious, problems in product development will usually fall into these categories.

Method Issues

Too many companies have no *real* product development process. We worked in the power plant engineering department of a large electric utility a few years ago. The department managed by engineering work orders (EWOs). Once you were assigned the EWO, every Monday morning you were asked to report on your work status. That was as good as it got for a design process. Unfortunately, the customers of the department—the power plants—weren't very happy with the quality, consistency, cost, and schedule performance of the engineering process.

At a consumer products company, pictures of their five-step product development process were posted in all of the hallways. When asked about the process, the scientists and engineers either didn't know about it or would freely state that nobody followed that process. After we trained several teams at the company's South American R&D center in the Design for Six Sigma method, one of the team leaders came up at the end of the session. She thanked us for the training and said, "Before you came, our development process was just a series of gates. You've shown us how to successfully get from one gate to another!"

Tool Issues (The Fishbone's Machine Category)

Companies employ poor or underperforming design tools. A lot of great tools can really help improve the development process. We've seen companies go through the waves of quality function deployment, Taguchi, concurrent engineering, stage gating, reliability, and so on. It often seems, though, that the main effect of introducing these tools is to add buzz words and acronyms to the development lexicon. As one of our friends noted, "At the fountain of knowledge, most people simply gargle and spit!"

Let's consider quality function deployment (QFD) as an example. Originally developed in Japan in the 1970s, QFD was introduced to U.S. companies by thought leaders such as Bob King and Don Clausing. Japanese companies had shown them how QFD could lead to faster, better product launches. So here's a tool that actually works, one that has been demonstrated to provide a measurable benefit to product development.

In typical applications, though, the gargle and spit approach often occurs. Here's a sequence of events we've seen all too many times. QFD is sold as a new idea to a company. A QFD consultant is hired and trains the R&D staff. The contract might include mentoring one or two teams in how to develop QFD's house of quality. Once the first couple of houses are built, the company will proclaim, "Yes, we know how to do this." The consultant is patted on the back, collects the fee, and goes off to the next client.

If you were to visit this company a year later, though, we'll bet one of two scenarios will have occurred. In the first, the company is back doing product development the old way. Some associates will remember (perhaps not so fondly) building the *house of pain,* and their typical comments will be "took too long, didn't see the benefit, our old way was good enough." If you wandered through the R&D offices, you'd see the QFD training books on the shelves, maybe a few computers with QFD software still installed and, perhaps, that lone QFD champion's voice still crying in the wilderness, but not much else.

In the second scenario, QFD is still employed, but it has become one of the checklist items in the development process. We walked into a development team room a while back and were pleasantly surprised to see a quite detailed house of quality on the wall. The big "oops" came when we asked an engineer what they did with the house. "We showed it at our last stage gate and management was happy. Now we are developing the product that we think the customers want!" As another of our friends says, "Well, that looks like some pretty expensive wallpaper to me!"

Now we happen to think that QFD is a great development tool. But if we are going to use the tool, let's make up our minds to drink deeply and not just gargle and spit.

One of the other failure modes we've seen is a focus mainly on the mechanics of the tool. In the QFD example just described, notice that we mentioned only R&D being trained in the tool. We often forget that it takes a *team* working together to deliver the best product for the customer. For QFD, the house of quality is the tangible deliverable. The truly important outcome of QFD, though, is the *alignment* that the team achieves around the requirements for the new product. If the meetings to "build" the house of quality do not include all key team members and do not incorporate building consensus around the requirements, the tool has failed to accomplish its purpose. The QFD house should, as a minimum, include participation from market research, R&D, and operations. We can't overstress the importance of these "soft" issues associated with many DFSS tools and methods.

People Issues

We'll try to keep this one short, especially because we'll talk much more about it later in this book. You probably know most of these issues in your company, anyway.

Silo mentality. Marketing, R&D, and operations are the usual trinity we have to address. Misalignment of goals, mistrust, and a throw-it-over-the-wall culture are just some of the issues we deal with. One R&D director tells the story of how she was asked to look into a development project that just wasn't going anywhere. She talked to R&D first and asked what the project was about. The answer: "Oh, we are developing a new adhesive technology." Next, she visited marketing. The answer: "This project is about a new platform of 'X' that we want to launch." Finally, she talked with operations. The answer: "Well, we are looking to really increase production efficiencies with this project." Three groups, three different goals for the same project!

Reward and recognition. In one company, marketing takes the lead on product development projects. Unfortunately, individuals are rotated into these positions for periods of only 18 to 24 months and they are rewarded mainly if they launch a product. The development cycle is constrained to the time they spend in that role. In 18 months, this company's development process is capable of delivering a line extension, but not much else. The company wonders why most of its development projects are low risk, low reward!

Leadership. We are not going to rant and rave here that your leaders need to be like Jack Welch, Larry Bossidy, or Attila the Hun, but this is an obvious, important development process factor (we've devoted the entirety of Chapter 6 to this factor).

Two common leadership themes seem to appear over and over, though. The first concerns prioritization. Too many development projects are thrown at finite development resources. Talking with a package designer the other day, we asked her how many projects she's working on. The answer was15! Do the math—even working a typical 60-hour week, that's about 4 hours per project per week. This company needs to learn about the Theory of Constraints and apply that learning to its development process's bottlenecks and throughput!

Second, we've found that the risk tolerance of many companies is usually pretty low. One of our friends did a risk/reward chart for his firm's development projects (Figure 1.1). Virtually all of the projects were in the lower left quadrant, not good news, because the company had some pretty aggressive growth targets.

In just 15 minutes a day, your development process can look like GE's, Johnson & Johnson's, or . . . (fill in the blank!)

This is the part of the introduction where we unveil the answer to all your product development woes. Well, not quite. Here's the best news we can give you.

Over about the last 10 years, companies around the world from GE to Johnson & Johnson have been improving their development processes using a framework known as DFSS.

Light bulbs, x-ray tubes, automobiles, appliances, medical devices, shampoo, soft drinks, and even frozen dinners are being designed faster, with a focus on what the customers truly want and with a view to preventing defects from occurring as far upstream in the development process as possible (no more throwing it over the wall). DFSS methods and tools have also been successfully applied to designing and redesigning business processes and services.

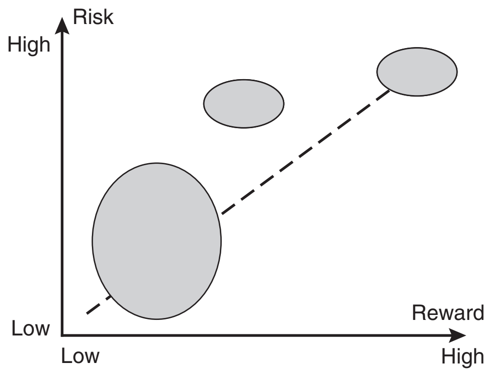

Figure 1.1 Risk/reward profile.

Before we get carried away, though, we'll make two disclaimers. First, we recognize that product development is a key contributor to organizational success. However, we know of companies that are successful in spite of their poor product development processes and vice versa. DFSS by itself will not guarantee a rapid rise in your stock price. If some of your other processes and structures are in poor shape, work on improving those, too.

Our second disclaimer is that time moves on. We'll share DFSS success stories from a number of companies. However, remember that Rome had both a rise and a fall. While we'll discuss sustaining the DFSS drive, we haven't been able to figure out how to make DFSS a permanent part of an organization's systems, structures, and culture.

Given these disclaimers, though, we've found that for the companies who have taken DFSS seriously, their results are positive and encouraging. For example, one company has seen its initial scrap/rework rates dropping from historical averages of 10 to 15% to 2 to 3% for products developed using a DFSS process. Another has seen its customers' satisfaction with new products increase dramatically. Following is a specific example from General Electric.

The Product: LightSpeed Computed Tomography (CT) Scanner (launched 1998)

The DFSS Process: GE incorporated input from more than 100 customers around the world to make sure the technology was meaningful, not just new.

The LightSpeed Scanner's advancements were the direct result of GE's Design for Six Sigma quality efforts, an approach that enabled GE scientists and engineers to develop more benefits and introduce the scanner at least one year earlier than what otherwise would have been possible. Specifically, this approach allows GE to develop nearly flawless products through a disciplined process control approach that is delivering tremendous results.

Six Sigma allowed GE to manage this complex technology—including R&D breakthroughs in materials science, computers, software, and electronics—to achieve a system with unprecedented speed and image clarity (according to PressLink Online AP Photo Express Network, PRNZ).

(Continued)

(Continued)

The Customer Response: "The speed is breathtaking," said Dr. Carl Ravin, professor and chairman of the Department of Radiology at Duke Medical Center. "A body scan that used to take three minutes now can be completed in approximately 20 seconds."

A GE Manager Response: "In the past, many projects were approved and implemented based on gut feeling and intuition. Designing a process to deliver 6σ capabilities requires quantitative data. Using gut feeling in a Design for Six Sigma methodology will not permit passing a tollgate."

SO HOW DOES DFSS WORK ITS MAGIC?

There's both a narrow and a broad answer to this question. Let's start with the narrow answer. The essence of DFSS is to develop a design where important product requirements are being delivered at Six Sigma levels of performance. Here, you will start by identifying the product's critical-to-quality (CTQ) requirements. These CTQs are based on what is most important about the product from the customers' perspective. Then, as you develop the product and production process, you'll deploy and allocate these requirements (this is called *flowdown*) to lower level (for example, subsystem, assembly, parts, process) requirements. You'll then predict how the variation in parts and process and noise variables will affect the variation the customer sees at the CTQ level. You'll use design scorecards to report the predicted design performance. You'll cascade requirements and establish transfer functions relating the design Xs to the CTQs. You'll perform Monte Carlo simulations to determine how variation in the Xs contributes to variation in the CTQs (recording the results on your scorecard). Just incorporating these DFSS elements in your design process can help. At the very least, you'll have checked your design against the variation enemy and made design improvements where insufficient margin to CTQ specification limits exist. Your product can then enter manufacturing with some degree of assurance that defect probabilities will be low.

Over the last few years, a broader view of DFSS has evolved. A number of companies view DFSS as encompassing the entire product development process and are using DFSS to improve their overall design process. Here, DFSS's scope has been expanded to address all product development functions from market research (voice of customer, QFD) to concept selection (TRIZ, Pugh methods) to detailed design (design cascade, design prediction, reliability, design for manufacturing/assembly) and production system design (process control, statistical process control [SPC], error-proofing, and lean). Some companies have also included advanced technology development and product portfolio management under the DFSS banner. In this broad view, DFSS aims to help you:

- Identify, prioritize, and resource the right product development projects.
- Manage advanced technology development outside of/in parallel with product development.
- Understand critical-to-quality customer requirements (the Ys).
- Develop superior design concepts to meet the CTQs.
- Predict design quality and eliminate defects upstream.
- Manage risks through failure mode and effects analysis.
- Identify and optimize critical process variables through simulations and design of experiments.
- Benchmark to demonstrate design superiority.

While these are noble and lofty goals, as a product development manager you face two questions—first, what to do, and second, how to do it. It's nice to talk about getting your VOC and using QFD to define your CTQs, which are then cascaded to CTPs through DOE and FMEAs, and then optimized and controlled through SPC and Poka-Yoke error-proofing (oops, sorry, we slipped into Buzzword City). That's your future world.

Today, though, you are faced with the current state of your product development process and staff. Your development staff today may do little in the way of customer research, likely defines product features as requirements, performs the failure modes and effects analysis (if at all) just before the next design review to make sure the box is checked, and thinks Poka-Yoke is something you do with chicken eggs.

Everybody knows they need to manage their diet and exercise, but there is no easy way to get there; no magic pill. The tough thing is changing habits. This same challenge applies to improving your product development process. Successful change will encompass personal, team, and organizational habits.

WHAT WE'LL PROMISE YOU

In developing this book, we actually followed the DFSS and Six Sigma process—including gathering the voice of our customers. Here's what our research told us you wanted:

DFSS Business Case

Building the Business Case for DFSS. What will the business achieve with DFSS? What are the benefits/costs? (chapters 3 and 5)

Success Examples from Others (Outside/Inside)—Value. What have others achieved? What can we expect to see (early wins, longer-term wins)? How can we share within our organization? (chapters 3, 6, 7, and 8)

Leadership

Leadership Linkage. How can senior leadership be sold on DFSS? What behaviors are expected of leaders? What can they expect of their organization? (chapters 5 and 6)

Commitment. How can we get the organization committed to adopting DFSS? (Chapter 5)

DFSS—What, How, and When

Emphasis on Process, Methods, and Application. Focus on the DFSS process (not details around specific tools—those details can be obtained elsewhere). (Chapter 4)

Specific Tools—Where, When, and When Not. Show where tools are useful/not useful; relevant examples of tools application within the DFSS process. (chapters 4 and 11)

Minimum Requirements and Expectations. What are the minimum expectations for development teams? (chapters 6 and 8)

Application/Impact of DFSS Method/Tools on Day-to-Day Design. How can we ensure that the adopters see

DFSS as benefiting the design process, not as extra work? How can we ensure that application occurs after the training? How do we keep the organization focused on DFSS? (Chapter 6)

Application in My World. How are development efforts currently performed? Where are the inefficiencies? How can DFSS make product development (in my world) better, faster, and cheaper? How can current design best practices be integrated with DFSS? (Chapter 5)

Chartering. How do we ensure that the right development projects are selected? How can we keep the DFSS momentum going beyond the charter? (chapters 4, 7–10)

Planning. How do we develop a vision of a DFSS organization? How can we introduce DFSS? What are effective models for deploying DFSS in different design organizations? What specific behaviors should leaders exhibit during implementation? What should the deployment plan look like? (chapters 5 and 6)

Deployment Strategy. What deployment strategies are effective for different types of design organizations? Give me tips for successful rollout strategies. What are best practices for training for teams? Who are the key DFSS stakeholders, and how do we engage them? (chapters 5 and 6)

Outlining Deployment Strategies and Drivers. What are the key deployment drivers? What messages should management hear? (chapters 6 and 8)

Cost of Implementation. What will it cost to implement DFSS? (Chapter 6)

Change Management. How will we lead the organization through the change (transformational leadership)? How will we avoid preaching DFSS as gospel and engage staff in a scientific, logical manner? How can senior leadership learn to think DFSS and talk DFSS with teams, creating expectations for application of key DFSS methods/tools? (chapters 5 and 6)

Doing. What does *good* look like for DFSS? How can we make the leadership issue tangible for leaders—specific behaviors to look for, specific questions to ask? What resources will be required? How can we make it part of doing business, not something special or extra? (chapters 4, 5, 6, and 11)

Design Process Integration. How can we integrate DFSS into the current design and development process (or vice versa)? (Chapter 6)

Integration with Existing Design Process. Think about implementing DFSS as a Six Sigma project. How do we ensure that DFSS doesn't slow down design/development? (chapters 5–8)

Infrastructure. What infrastructure will be needed to support DFSS (training is obvious; others include databases, references, websites, software, and rewards and recognition)? (Chapter 6)

Roles and Responsibilities. What are the specific roles and responsibilities? What are the expectations for each role deploying and doing DFSS? (Chapter 6)

Role of Training. How does learning work in a specific organization? Should we use internal or external trainers? How do we link training to action? How should training be applied—key projects versus the masses? (Chapter 6)

Consulting and Support. What is the role of the consultant? How should he or she work with the deployment team? Are dedicated resources (for example, DFSS Master Black Belts) required? How can they be best employed? (Chapter 6)

Subject Matter Experts (SME). What SMEs are needed? How can they help the deployment process? What coaching models work and don't work? (Chapter 6)

Mentorship. What mentorship models work? (Chapter 6)

Check and Act. What assessment models work? How can success be measured? (chapters 5, 6, 7 and 8)

Metrics. What are examples of good metrics for the design process? How should metrics evolve? How should behaviors be reinforced? (Chapter 7)

Sharing and Learning. What sharing models work? How can teams share learnings? How can they avoid the same struggles? (chapters 5 and 6)

Path to DFSS and Assessment Tool. How can DFSS maturity be assessed? (chapters 5, 6, and 7)

Pitfalls. What are the pitfalls of implementing DFSS? How can these be avoided? (Chapter 6)

DO YOU HAVE THE GOTTAWANNA?

Many of the organizations we have worked with have a common cultural barrier to DFSS—they are already successful. Nothing stifles success like success. But fortunately for many of these organizations, they also shared one other major cultural element: Their leaders were not satisfied with "same as" success. They wanted to give their customers products that actually met and exceeded the customers' needs. They wanted to stop wasting resources "cleaning up the mess" of a barely producible product. They wanted to be leaner, faster, and more profitable every year. We will show you how DFSS has helped these organizations and how it will help you, too, if you have the *gottawanna*.

If you are ready, let's get on the field and start our way to DFSS victory!

2

What Is Design for Six Sigma?

INTRODUCTION

Before we can help you implement Design for Six Sigma, let's spend a few pages describing what DFSS is. In case you aren't familiar with Six Sigma, we'll start with some basic concepts.

Our first exposure to Design for Six Sigma was as an improvement to an existing design process. Over the last few years, though, DFSS's scope has expanded to include virtually the entire product development life cycle. We'll bracket this by first describing DFSS in its narrow sense and then expanding to the bigger picture of today's DFSS.

WHAT IS SIX SIGMA?

If you are already familiar with Six Sigma concepts, you can skip this part and head on down to "What Is DFSS—Narrow Sense?" section. On the other hand, this won't take you too long to read and may provide insight into some of the DFSS method and implementation discussions later.

Six Sigma has come to mean a lot of things. As an *improvement system,* Six Sigma has been around since the late 1980s. Motorola was the first company to develop and apply Six Sigma as a disciplined, breakthrough improvement method. General Electric then adopted Six Sigma and popularized the method through both its Six Sigma results and the public spotlight that Jack Welch put on its Six Sigma application. Most large companies have at least tried Six Sigma, with varying degrees of adoption and success. Although the idea of Six Sigma is less than 20 years old, the tools and methods associated with Six Sigma are not new. Many of these, especially the statistical and process analysis methods, were around for quite a few years before the phrase *Six Sigma* popped up.

Six Sigma *performance* corresponds to a certain defect rate—about 3.4 defects per million opportunities. Where did this come from? Well, the basic idea behind Six Sigma is simple. If your process can produce parts or products whose characteristics have very little variation in relation to the customers' specification limits, then you will produce very few defects. The key word is *variation*. Most of us are comfortable with using the *mean* as a measure of average performance. Fewer are comfortable with measuring variation in terms of a *standard deviation,* or *s.*

Simply speaking, the *Sigma level* of a process measures how far the mean of your process is from the specification limits *measured in standard deviation units.* Figure 2.1 shows you the formula for the mean and standard deviation and also shows a process whose mean is four standard deviations from the specification limit (that is, a *Four Sigma level* process).

For example, say your customers expect you to produce a part that has a requirement that is of particular importance to them. We'll call that requirement "critical to quality" or CTQ. The customers want you to hit a target of 100 and stay below 105 for the part's CTQ. You make 50 or so parts and calculate your process's mean to be 100. Good news! Your process is centered on the target. However, if you calculate the standard deviation of the parts to be 5, you should be worried about the variation in your process. A little statistical theory, invoking the normal distribution, will predict that your process will produce 16 defective parts out of 100 opportunities. Figure 2.2 shows you the picture of this situation.

Because this process's mean is one standard deviation away from the specification limit of 105, we'd conclude that this process

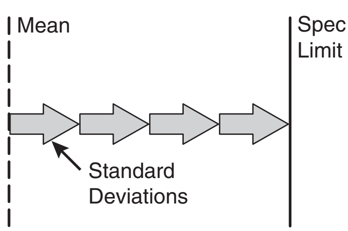

Figure 2.1 Mean, standard deviation formula, Sigma-level picture.

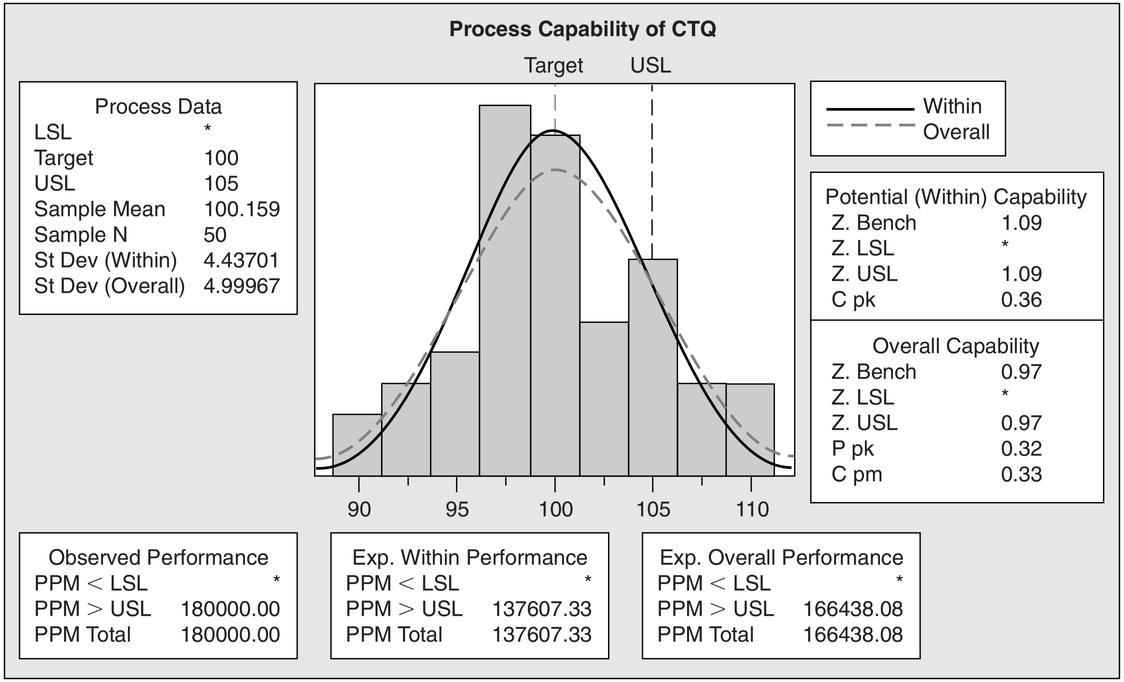

Figure 2.2 Process capability with mean = 100, sigma = 5, target = 100, USL = 105.

is performing at One Sigma. Be careful; the language can be confusing. When you hear the term *One Sigma process,* that's not the value of the standard deviation, but rather how many standard deviations the process' mean is away from the spec limit(s). To keep from confusing you, we will use the term *Sigma level* when we're describing product, part, or process, performance, and *Sigma* when talking about the standard deviation. In our example, Sigma is 5, but the Sigma level is 1.

In Figure 2.2, notice the Z term in the upper right box labeled *Potential (Within) Capability*. Z is the symbol used to describe Sigma level. You will hear these two terms used interchangeably in the Six Sigma world. You will hear "Z equals 6," or "Z value is 6," "Sigma level is 6," or "the process is running at a Six Sigma level." All of these mean the same thing.

Figure 2.3 shows a process that is operating at a Six Sigma level. As you can see, when there are six standard deviations between the mean and the specification limits, there is a very, very small chance that a given part will be produced outside of the spec limits. Even if this process shifts and drifts over time, it can be expected to produce no more than 3.4 defects in one million parts or opportunities.

So why is Six Sigma good? Why not Five Sigma? If Six is good, why isn't Seven Sigma better? First, here's a simple answer. When Motorola benchmarked world-class performance in the 1980s, the best-in-class Japanese electronics companies were operating at about a Six Sigma level. Why were *they* operating at Six Sigma? The deeper answer lies in the economics of production. The best-in-class Japanese companies were minimizing the loss associated with their products' variation; in other words, they were following Genichi Taguchi's quality approach.

Don Wheeler elaborates on the economics of Six Sigma in his paper "The Six Sigma Zone." When we wrote this book a downloadable copy was available at his website, www.spcpress.com. His argument is similar to Taguchi's quality loss concept and is based on the *excess cost of production.* These excess costs are incurred when the product or part is not produced on target. If you can achieve operational process capabilities (Cpk) in the 1.5 to 2 range, you will minimize these excess costs, which will likely either increase your profits or allow you to reduce your price in the face of competition. If you multiply Cpk by 3, you'll approximate the Sigma level or Z of the process. Hence, a Cpk of 2.0 is equivalent to a Six Sigma level process. Don't be fooled by fuzzy arguments for why you should pursue Six Sigma; there is sound *economics* behind this important business concept!

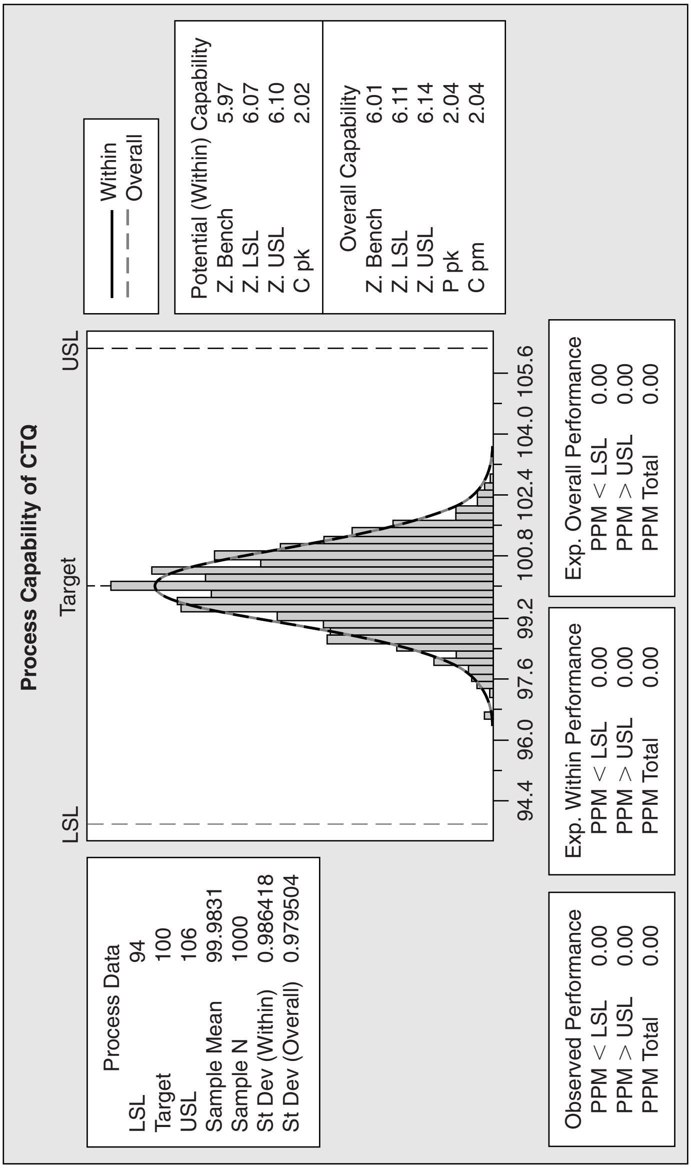

Figure 2.3 Six Sigma process.

So with Six Sigma being the goal, and since most companies find they operate at somewhere between Two and Four Sigma, how do you get there? To help them achieve Six Sigma, Motorola developed a four-step methodology: *Measure, Analyze, Improve, Control (MAIC).* The continual application of MAIC helps Motorola identify and eliminate sources of variation in its processes.

When GE adopted Six Sigma, its management added a *Define* step, which identifies the current process and clarifies what the customer cares about: the process' CTQs. GE found that this prestep was helpful when applying MAIC to *transactional* rather than *manufacturing* processes. This leads us to today's DMAIC improvement method. You'll see variations of DMAIC. For example, one business is concerned with *replicating* improvements, so it employs DMAIC-R (R = Replicate). We've found that *DMAIEC*—splitting the Improve step into two parts: Identify Improvements and Execute Improvement—is useful in the kind of improvement projects we work with. We'll use DMAIEC here:

Define—The problem is defined and the business value of the problem quantified; resources are marshaled; the customers, their needs, and the critical-to-quality metric operationally are defined; and finally the current process is defined.

Measure—The current process performance is measured, the baseline Sigma level calculated, and the process sliced and diced to see whether some leverage area exists (Pareto analysis for defects, theory of constraints for bottlenecks, etc.).

Analyze—The process variables responsible for performance are identified (if the CTQ is our outcome or *y-variable,* then the Analyze step represents the search for the inputs or *x-variables*) through a detailed process analysis, a statistical analysis, or a combination of both.

Identify improvements—Countermeasures to improve and stabilize the Xs are identified, practical methods to implement the countermeasure(s) identified, cost-benefit tradeoff studies performed, and the process changes piloted (with data-based results backing up the countermeasures' effectiveness).

Execute improvement—The full-scale change is implemented with the necessary process changes being executed across the department or organization. Results of the full-scale change are measured to ensure its effectiveness.

Control—The Control step recognizes that effective changes need to be made a permanent part of the process. A process control system is developed and implemented, identifying who is accountable for process performance, what the standard process is, how it should be measured, and what actions are to be taken if the process is not performing as desired.

Figure 2.4 shows the steps and the key activities performed in each.

The Six Sigma toolkit today resembles a well-stocked hardware store. Here is a list of tools from a four-week course delivered by one of the authors:

MS Office (including Project and Visio)

Statistical software (Minitab, JMP)

Black Belt roles and responsibilities

Improving existing products and services—DMAIEC

Project chartering

Project reporting and reviews

Voice of customer feedback

Developing indicators

Basic data collection

Measurement system analysis

Line graphs

Run charts

Sampling

Histograms

Process capability

Process flowcharts

Process analysis methods

Bar charts

Pareto analysis

Pie charts

Radar charts

Cause and effect analysis

Contingency analysis

Scatter diagrams

Correlation analysis

Regression analysis—simple, linear

Probability distributions

Hypothesis testing

Parameter estimation

Confidence intervals

- Sampling
- Single-factor experiments
- Analysis of variation (ANOVA)
- Design of experiments
- Reliability terms and definitions
- Reliability management
- Failure modes and effects analysis (FMEA)
- Fault-tree analysis
- Weibull analysis
- Selecting and implementing process changes
- Cost-benefit analysis
- Evaluating the effects of changes/standardization and replication
- Controlling processes
- Process management charts
- Control charts
- Creativity methods
- Performance and process benchmarking
- Tolerance development and analysis
- Reliability testing/accelerated testing

The people (usually future leaders) who receive this training and lead improvement projects are known as Black Belts. Black Belts are often assigned full-time to solve important business problems or attack business opportunities. Green Belts will work part-time on smaller business problems/opportunities. Master Black Belts actively support business improvements and mentor Black and Green Belts.

Other Six Sigma roles include Champions and Sponsors, who can provide resources and remove barriers for the improvement teams, and Finance (or Money Belts) to ensure the projects make business sense and to provide their blessing to recorded project benefits, especially in today's Sarbanes-Oxley world.

To help the organization implement Six Sigma, drive its application, and assess Six Sigma's effects, a Six Sigma Program Office may be established and there may be Six Sigma leads assigned within the business' various departments/divisions.

Executives are accountable for promoting the application of Six Sigma in their divisions, ensuring that the rewards and recognition systems are consistent with the behaviors Six Sigma demands, and, most important, ensuring that the business strategy is defined, measured, and accomplished by Six Sigma projects (where needed).

The previous text describes a basic Six Sigma initiative. There are as many flavors as there are companies that have adopted Six Sigma. For example, Six Sigma and lean improvement methods are

Define	Measure	Analyze	Identify	Execute	Control
Charter the project Identify stakeholders Select team Plan the project Define the current process Obtain customer CTQs	Gather initial metrics Determine current “sigma” Address “low-hanging fruit” Stratify data Determine initial value proposition	Develop cause and effect hypotheses Gather causal data Determine and validate root causes (Xs)	Identify counter-measures Select practical approaches Design future state Predict new “sigma” Perform C/B and risk analysis	Develop control methods Develop dashboards and scorecards Train Execute Measure results Manage change	Report dashboard and scorecard data Create feedback loop and adjust process Identify replication opportunities Develop future plans

Figure 2.4 DMAIEC process improvement method.

often linked as companies strive to take waste out of their supply chains and become more responsive to the market. In the software world, we've seen Six Sigma used to support improvement of software development processes. At least one company has integrated Six Sigma with its goal of achieving Level 5 on the Software Engineering Institute's capability maturity model-integrated (CMMI).

While it sounds obvious, many companies miss the connection between Six Sigma and achievement of *strategic goals.* Strategy often comes down to either improving existing processes' performance or designing new products/services/processes. Six Sigma's methods are perfectly suited for this.

Although application of Six Sigma tools and methods will *lead* to reduced costs (less waste, rework, warranty), some companies have corrupted Six Sigma by treating it primarily as a cost-reduction program. If your Six Sigma program is focused only on cost reduction, you can expect that it will only last for about a year before it disappears!

A thriving basic Six Sigma culture will lay the foundation for your efforts to promote Design for Six Sigma. To the extent that one or more elements are missing or not working well in your company, your DFSS implementation will have to compensate. Three simple examples will suffice:

1. If manufacturing has embraced Six Sigma, you will be able to enlist the staff's support for your DFSS effort. They are probably tired of having to perform DMAIC projects because of the defects your current design process is producing.

2. If your people have already been trained in the Six Sigma toolkit, you'll find that quite a few of these are also part of the DFSS toolkit. This can reduce the learning curve and the amount of time you have to devote to training and skill-building. Consider, though, which functions have these skill sets. Often, manufacturing and operations have been trained in and are skilled in Six Sigma, but not the research and development staff.

3. If your company has tried Six Sigma and discarded it, then you are in for an even greater challenge trying to implement DFSS. You might be able to pull off a design process improvement initiative, but you may be better off not calling it DFSS.

THE IDEA OF IMPROVING DESIGN PROCESSES

OK, let's move on to Design for Six Sigma. It may be hard to believe, but occasionally we run across individuals who do not accept the idea of design as a process. Just in case you have doubts, we'll try to dispel those with the following example.

In 1988, one of us (John) was working at Florida Power & Light (FPL) in a statistical specialist role (today, we'd call him a Master Black Belt). At the time, FPL was learning and applying its quality improvement program to business problems/opportunities.

FPL's Japanese counselors had spent almost three years helping us learn how to improve our business processes through systematic process and root-cause analysis. It had been a long road. We weren't dumb, but it took us a year to just get good at doing Pareto and cause/effect analysis in our counselors' eyes. Just as we started to think we could DMAIEC our way to business excellence, one of the counselors remarked, "You must now learn to move from preventing the recurrence of problems to preventing problems through prediction!" You could see the FPL eyes rolling: "Oh, boy, here we go again."

The counselor had an important point, though. He was telling us to start looking at the processes we employed to *design* power plants and transmission and distribution systems. He was asking us to identify and eliminate defects during design *before* they were translated into steel, copper, and concrete.

For example, counselor Dr. Noriaki Kano reviewed our plans to add two gas turbines to one of our power plants. He asked Bill Hensler, the plant manager, for his *quality plan*. Of course, we didn't have a clue what he was talking about. It turned out Dr. Kano was simply looking for evidence that we knew what the Ys (outcomes) were for this project and that we had identified *and were managing* the critical inputs, or Xs, that would drive achievement of these outcomes. So, with their encouragement, we dug in and found that there were lots of opportunities in our design processes.

We supposedly cared a lot about our system's reliability. But do you think we demanded reliability from our equipment suppliers? Not really. How about doing a simple failure modes and effects analysis for a new transmission system design? No, too much trouble—we don't have time for that! How about keeping the drawings updated so the engineers could work with a good set of as-builts? Again, some folks didn't quite see the value in that. We were in a meeting once where the drawing manager actually stated, "Engineers don't need updated drawings!"

So as we started working on these opportunities, we had a couple of novel thoughts. First, design *is* a process. It has inputs, a *product* is produced (the design output), and some defined steps have to occur between the input and the output. Second, by treating design as a process, we could identify the Xs in the process that were producing design defects and work to eliminate those defects. We also found that the same systematic process/root-cause analysis approach we were applying to power plant failures could be used to analyze and improve our design processes. At one point in the late 1980s, we had over 60 teams working on design process improvements.

WHAT IS DFSS: NARROW SENSE?

So, now we'll address the Design for Six Sigma process. Suppose you are the head of a Stone Age research and development organization. You are working on a new idea—combining the wheel and axle to transport objects. Imagine that you're tired of dragging those big blocks needed to build Stonehenge. There is a large market for these, and you plan to build an assembly line to make the product. Fred will be in charge of boring the hole through the wheels and Barney will shape the axles' outside diameters. Wilma and Betty will assemble the two components (Figure 2.5).

Let's apply DFSS to this situation. There is only one requirement you are worried about—the gap between the axle and the wheel. There is an upper specification of 0.005 feet; if it's greater than this, the wheel will slip off the axle. There is also a lower specification of

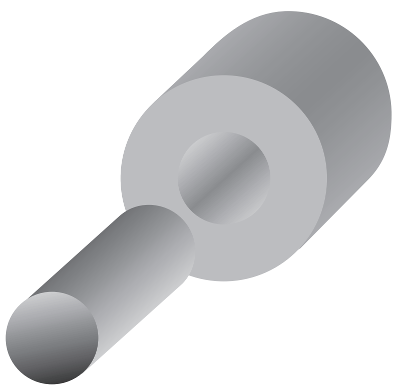

Figure 2.5 Wheel/axle assembly.

0.00 feet; otherwise there will be an interference and Wilma and Betty won't be able to assemble the two parts. Figure 2.6 shows our specifications.

This critical requirement, a CTQ, is a Y (outcome) that you can't control directly. To control the CTQ, you need to identify the Xs (variables or factors) that drive the CTQ and how they combine to achieve the Y; we'll call this a *design cascade.* Here, the only two Xs are the inner diameter of the wheel (Fred's responsibility) and the outer diameter of the axle (Barney's responsibility). We'll examine a simple relationship between the Xs and the Y:

$$Y = f(x_1, x_2)$$
$$\text{Gap} = \text{Wheel diameter} - \text{Axle diameter}$$

The formula is known as a *transfer function,* because it helps us *transfer* both the mean and variation from the Xs to the Y.

Suppose the axle diameter has been set at 0.400 feet. If you wanted to achieve an average gap of 0.0025 feet, the wheel diameter must be set at 0.4025 feet. Entering these average values into the transfer functions confirms that the targeted gap will be achieved.

But let's not forget the specification limits. Because Fred and Barney will be making a lot of these shafts and axles, we can expect some variation to occur in the part dimensions. The question that we now face is how much variation is allowable in the parts—or, equivalently, what specification limits should we place on the part dimensions? We really can't answer the question until we've posed another interesting question: How often are we willing to fail at achieving the CTQ? Or, asking the question in just a slightly different way, What is our target Sigma level for this CTQ? If we set the target Sigma level at 6, we already know that fewer than 3.4 out of a million axle assemblies will be defective. If that seems to be too

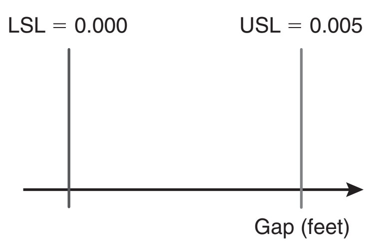

Figure 2.6 Axle/wheel specifications.

Table 2.1 Sigma level and corresponding defect rate.

Sigma level (Z)	Defects/million opportunities (DPMO)
2	308,537
3	66,807
4	6,210
5	233
6	3.4

good, you may decide to choose a lower target Sigma level, by looking at Table 2.1.

Remember that this economic decision is not just based on the cost of defects; there is an *excess cost of production* associated with any deviation from the target. Let's go with a targeted Sigma level of six and see where it takes us. To achieve this at the assembly level, the parts' and assembly process' allowable variation (spec limits) must be defined, consistent with the CTQ requirement. This raises yet another question: How can we transmit variation from a Y to its associated Xs?

For an assembly such as our axle and shaft, we can use a simple but important statistical property known as *additivity of variances.* If you know the standard deviations of the inputs to the assembly, and the *transfer function* between the Y and its Xs is linear, the sum of the squares of the Xs' standard deviations equals the square of the Y's standard deviation:

$$s_Y^2 = s_{X_1}^2 + s_{X_2}^2 + \ldots s_{X_n}^2$$

Where S_y = Standard deviation of the output, Y

S_{xi} = Standard deviation of the inputs, $x_1, x_2, \ldots x_n$

If you'll permit us a little magic at this point, we can also use this relationship to define the allowable tolerances of the Xs to achieve a required Y tolerance:

$$T_Y^2 = T_{X_1}^2 + T_{X_2}^2 + \ldots T_{X_n}^2$$

Where T_y = Total tolerance of the output, Y

T_{xi} = Tolerance allocated to each of the inputs, $x_1, x_2, \ldots x_n$

For the axle-shaft assembly, the tolerance width is 0.005 feet (USL = 0.005 feet, LSL = 0.000 feet). The transfer function from the gap to the axle and shaft diameters is linear, so we can use the additivity of variance property:

$$T_{Gap}^2 = 0.005^2\ feet^2 = T_{Axle}^2 + T_{Shaft}^2$$

Suppose Fred and Barney use similar fabrication processes to make the axle and shaft. We decide to equally allocate (or budget) the allowable tolerance to them:

$$T_{Gap}^2 = 0.005^2\ feet^2 = 2 \times T_{Component}^2$$

or

$$T_{Component}^2 = T_{Gap}^2 / 2$$

$$T_{Component} = \sqrt{T_{Gap}^2 / 2} = \sqrt{0.005^2 / 2} = 0.00354\ feet$$

With a resulting tolerance half-width of:

$$0.00354 / 2 = 0.00177\ feet$$

Of course, Fred and Barney have to hold these component tolerances at a Six Sigma level in order for the assembly to be produced at Six Sigma level. Note that we're assuming Wilma and Betty's assembly process doesn't introduce any variation, which is generally not the case in the real world.

Applying these specifications to our suppliers is all well and good, but how do we know they can meet the component requirements (for Fred's wheels: 0.4025 +/− 0.00177 feet at Six Sigma Level; for Barney's axles: 0.4000 +/− 0.00177 feet at Six Sigma Level)? Because this is a Stone Age example (pre-computers), we'd probably have them make 30–50 samples, measure the results, and perform a capability calculation. By the way, we had better check Fred and Barney's measurement systems—these will introduce additional variation into the mix.

In today's Silicon Age we will use a Monte Carlo simulation to predict the design's performance before we actually cut tools and build parts. The simulation will allow us to predict how well we expect to achieve the CTQ's performance as a function of the design and assembly Xs, identify the key contributors to CTQ variation, and then, if necessary, trade off variation across suppliers or change how variation transmits from the suppliers to the CTQ (for example, improve the robustness of our design). Figure 2.7 shows a typical variation simulation, using the Crystal Ball Monte Carlo package.

Once we're happy with the simulated design's results, we'll order tools and verify that the design meets its predicted performance.

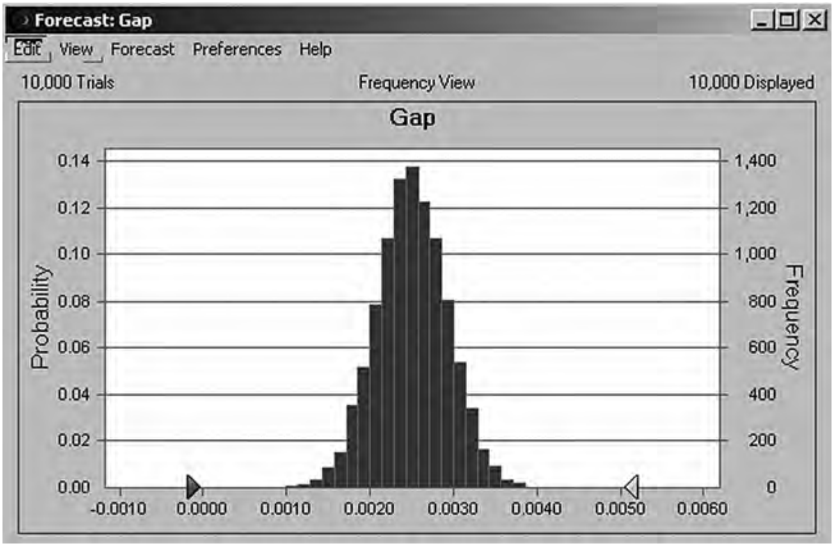

Figure 2.7 Monte Carlo simulation of gap CTQ.

Real parts will be assembled and the Sigma level of the CTQs actually measured.

So that's Design for Six Sigma (narrow sense) in a nutshell. Let's summarize some of the key ideas:

CTQ definition—DFSS depends on clearly defined product requirements that can be measured. The complete CTQ will generally include a product function, a metric, targets and limits as appropriate, and a Sigma level target.

Cascade—Once the CTQs (Ys) are defined, during design we will systematically identify the important Xs that drive achievement of the CTQ. Cascade includes both a qualitative understanding of the Ys and Xs and, through the transfer function, a quantitative understanding of the Y = f(X) relationship.

Tolerance allocation—For complex systems, as the design work progresses, the design team will define lower-level (that is, subsystem) requirements in terms of both a target and associated allowable variation (tolerance band plus targeted Sigma level).

Design prediction—To ensure the design meets its CTQ goals, the team will predict how the variation in parts and assembly processes will transmit to the variation that the customer sees at the CTQ level.

Trade-off and design robustness—If the predicted performance is less than adequate, the design is iterated, either through tolerance tradeoffs across suppliers or through the robustness of the design to sources of variation it will experience.

What benefits do you see from this approach? How much of this are you already doing? For example, many design teams we work with do stack-ups of dimensional tolerances. But they often stop the stack-up short of the CTQ level or perform it for other reasons—for example, analyzing assembly interferences. As you begin your DFSS journey, identify these supporting practices and use them to help get the DFSS message across to your developers.

WHAT IS DFSS: BROAD SENSE?

In the previous example, we started with a known product, with known CTQs and a given design. We also assumed that there was a market for the product, willing to pay enough for the product for us to cover our costs and make some amount of profit.

You can see where we're going with this. If we design a Six Sigma product but don't really know the right CTQs, we'll have a high-quality product that is just sitting on the shelf, because it doesn't fulfill customers' needs. If there is little market for our Six Sigma product, again, that's bad news. Our design concept may be flawed in that it is not robust to sources of variation. Here, no matter how tight we squeeze our suppliers' specifications, we will fail to achieve Six Sigma at the CTQ level (or we will do so at excessive cost). Our goal, in fact, should be to develop a design that is tolerant of variation at the parts/process level.

Given these challenges, a number of companies have taken a broader approach to Design for Six Sigma. They treat DFSS as an opportunity to improve the overall new product development (NPD) process. For example, one consumer products company was faced with the problem that its NPD process generated only line extensions of existing products and technology. Its early DFSS wins came from developing good product development *charters* that quantified the value of the design opportunity.

A medical products company had a *prototyping* culture. Its engineers were encouraged to quickly jump to design solutions, assuming that they understood the doctors' and hospitals' needs and wants. This company's early DFSS wins came from improving its voice of customer listening processes. Instead of presenting product prototypes and getting limited feedback on features from the doctors, the engineers learned how to better observe the doctors working in their surgical environment and truly *hear* the doctors' voices. This improved *listening,* in combination with a disciplined cascade approach, allowed them to develop products with the performance factors and functions the docs really cared about.

Table 2.2 NPD gaps and improvement ideas.

NPD function	Gap	Improvement ideas
Selecting NPD opportunities	Choosing NPD projects with small value	Improve portfolio management; focus on business case for projects and adopt multi-generational thinking to identify bigger wins
Defining and resourcing the project	Poorly defined project scopes, insufficient project resources	Improve project chartering, adopt disciplined, cross-functional review approach to defining and assigning project resources, and apply lean principles to the NPD process
Understanding customers' needs	Prototype-First approach, limited customer voice in product development	Improve customer listening processes, improve requirements definition (e.g., through Quality Function Deployment), improve cascading of requirements into the design; consider product validation activities throughout NPD process
Developing a feasible product	Focus on a single prototype, science projects in development, poor manufacturability of designs	Adopt Pugh Concept Convergence approach, incorporate TRIZ and/or axiomatic design principles in NPD process, require design prediction of performance (design scorecards) prior to release to detailed design, incorporate *Design for X* (X = manufacturing, assembly, reliability, etc.) practices and reviews, bring manufacturing staff (and others!) into design process as early as practicable

Continued

Table 2.2 NPD gaps and improvement ideas. *Continued*

NPD function	Gap	Improvement ideas
Developing a producible product (and production process)	Tight tolerances identified on drawings relative to suppliers' capabilities, difficult to assemble designs, designs sensitive to parts, assembly variation	Include robust design principles and practices (e.g., Taguchi approach), include lean principles in process design, predict performance of design (variation transmission from parts, process to CTQs)
Verifying and validating the NPD output	Surprises in verification, validation, early production with high yields, followed by yield degradation	Add verification and validation activities throughout design process. Incorporate robust design principles and practices and *test* prior to manufacturing release
Post-launch production issues	Too long to stabilize design in production; production issues require reinventing the design	Design capability predictions prior to launch, design reviews to include manufacturing staff, improve capture and transfer of design knowledge to production staff (design intent)

In the broad sense of DFSS, you may identify gaps in almost any area of your NPD process and then work to close those gaps. Table 2.2 lists just a few ideas.

Obviously, you will have a hard time tackling all of these issues at one time. Don't treat DFSS as a "one-shot" deal. If you consider it a journey, you'll be more successful. We'll talk more about this in chapters 5 and 6.

KEY POINTS

You should walk away from reading this chapter with:

- A basic understanding of Six Sigma as a performance metric and as a business improvement approach
- An understanding of what it means to design a product for Six Sigma performance at the CTQ level and how to achieve this through design analysis and prediction
- A foreshadowing of how you will begin to implement DFSS—through a gap analysis of your current NPD process and systematic improvements to same

3

What DFSS Can Do for You and Your Company

INTRODUCTION

Now that you have a basic idea of the DFSS process, let's talk benefits. Dr. Joseph Juran is one of our heroes. Many years ago, we listened to him talk about a company's *language*. To engage engineers, he said, you must speak their technical language. When speaking to senior management, you engage them by speaking in their language—that of money. Middle managers must then be bilingual, because they need to communicate in both directions. Of course, there is that rare senior leader such as Mr. Soichiro Honda, who noted that he would rather hold a piston than a financial statement.

This chapter will follow Dr. Juran's advice. First we will discuss the financial benefits of DFSS. To obtain support and the resources needed to begin your DFSS journey, you'll have to build the business case and convince senior leaders that there are dollars to be gained. Next, we'll talk about the organizational benefits—reducing the friction that occurs in a typical product development organization (middle managers will live here). Finally, we will talk about the benefits your engineers will see when they begin practicing DFSS methods and tools. To wrap up, we'll share a *DFSS journey* example. This example will help you see both the benefits of systematically improving your development process and the work needed to get there—no pain, no gain! Looking back, we would have given an arm and a leg to have had something like DFSS when we were younger.

FINANCIAL BENEFITS OF DFSS (FOR SENIOR LEADERS)

We'll start with the cost side of the equation, where the case is relatively easy. Most studies of new product development processes start out with figures on the percentage of new products that fail in the market. Typical reported numbers fall between 40 and 80%. The Product Development and Management Association defines *failure rate* as "the percentage of a firm's new products that make it to full market commercialization, but which fail to achieve the objectives set for them." The financial consequences of these failures can range from a few thousand dollars for the backyard inventor to the $5 billion catastrophe of the Iridium satellite system.

What is the cost of such a failure? Tangibly, the time spent by the development team, investment in plant and equipment, raw materials and purchased parts/assemblies, sales, advertising, and warranty costs all go into the expense side of the equation. Stand these figures up against the revenue gained from the product. Intangibly, what have product failures cost you in the eyes of your customers? How has your reputation suffered because of the failures? What other products might you have launched with the resources you had to waste making things right?

How many of your new products launched over the last few years fall into this category? Do the numbers—what does this tell you about your product development process? Next on the list of financial costs of poor design quality is the cost of recalls. Although we love our 1995 Mustang, there have been several recalls on the car since we bought it. How much does Ford spend on these recalls?

Recently, a large medical device company has been in the news because of alleged design defects in their products. For example, three of its products are the subject of a class I recall by the Food and Drug Administration. This means that there is a reasonable probability that if a particular device is malfunctioning, it will cause serious adverse health consequences or death. The cost of these design defects can be measured in terms of product replacement costs, potential legal liability, and the impact on the company's value on the stock market—not a pretty picture.

Finally, the basic Six Sigma message needs to be restated. Whether you want to invoke Taguchi or Wheeler, there is a *quality loss/excess cost of production* when a part or product is produced off target. Figure 3.1 shows this picture.

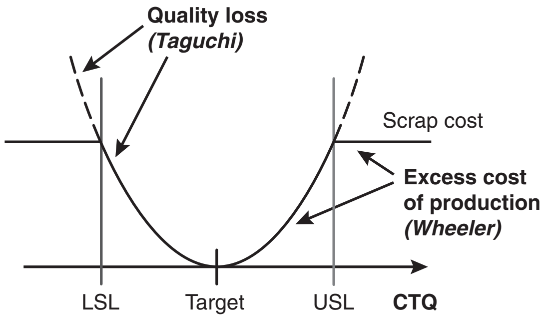

Figure 3.1 Quality loss/excess cost of production—off-target performance.

In keeping with the introduction's theme, we're not going to claim that DFSS will make all of your product launches wild successes. This sounds like an undersell, but the best we'll promise is a reduction in the failure probability (with a resulting decrease in failure costs).

You will decrease the probability that a low-value product will enter the pipeline. You will lower the probability of developing a product that does not meet its customers' needs. You will decrease the chances that your product will languish in development while a needed technology is being invented. You will decrease the likelihood that the product will have problems in production. You will decrease the probability of expensive warranty/recall costs. And, finally, you will decrease the quality loss/excess cost of production caused by the variation your customers see in your products.

Some examples follow.

Consumer Products Company

Although this company adopted and trained its staff on a full-scope DFSS development process, its biggest, earliest benefit came from the front end of the process. The organization had a habit of throwing resources at a lot of product development ideas without considering the market/business opportunity. When we did the initial team training, we didn't think that *project chartering* would be a big deal, but it turned out to be a huge benefit.

Of five pilot DFSS teams, two were killed at the charter. Leadership viewed this positively because those development resources were deployed to other, more valuable projects. The organization soon began requiring each development team to pull a charter together. Seeing what was happening, the R&D director organized charter reviews; he and his direct reports would take a day out of their busy schedules and review seven or eight teams, with the first couple of teams usually getting the most attention.

In addition, the teams were encouraged to think in *multigenerational* terms. The focus was on how to take an idea that was considered small and grow the idea through additional products that used the technology or through extensions of the first-generation product.

Medical Device Developer

A medical device company had grown quickly by being a *fast-follower* in the market. To continue its growth, it had to develop innovative products that would leapfrog the competition. We'll highlight two of many changes this company made to its development process:

> *Voice of customer/product requirements*—In its fast-follower mode, the company focused on a prototyping approach to new products. "Let's get something in the hands of the surgeons and see how they like what we've done to it!" To transition to a market leader, the company recognized that a better, richer voice of customer (VOC) gathering process was needed. Marketing took the lead on benchmarking VOC best practices and developed an approach that would help the product developers really understand the surgeons' needs. Although sometimes frustrating at first (what new process isn't?), the approach did begin to improve the identification of product requirements. The teams began to understand what product functions, instead of features, the surgeons were looking for and how well those functions needed to perform. These solution-open requirements gave the developers the opportunity to explore different design concepts instead of locking into a solution too quickly.

> *Cascade and prediction*—To ensure that the products met the requirements and were producible, the requirements/CTQs were cascaded to parts and process requirements, clearly linking the Ys with the Xs. Predictions of the designs' capability allowed developers and sponsors to clearly see (and mitigate before production) the designs' risks.

> *Results*—One of the product leaders came to kick off a DFSS training session. Her comments to the class included, "We're starting to knock some balls out of the park with our product launches. One surgeon picked up a new product and, without any instruction, began to use it intuitively. Our stabilization times used to average 12–15 months; now we're down to one month!"

Another product team saved hundreds of project hours because they were able to predict their product performance instead of spending the time and money building prototypes to test product performance. The project leader conservatively estimated that they launched the product two months earlier than they would have without DFSS.

Finally, a third product team was established to work with an outside company on an acquisition of new technology. The technology seemed quite promising, but the purchasing organization (our client) was disciplined enough to put a strong contract in place that allowed the team to investigate (a) the true needs of the customer and (b) the ability to mass-produce this product before it was required to pay the bulk of the contract. The team used a disciplined VOC gathering process and analyzed manufacturability of the design. They found that the prototype in their hands wasn't of great use/value to the customers, and it couldn't be mass-produced either. They released the contract in record time, allowing the seller to work on other ideas and allowing the buyer's engineers to work on a more valuable product.

These are generally the visible costs. If they aren't enough to motivate your leaders to improve the development process, consider some of the internal costs.

REDUCING INTERNAL FRICTION THROUGH DFSS (FOR MIDDLE MANAGERS)

We spoke with some folks recently about their nutrition products development process. While the company is currently successful because of a blockbuster product it developed, its pipeline isn't as full or productive as it would like. It has been launching products, but there have been significant production problems. Instead of development resources being occupied with new product development, executives are flying around the country to their plants and supplier facilities trying to fix production problems. They are honest enough to admit that they don't spend enough time on getting the design *right*—so that it can be produced reliably. The R&D folks are in a squeeze. They are being pressured to get product to market faster, but they are also having to deal with today's crisis. The time they spend trying to stabilize the product prevents them from putting development resources where they belong: in up-front *preventive, predictive design activities.* They just don't have the resources to do both.

DFSS will minimize these problems through its emphasis on *cascade/flowdown*—linking the product requirements qualitatively and quantitatively to parts and process Xs. By predicting the influence of X variation on the product requirements/CTQs, you will build assurance that the product can be produced within the required tolerances. If there is a problem in production, the documented cascade will help production engineers quickly pinpoint either which X is contributing to the Y problem, or what impact a change in an X will have on the product's Ys.

Another source of friction is the rework that occurs in development. You are probably familiar with the throw-it-over-the-wall syndrome—market research obtains requirements from the customer, throws them over the wall to development, which in turn develops a product and throws that over the wall to operations and sales.

If this isn't bad enough, consider how many times things change during development. Marketing or sales rushes into the development team meeting with a new feature that *absolutely* must be incorporated into the design. Development comes up with specs that suppliers can't meet, and the design has to be reworked. How many times have tools been ordered and reordered because the parts don't work together?

One of our clients tells the story of a production machine that was built overseas and shipped to their facility but had to be scrapped (cost: $20 million) because it didn't mate with the rest of the processes. Had they sent over the mating equipment (at a shipping cost of about $30,000), the problem would have been avoided. Just read *Dilbert* each day and you'll see Scott Adams poke fun at cases like these. How much does this cost?

Although not unique to DFSS, the method incorporates both business reviews and design reviews. At appropriate design points, we'll conduct a disciplined business review of the current state of the design by all stakeholders: market research, R&D, operations. This helps ensure that everybody agrees with design decisions and the current risks. Design reviews (usually before an associated gate) help ensure that the technical aspects of the design have been examined by peers and/or outside experts. See Chapter 4 for the purpose of each review. More importantly, DFSS is clear about the deliverables associated with each phase and how these deliverables are *frozen* (OK, at least somewhere close to being frozen) for the specific product under development. This helps the organization minimize changes that affect both the quality of the design and its development/launch schedule.

DFSS PERSONAL BENEFITS FOR PRODUCT DEVELOPERS: WHAT'S IN IT FOR ME? (WIIFM?)

We've lived and worked in engineering and development organizations. We've sat with our friends at lunch and complained about what was wrong. We've sat across the table from our customers and listened to their complaints about our products. That's not fun! As a product developer or engineer, you want to be successful. You want to have your name associated with successful products that make customers happy and make a lot of money for your company. You hate it when the development process prevents you from doing a good job.

So how will DFSS help you, the product developer? One of our friends is a pharmaceutical developer in Belgium who has embraced DFSS. He likes to ask a simple question: "What would you pay for . . .?" As a product developer/engineer, what would you pay for?

A clearly scoped development project

Understanding what the customer really wants, rather than the opinion of marketing, management, sales, or (fill in your favorite player here)

Product requirements that are rational, that you personally helped develop, and that you can clearly measure

The ability to develop a product concept that focused on meeting customer needs

Not having to invent a new technology in the middle of a development project

Not having whiplash from all the changing requirements or a product with ever-changing features

The ability to predict how well your product will perform in the face of process and supplier variation

A rational identification and mitigation discussion around the risks associated with your product

The knowledge and assurance that your product will be producible

Having to give up all those airline miles and hotel nights because you don't have to solve postlaunch problems with the design

No more late in the day (or at night) phone calls from manufacturing with the dreaded question "Can you tell us why . . ."

Actually being able to say, "That part of the design is frozen"

Feeling less and less like you're trapped in a *Dilbert* cartoon

One young engineer came up to us after a DFSS class. She said, "We've had a stage gate process for about a year, but you've shown us how to be successful in getting from one gate to another!"

ONE COMPANY'S PRODUCT DEVELOPMENT IMPROVEMENT JOURNEY

Because of confidentiality issues, this example of a company's journey toward improved product development has been disguised. You'll see that this company has gotten better, that the journey may not be pretty, but that its managers are convinced they are on the right path. What follows was adapted from a speech by the head of R&D to his product development teams.

The Company

About 20 years ago, YouAreHere, Inc., was founded with a clear purpose: to provide the military, business, and consumers with access to global positioning technologies and applications. Today, we are recognized as a leader in these solutions, and YouAreHere positioning systems can be found throughout the world in ships, aircraft, military vehicles, trucks, autos, yachts, and personal, hand-held applications.

As a GPS technology leader, our in-house product development expertise is responsible for overall product design and we control all key processes including development, manufacturing, sales, and service. To date, we have avoided the virtual model adopted by some of our competitors where they develop conceptual designs and then outsource detailed design and manufacturing. Our motto comes from Einstein's saying, "Everything should be as simple as possible, but no simpler."

Product Development History

Fifteen years ago, the GPS industry was still new. A few large companies dominated the military industry, but as the GPS signals were made available to civilian applications, many small companies were formed, developing products and bringing them to market. Quality standards were practically nonexistent. The technology available was new and marginally reliable. Customer expectations and service were generally low.

Our Product-Development "Infancy"

The first products we developed were trivial by today's standards and developed with no recognizable process or methodology. Simply by applying good engineering to innovation, we produced a new product every year or so. By 1995 we still only had about 10 products in our portfolio but were in the early stages of planning to invest heavily in a new, custom-built factory. The factory was planned to cost our entire year's sales.

By then, we had grown our development team to about six engineers. The "team" wasn't organized. There was an intrinsic feeling of what needed to be done but absolutely no processes or organization or management to make it happen. We worked on an interesting collection of projects with no concept whatever of project management or planning.

In fact, it was never entirely clear who reported to whom. However, new products did appear, and we moved into our new factory with, for the first time in our history, a significant corporate debt.

In 1998 we were still trying to find our feet in our new factory, still spending heavily, still had no detectable new product development process. However, due to the rapid development in electronics and the growing application and awareness of GPS technology, the market was growing rapidly and we were forced to make changes.

It's Hard to Grow Up

In the late 1990s, YouAreHere was in real trouble. We had a new plant to pay for and expensive automation systems that didn't work.

The causes included a familiar litany: no particular structure anywhere in the company, no real management. Product failures, poor delivery and service performance were rampant everywhere. We needed to take action quickly. The first semblance of a management team was put together with each member being assigned very clear responsibilities.

To say we put in place a new product development process would be a very gross exaggeration. We had to develop and get to market enough new products fast enough to survive. As in most potentially catastrophic situations, it was essential to make decisions quickly. This wasn't the time for hiring consultants or team-based decisions (we didn't even know what a team was). People were just told what to do.

One management principle is that the fastest way to do anything is to put together a small autonomous group of the right people, with a strong leader and a clear objective. Our product development process was driven entirely by one or two people, directly supervising a handful of engineers. This was essentially our entire new product development process, and it was simple but not sustainable.

We launched products in rapid succession, creating a full range of consumer, automotive, and marine GPS applications. The products were far from perfect and were driven into an uncontrolled manufacturing process. However, they were well received and kept us in business. So it was obviously possible to develop new products much faster than we had done previously, but how should it be done?

To survive and grow, we had to use all the knowledge in the company to develop new products better, more reliably, and faster than our increasingly competent competition. New product development needed to become a "total company" process.

Are We Adults Yet?

The biggest change to our new product development process was actually driven by our manufacturing leaders. In 2000, they convinced senior management to embrace Six Sigma as a "total company" improvement method. We restructured the entire company around a team-based implementation of Six Sigma.

Our implementation of Six Sigma required analysis of everything that we did to get a product to a customer; broke this down into manageable, clear process steps; and gave responsibility for each process to small teams (Six Sigma Improvement Teams, SSITs). Every SSIT owned a process, or several processes, over which they had complete control. They were given responsibility to supply the next downstream team with continuously improving quality. Linking all of these processes together gave us a chain connecting our whole business from suppliers to customers, encompassing everything we did.

Although this showed us the connection from market input through design to purchasing and manufacturing to sales and our customers, it did not show us how to make the process of new product development and introduction actually work.

With the Six Sigma process structure in place, we tried to find the best method to utilize it to develop and introduce new products. One of our Master Black Belts came from a large company and had experience with Design for Six Sigma. Through her guidance, we chose a six-phase model of product introduction, based on the DFSS Define–Measure–Analyze–Design–Verify/Validate–Implement (DMADVI) model along with a "kit" of design tools such as QFD, design of experiments, statistical tolerance design, and so forth. We worked out how to overlay and mesh this with our SSIT structure and expected an immediate transformation in our new product introduction process.

Oops, A Few Steps Backward!

Surprisingly, it didn't work. In fact, virtually nothing worked for some time. Our productivity didn't improve, our quality may even have gotten worse, and our new product introduction rate was flat. We flipped between standing back and trusting our SSITs to do the job, and jumping in and telling people what to do. Since the total company was based on the SSIT structure and new product introduction was a total company activity, if the SSITs didn't work, neither would new product introduction.

However, we needed to continue to deliver new products to stay in business. We tried to understand and use the new business processes, but when a product had to be delivered on time, it was easier to use the established methods. The gaps between where we were and where we wanted to be were very stressful, but the new products continued to appear.

So what was wrong? We probably expected too much too quickly. DFSS is a skill, a process in itself to be improved with practice over time. The demands placed on our team leaders are very high, the right people are essential, and it takes time to recognize this and find the right people. Our DFSS application was initially focused on just use of the tools and missed some essential steps, mainly around the management of new product development projects. For example, we were still working in silos, where marketing would gather the customer needs, the engineers would then use

QFD to develop requirements and the product design, and then turn it over to manufacturing with little up-front input from our operations partners.

We thought that just applying the tools and continually improving the functional processes would make our development process better. Finally we realized that we had to also adopt a cross-functional view of product development, with leadership through our management team. Although we were using DMADVI, we needed to learn to focus on the DFSS *process.* Among other things, this meant that we needed to ensure that DMADVI's "gates" were taken seriously (for example, through a disciplined review of the development deliverables and "sign-off" by all stakeholders). We also needed to break down the functional barriers—for example, our QFD sessions are now conducted with the whole team in place—marketing, R&D, and manufacturing. Even though we are in a fast-paced industry, we needed to stop the "requirements churn" that frustrated our engineers.

OK, We're Out of the Nest

So where are we now? I would like to say that we have a highly effective, smooth running, reliable, predictable new product introduction process that consistently delivers the right product at the right price at the right time.

However, DFSS is not about just being better than your competition. It's about getting better faster than your competition. Nothing stands still. Although the standards we work to now and the quality of our products make our efforts of 20 years ago look amateurish, the whole world has moved just as fast, or faster, than we have. Instead of our competition being a handful of small specialized, quirky companies, we are now head to head with global, vastly competent, multinational firms.

We have deliberately moved forward in technology, closer to the leading edge. This, ironically, reduces product life times and gives us the privilege of discovering the bugs in our suppliers' new components technologies!

Our expectations of new product introduction are higher. Each new product is critical to our business plans. Delays cost profit and growth. A quality problem at launch can virtually kill a product. However, like our products, the "devil is in the details." We need to continue to refine our processes; improve team effectiveness; eliminate unnecessary, non-value-added design and development work; and focus our design efforts on our strategic goals.

Next Steps

Where do we go next? With our fundamental DFSS development processes (almost) working and improving, hopefully quickly enough, we now need to be sure that we point this sharp, powerful, DFSS tool in the right direction. Our new product "hit rate" has always been high. We introduce very few products that flop. However, the stakes are now higher and our wide range of capability means we could do almost anything we choose to do. It's more important than ever that we choose to do the right things.

Since the company was started, there has always been an almost instinctive feeling of where we are going and how to get there. With a small company this is easy. A few times it has been spelled out, sometimes before we go somewhere, sometimes just after we have arrived!

We are now trying to improve our sense of direction and get everyone in our 500-plus-strong company to go in the same direction. Much of our energy now goes into defining our strategies and, in particular, our product development strategies.

We're moving from chaotic new product development to a strategically driven new product introduction process. Everything we do, every product we develop should take us a step forward toward our strategic vision, and then before we reach it, we should move the goal forward. It (hopefully) never stops!

KEY POINTS

You should walk away from reading this chapter with:

- The ability to define the *financial* benefits of DFSS—lower product failure rate, fewer product recalls, lower warranty costs, lower costs associated with excess variation (Wheeler's and Taguchi's points)
- The ability to define the *friction-reducing* benefits of DFSS—fewer R&D resources wasted fixing production problems, less rework during development, fewer disagreements between key product development functions
- The ability to define "What's in it for me?" benefits of DFSS—higher job satisfaction, or as Deming said, "Joy in Work" for your organization
- An ability to talk to different stakeholders about these benefits

4

DFSS: The Method and Roadmap

INTRODUCTION

Here, we'll show you what *good* can look like in a DFSS-based new product development process. This process will help you understand the broad application of DFSS as we described in Chapter 2.

At the 50,000-foot level, a quick web search will reveal several, seemingly different methods: DMADV, DMADVI, DMADI, I^2DOV, and so forth. At this level, there's not much difference. As you drift down to about 10,000 feet, you'll start to see some minor differentiation.

The real differences appear as you get closer to the ground. If you design complex equipment, DFSS will have a systems-engineering flavor. If you design products whose malfunction can cause harm or injury, DFSS will have a strong risk management flavor. For some companies (pharmaceutical pill production, for example), the basic production equipment may not change a lot from product to product. For them, the process design challenge is more about getting the recipe right. For equipment and devices, though, the production process may be unique for each product or one line may produce a variety of products. Here, DFSS will incorporate lean principles and practices.

For this chapter's journey, we've decided to use DMADVI as our DFSS framework. We'll describe how DFSS works for a developer of moderately complex products—somewhere between the Stone Age R&D shop of Chapter 2 and the next generation of the starship *Enterprise.* This will allow us to illustrate the main themes and methods without getting the description too simplistic or too complicated.

A note on the tools/methods described here: A great many tools/methods have been developed to support product development efforts. We have not tried to include them all but rather have attempted to identify at least one way of accomplishing the intended function. If you have a favorite (or even better) tool/method, please feel free to employ it in your development process' toolkit.

DMADVI: A QUICK HISTORY AND 50,000-FOOT VIEW

DMADVI stands for *Define–Measure–Analyze–Design–Verify (and Validate)–Implement*. In the late 1990s, GE tried to stretch DMAIC method into a design process that could be applied to business processes, the result being DMADV. Early applications of DMADV were focused on redesigning existing processes, particularly in GE's financial services companies. Teams were expected to first *Define* the business problem or opportunity, *Measure* the current process' performance, and then *Analyze* the root causes of the performance gap. So far, the method is not much different from DMAIC. The switch occurred in the Design phase, where the teams would design the new business process and test it in the Verify phase.

While this approach can work well for redesigning a process, GE's Eric Mattenson recognized that the company's growth goals required development of new services. He wanted a method that could be used to design a completely new business process—from a clean sheet of paper. In the late 1990s, we helped Eric turn DMADV into a new process design method by changing the intent of the MADV phases. *Measure* became a requirements definition phase employing quality function deployment to define the new process' requirements. *Analyze* became the concept definition and feasibility phase. The *Design* phase was beefed up to include more prediction and risk management tools such as design scorecards, variation prediction through simulation, and failure modes and effects analysis. Because we changed the intent of the Measure and Analyze phases, we should have changed their names. However, "DMADV" had already gained some brand recognition inside GE and so we left the phase names alone and focused instead on the phase details.

So how did DMADV cross over to become a product design method? Around the turn of the century, Johnson & Johnson decided to include design excellence as one of the pillars of its Process Excellence initiative. Elizabeth Iversen (a GE alumnus) brought together an internal team of product design experts to define the methodology and toolkit. They picked DMADV as their new product development framework and then worked to turn DMADV into a real, robust product development method. We were fortunate to be on the consulting team that supported this J&J effort. J&J has since applied DMADV (or a tailored version of DMADV) to medical device design, consumer product design, and pharmaceutical design as well as applying it to design/redesign of business processes, its original incarnation.

One thing that has always bothered us about DMADV is that it seems to trail off just when the fun starts—implementation. The design is ready for launch at the end of Verify/Validate, but it's still not out there. Therefore, we'll add an *Implementation* phase, and our method becomes DMADVI. Figure 4.1 provides a DMADVI block diagram.

At the 50,000-foot level, then, DMADVI's phases have the following objectives:

> *Define phase*—The business decides to apply resources to a product design effort. To ensure that the opportunity makes sense from a market and business perspective, the team first focuses on developing their project charter. Define is obviously critical; if the wrong projects enter the pipeline, the organization will be wasting development resources. Part of the chartering effort will be to identify gaps in existing knowledge about the opportunity; the idea may enter DMADV with varying degrees of maturity. Define is also the getting-started phase, and the team develops their initial project plans and project management processes. A charter review by project sponsors ensures that the team is working on the right product for the right market with the right goals.

> *Measure phase*—Measure is all about the customer and current competition. The team gathers the customer's voice, understands the competitive environment, and then translates this information into a set of prioritized, measurable product-level requirements. The most important of these are the critical-to-quality requirements, or CTQs. A business review at the end of the Measure phase aligns the team and its sponsors on the requirements that the development team will try to achieve.

> *Analyze phase*—Design concepts are identified and analyzed for their feasibility. Here, feasibility includes capability to meet the set of product requirements/CTQs, identification of potential product risks (with mitigation plans), and an early look at how producible the design will be (design for manufacturing and assembly), how reliable the product will be (design for reliability/maintainability), and so forth. We will seek the customer's feedback to either

Define	Measure	Analyze	Design	Verify/ Validate	Implement
Launch the project	Identify customers	Develop product required functions	Develop detailed product designs	Build pilot processes	Build full-scale processes, train staff
Define outcomes	Define state of knowledge	Develop detailed conceptual product designs	Develop detailed production processes	Validate pilot readiness	Perform start-up testing
Scope project	Develop and implement customer research plan	Develop high-level production processes	Refine capability and gap evaluation, perform tradeoffs	Perform pilot testing	Analyze gaps, determine root causes
Identify stakeholders	Translate customer needs to product requirements/ CTQs	Predict capability and evaluate gaps	Develop process control and validation plans	Analyze gaps, determine root causes	Transition to process owners
Select team	Specify targets, tolerance limits, and sigma targets			Evaluate scale-up potential	Evaluate and close design project
Determine project approach				Develop implementation and transition plans	
Create project plan					
Define project controls					

◆ = Phase Review

Figure 4.1 DMADVI product development phases.

physical or virtual prototype(s) here. A business review, likely preceded by a technical design review, approves the design concept for further development where large capital expenditures can occur.

Design phase—Once the concept's feasibility has been demonstrated, detailed design occurs. The product is optimized both to achieve targeted performance and to minimize variation around the targets. Sigma Level predictions will be developed for the CTQs and, perhaps, for other important/high-risk requirements. The production process is also designed and optimized to achieve its quality, capacity, cost, and safety requirements. Production tools will be designed and ordered, and the assembly process layout will be developed with lean principles guiding the layout of both the process and its associated supply chain. The company will begin to make the capital investments needed to produce the product. A final technical review of the design ensures that the design meets its requirements and that necessary trade-offs are recognized, and then the product design is frozen. A business review at the end of Design ensures the business that all important product risks have been identified and mitigated.

Verify and Validate phase—In this book, *verification* is defined as the ability of the product/process to meet its requirements; *validation* is defined as the ability of the product to meet the needs of the customers. Given these definitions, testing of the product will occur against its requirements, the process is verified to be able to consistently produce the product within specifications, and customers' reaction will be obtained to validate that the product meets their needs. The design predictions begun in the Analyze and Design phases will be verified using the assembly process and supplier parts. A business review during this phase provides the team with the approval to launch the product.

Implement phase—There can be significant overlap between Verify/Validate and Implement phases. Somewhere along the way, a production process is built. Process qualification activities (installation qualification for new equipment, operational qualification to demonstrate equipment capability and process qualification to verify overall process capability) will generally occur in the Verify/ Validate phase. Activation of the supply chain, ramping up to production volumes, and shipment/distribution of the product are all Implement activities. Early problems with the production process must also be addressed (if we've done a good job in DMADVI, there are not too many problems).

So when does DMADVI end? Some companies will hold a post-launch review of the product (six months post launch, for instance); we'll use a milestone such as this to define the end of a DMADVI project. Note that you may also consider a performance-based end, perhaps when certain business and process goals (for example, quality and quantity goals) are being met.

At a high level then, DMADVI just looks like a normal design and development process. We're going to drill down a bit further into the phases now to show you where the Six Sigma-ness of DMADVI kicks in.

As you review these phases, keep in mind that we're describing DMADVI in a linear fashion so that it's easy to understand. Practically, teams will implement DMADVI in a nonlinear fashion. For example, when we spoke recently with a product development leader about his team's activities, he had work going on in the Measure, Analyze, *and* Design phases. Likewise, when we discuss phase reviews, we're not suggesting that all work for that phase must be complete before the review is held and approval to move ahead obtained. The key questions at each phase review are "Do we have enough information to decide to move ahead with this project?" and "Are the key players (marketing, R&D, operations) in agreement about the decision?"

As you start to think about implementing DFSS in your organization, one of the early choices you'll make is whether to adopt DMADVI as your design framework or to map DMADVI into your current design process. Also, as we've noted several times, you should think about the current gaps in your design and development processes. You may decide to start your DFSS journey by improving just a few steps or elements of your current process.

DEFINE PHASE: OVERVIEW AND TOOLS

Purpose

In the Define phase of DMADVI, the product development project is chartered, planned, and resourced. The market concept for the product is clarified, in terms of unmet customer needs and/or a value proposition. The development team (at least a core team of marketing, R&D, and operations) will be in place. They have determined project scope, initial business, customer, and technical goals; have some understanding of the market/competitive environment and project risks; and have an initial plan to do the development work. The team also understands how their project fits within the overall product portfolio plan for the company.

The Define phase is where the company transitions from the "fuzzy front end" to actually committing development resources. There may be a significant body of "knowns" that the team can use in developing their charter—inputs from customer/market research, technology and competitive assessments, business strategy, and previous product generations. As part of chartering, though, the team will likely work to address "unknowns"—questions they want answered to provide assurance that the new product has a reasonable probability of being desired by the customer and that it will make money for the business.

Key Inputs

- Business and strategic plans
- Market research
- Customer needs (high level)
- Competitive information
- Current-generation products
- "Mature" technology

Steps

Note: The activities are listed in a very general sequence of events. A number of these activities can occur in parallel.

Launch the project—Decide that this product should be designed/redesigned (based on market research, company strategy, customer input, and value proposition). Assign overall responsibility for the project.

Scope project—Determine the boundaries of the project. Determine the project deliverables, what is in and out of scope for the project. Product designs may be divided into a series of releases or generations. Note that the product should employ known technologies with technology development occurring outside the DMADVI process. This is important to avoid schedule-busting science projects.

Define outcomes and goals—Determine how the success of the project will be measured (hint: consider customer, business, and technical goals). What unmet customer needs will be addressed? Will the design/redesign reduce cost or increase revenue or market share? Are there technical goals, such as compatibility with previous generations or other products in the portfolio?

Identify stakeholders—Who will be affected by the new design and can influence the success of the design project?

Select team—Determine full- and part-time members of the team. Which disciplines or departments should be involved? Early on, a skeleton team may take the design through feasibility (end of Analyze), with a larger team working through the rest of DMADVI.

Determine project approach—DMADVI provides a generic framework; determine how DMADVI will be tailored to the specific project. For example, a line extension will likely require less voice of customer input than a new product.

Create project plan—Develop a work breakdown structure, network diagram (for example, a PERT chart), and/or a Gantt chart. Estimate time, cost, and resource requirements for the project. Assess the risk of achieving the plan (methods include critical path method, Monte Carlo simulation of the plan, critical chain method; the latter is based on Goldratt's theory of constraints).

Define project controls—Develop communication plans, change control (for the design), change management (for stakeholders, staff), review plans (design and business), and review risk and opportunity management processes. Table 4.1 shows the primary tools used in the Define phase.

Key Outputs

- Approved project charter (scope/deliverables, goals, metrics, targets, business case, milestones)
- Skeleton team
- Project approach and plans (draft)

- Design process controls
- Risk/opportunity management processes
- Stakeholder analysis
- Multigenerational plans

Phase Review

The first project review occurs when the team completes the Define phase. Sponsors review and approve the team's charter, multigeneration plan, and draft project approach and plan. Sponsors review the team's current risk and opportunity assessment of the project and concur with the team's mitigation actions. Some companies include a "charter check-in" in the middle of Define to make sure that the project fits strategically, will be supported, and will get funded; this can help kill projects early.

MEASURE PHASE: OVERVIEW AND TOOLS

Purpose

In the Measure phase, the team will obtain the voices of the various customers of the product. These will include customers external to the business who will buy and use the product, trade partners (for example, retail outlets such as Wal-Mart or Target), internal customers and the stakeholders who will be affected by or who may influence the success of the project (management, regulatory bodies, among others). The goal of this phase is to develop a set of requirements (some of which will be critical to quality) that the design team can use as inputs to their design. A clear linkage between the voices and requirements must be established in this phase.

Key Inputs/Use

- Approved project charter (scope/deliverables, goals, metrics, targets, business case, milestones)
- Stakeholder analysis
- Multigenerational plans
- Preliminary voice of customer
- Existing customer information
- Market research studies

Table 4.1 Define phase tools.

Tool	Use
Market research studies	Understand unmet needs of the customer, identify opportunities for new products.
Process capability studies	Understand current production process capabilities, technology gaps.
Technology survey/ assessment	Understand technologies that may be applied to achieve unmet customer needs or improve capability to meet existing needs.
Competitive analyses	Understand current competitive products, predict future direction of competitive offerings.
Benchmark studies	Understand current and future competitive offerings, but scope is broader than own industry.
Analysis of government/ regulatory requirements	Understand regulatory direction and threats to current products and opportunities for new products (for example, environmental protection changes).
Value proposition	Understand all internal and external stakeholders and what benefits the new product will offer each.
Chartering	Understand and achieve agreement on scope, goals, business case, timeline, and resources needed for the development project.
SIPOC mapping	Understand usefulness of scoping a project using SIPOC (Supplier–Input–Process–Output–Customer) maps.
Multigenerational product planning	Develop and scope a vision for series of product releases, aimed at sustaining market leadership position and identifying needed technologies (for research by advanced technology process).
Project risk and opportunity management	Identify, quantify, prioritize, and mitigate business risks; understand technical, organizational, external, and other risks and opportunities faced by the project. SWOT (strengths, weaknesses, opportunities, and threats) analysis is useful here. Project risks may be mapped on a severity/likelihood map. Stakeholder analysis is useful throughout the project.
Project management and planning	Develop and resource the overall plan for the product development project. Develop project control processes (configuration management, scope change management, communication plans, etc.).
Team management	Develop and implement processes to ensure effective, efficient teamwork (idea generation, decision making, conflict management, meeting, etc.).

Steps

Note: The activities are listed in a very general sequence of events. A number of these activities can occur in parallel.

Identify customers—Determine external, internal customers; review stakeholder list generated in Define phase.

Define current state of knowledge about your customers' needs—Review existing customer information, including complaints, compliments, and market research studies.

Determine what more you need to know about the customer—Review your project charter and existing knowledge of customer needs, looking for the gaps in knowledge your team needs to fill.

Develop and implement customer research plan—Determine what information must be collected, determine appropriate voice of customer methods (interviews, focus groups, surveys) to fill the gaps in your knowledge of the customer.

Translate customer needs to product requirements/CTQs—The voice of the customer is generally obtained in their language. A filtering and translation process takes the customers' voices as input and develops a set of requirements stated in the technical language of the product. We call these *product requirements*. Quantitative research is used to help the team prioritize the customers' needs.

Quality function deployment (QFD) can be employed to organize and structure the VOC gathering/translation process. The translated requirements (and other business requirements, regulatory, code requirements) are measurable, testable, and feature-free requirements. The most important of these product requirements are identified as the CTQ requirements (planned for heavier focus because they are differentiators or difficult to design against). Deciding which requirements are critical helps the team focus on what will differentiate their product in the market. A typical product may have 30–100 requirements, but only 3–5 of these will be CTQs.

Specify targets, tolerance limits, and Sigma targets—Numerical goals are set for the product requirements. Allowable variation and defect rates (that is, Sigma targets) are established to help the design team objectively judge their design. As a minimum, the CTQs will receive extra design attention; these are recorded on a design scorecard. Table 4.2 shows the primary tools used in the Measure phase.

Table 4.2 Measure phase tools.

Tool	Use
Voice of customer/ market research tools (in-depth interviews, focus groups, surveys)	Gather detailed qualitative needs from customers (interview, focus group), prioritize needs or translated product requirements through quantitative research (surveys).
Image diagrams	Method of obtaining a picture of what life is like for the customer using the product.
Requirements diagrams	Organization method to identify and group product requirements into themes.
Affinity sort	Organize large number of ideas (here, customer needs) to identify themes.
Structure tree	Develop a hierarchy of ideas (here, high-level to detailed customer needs), often used as input to QFD's house of quality.
Quality function deployment (QFD)	Matrix-based translation tool. In the Measure phase, QFD is applied to translate prioritized customer needs into a set of product-level requirements (with associated priorities, metrics, targets, and limits).
Conjoint analysis	Market research technique used to identify the utility customers have for product features or requirements.
Kano analysis	Market research technique to identify "must-be," "one-dimensional," and "delighter" needs or product requirements.
Radar chart (also known as spider diagram)	Graphical technique to display many attributes' performance on one chart. A common Measure phase use is to show importance and satisfaction gaps of needs/requirements.
Product requirements management	Systematic identification, deployment, and verification process to ensure requirements are met by the design; trade-offs are consciously made.

Key Outputs

- Voice of customer/stakeholder (all research performed as part of Measure phase: customer segmentation analyses/decisions, complaint/complement analysis, interview/focus group transcripts, competitive analyses, importance/satisfaction survey results, etc.)
- Product requirements, including function, metric, targets/limits, and variation or Sigma Level targets (the most important of these identified as the CTQs)
- Basis for defining product requirements: quality function deployment, needs/requirements matrix, others

- Design scorecard at the CTQ level
- Updated Define phase deliverables (charter, multigenerational plan, risk/opportunity management, stakeholder management, etc.)

Phase Review

The second business review occurs when the team completes the Measure phase. Sponsors review and approve the team's product-level requirements/CTQs (and their basis in the VOC/competitive information is gathered). Updates/revisions to the charter, multigenerational plan, and the project plan are also reviewed and approved. Sponsors review the team's current risk/opportunity assessment of the project and concur with the team's mitigation actions.

ANALYZE PHASE: OVERVIEW AND TOOLS

Purpose

In the Analyze phase, the design team will identify and evaluate possible design concepts to meet the requirements defined in Measure. The decisions made in this phase will determine a large percentage of the ultimate quality and cost of the product. Moving too quickly through this important phase can limit the potential market for and ultimate success of the product. Here, the team will develop concepts that may fulfill the product requirements, analyze these concepts, and select the optimum design concept. Finally they will begin to verify that the design concept will meet its customers' requirements, through feasibility analyses and early capability predictions.

Key Inputs

- Project charter
- Project plan
- Multigenerational product plan
- Risk/opportunity management plan
- Product requirements/CTQs
- Design scorecard—CTQ level

Steps

Note: The activities are listed in a very general sequence of events—a number of these activities can occur in parallel.

Develop product functions—Functional analysis takes a complex product and breaks it down into the functions that must occur for the requirements (CTQs, for instance) to be met. This analysis sets the stage for identification of product concepts.

Develop conceptual product designs—Benchmarking, structured invention (for example, TRIZ), and other creative methods are employed to identify concepts for the product functions (in parallel with intellectual property searches to ensure that the technology is available for use). The various functional concepts are virtually assembled into an overall product concept. Alternative concepts are evaluated and a best-fit concept is selected. The Pugh matrix approach is a useful means of identifying, iterating, and converging on the best concept.

Design *cascade* begins here. Cascade (or flow-down) is the systematic linking of the highest-level Ys (product requirements) to lower-level Ys (for instance, subsystem, assembly, component, process requirements) and eventually to the process parameters (fabrication and assembly Xs) that will determine the product's quality. Qualitative cascade ensures the design effort is focused on meeting its overall requirements and can be accomplished by continued application of the QFD method, structure trees, or other hierarchical analysis. Quantitative cascade allocates the allowable variation at the CTQ level to each of the lower-level requirements that affect the CTQ (error budgeting). The team uses their engineering knowledge, simulation, or experimentation to develop transfer functions that help them understand how their CTQs will behave as a function of the design (and eventually process) Xs; that is, the $Y = f(X)$ relationships.

Identify high-level production processes—Production process technologies are investigated for the selected concept. Feasibility studies will ensure that the product can be produced with available, mature production technologies (again, avoiding science projects in the middle of development).

Evaluate feasibility, predict capability, and evaluate gaps—Depending on the product requirements, analyses, predictions, and prototype tests are made to assess the ability of the concept to meet requirements and to ensure the design concept is as "robust" as possible. This is *flow-up,* moving from the Xs to the product CTQs or Ys. Taguchi's parameter design principles recommend that the design concept be subject to different noises—with design parameters then set to minimize the sensitivity of the design to noise. Design simulations (such as finite element analyses) can be used to predict the performance of the design and assess feasibility. Design for X principles will be proactively applied to the design (here, X = manufacturability, assembly, environment, human factors, reliability, maintainability, inspectability/testability, and others, as appropriate). The output of the prediction/ feasibility efforts is recorded in the requirements workbook and on the design scorecard for the CTQs. Product risks are identified, prioritized, mitigated, and documented via failure modes and effects analysis. Table 4.3 shows the primary tools used in the Analyze phase.

Table 4.3 Analyze phase tools.

Tool	Use
Quality function deployment	Analyze phase application of QFD is to deploy product requirements to subsystem and/or subassembly requirements.
Functional analysis breakdown (tree diagram)	System engineering technique to identify and decompose functions required to be performed by the product. Physical/software/human architectures are defined to meet these functions.
Creativity techniques	Brainstorming, attribute listing, mind mapping, and other techniques used to identify design concepts (individually or in a group setting).
Benchmarking	Systematic method of identifying technologies that can be applied to the design concept (either product or process).
Intellectual property analysis	Systematic identification of technologies aimed at avoiding use of technologies owned by others (early in concepting) and then protecting company-developed technologies (through patents, trademarks, copyrights, trade secrets, etc.).

Continued

Table 4.3 Analyze phase tools. *Continued*

Tool	Use
Pugh concept convergence	Matrix-based approach to identify, evaluate, and converge on a superior design concept that meets the product requirements (can be used at any design level, including production process concepts).
Cascade/flowdown	Process of deploying and allocating higher-level to lower-level requirements. Analyze phase cascades product-level requirements to subsystem requirements (minimum).
Transfer functions	Quantitative relationships from CTQs (and other important requirements) to lower-level requirements and, eventually, to process variables.
Simulations (e.g., finite element analysis)	Computer representations of the product developed for the purpose of more rapidly, cheaply (and, in many cases with greater insight) predicting product performance than physical prototypes.
Design of experiments (DOE)	Statistical technique to evaluate which design factors affect the CTQs/requirements, optimize the design, and ensure its robustness (i.e., Taguchi techniques). Analyze phase purpose is to identify most important design factors as part of feasibility assessment.
Industrial design/human factors	Discipline whose purpose is to ensure that the human interface with the product is well designed.
Design control/ configuration control	Design process to ensure that requirements, features, and design parameters are identified and controlled.
Design for X	Proactive design techniques to consider “X-factors” such as design-to-cost, manufacturability, assembly, environment, maintainability, inspectability, and testability.
Reliability methods	Proactive design techniques to ensure that CTQ performance is achieved over the intended product life. Examples include reliability target setting and allocation, reliability prediction, failure modes and effects analysis (FMEA), fault tree analysis (FTA), reliability testing. If the product is maintainable, this includes setting and allocating maintainability targets, as well as maintenance strategy development.
Consumer feedback	Market research methods to obtain feedback on design concept(s).
Design reviews	Meetings between the team and peers/experts to review the design and associated analyses. Analyze phase purpose is to challenge the design concept and identify weak points before moving to the production-version design.

Key Outputs

- Functional analyses, correlated to goals/needs
- Selected conceptual design(s) (including *virtual* or prototype models)
- High-level design drawings (layouts, flowcharts, schematics)
- Subsystem (and lower-level) requirements (from cascade—traceable in both directions—up to CTQs, down to subsystem or lower level)
- Process technology selected
- Supporting feasibility analyses/test results:

  —Calculations (including transfer functions)
  —Predictive analyses (FMEA, EMEA, FTA, FEA, stress)
  —Experimental results
  —Test plans/results
  —Design for "X" analyses (manufacturability, assembly, maintainability, environment, human factors, others)
  —Regulatory and environmental impact analyses
  —Cost estimates to produce product
  —Customer/consumer feedback on concept

- Maintenance strategy
- Design scorecard (typically comparing average performance to targets)
- Updated deliverables from Define and Measure phases

Phase Review

The third business review occurs when the team completes the Analyze phase. Sponsors review and approve the team's design concept (and its ability to meet the product requirements, as determined by the feasibility studies). A technical design review of the concept by peers and/or experts can provide valuable input to the business decision makers. Updates and revisions to the charter, multigenerational plan, and project plan are also reviewed and approved. Sponsors review the team's current risk assessment of the project and concur with the team's mitigation actions.

DESIGN PHASE: OVERVIEW AND TOOLS

Purpose

Here the production version of the product, as well as that of the production process, is developed and finalized. Design analyses/prediction activities are refined and completed; the product has been determined to meet its requirements; trade-offs are made where necessary. In preparation for verification and validation of the design, process controls and validation plans are developed.

Key Inputs

- Outputs of Define, Measure, and Analyze phases
- Feasible concept(s)
- Preliminary production process design

Steps

Note: The activities are listed in a very general sequence of events—a number of these activities can occur in parallel.

Develop detailed product designs—The work done in the Analyze phase is continued at the detailed level. Materials are selected; assembly and parts designs are completed. Cascade continues to part requirements. Drawings are developed with critical parameters/ dimensions noted. By this phase's completion, the design will be developed to the point where it can be produced using production-version equipment and processes.

Develop detailed production process design—Likewise, the production process design is developed. Technologies are selected for part fabrication and assembly processes defined. Suppliers are selected for outsourced parts/ assemblies. Tooling design and analysis (such as mold fill analysis) occurs. Measurement systems are selected and/ or designed. Cascade is completed with identification of the critical process parameters (Xs) that affect the design's quality.

Refine capability and gap evaluation, perform tradeoffs—In the Analyze phase, capability analyses were begun to assess the ability of the design to meet its requirements. Generally, these analyses will ensure that the design meets its targets. Here in Design, the team will also determine how variation in the product and part Xs will flow up to the CTQs or other Ys. Many development teams are familiar with performing tolerance stack-ups to ensure that design requirements are met. The concept of predicting variation in the CTQs should not be a novel idea, although actually doing so is generally new to many teams. The design scorecard should be completed here, with accurately predicted variation (rather than simple tolerance highs and lows).

The team should be able to walk into a design or business review and state how well the design is predicted to perform in terms of both average and Sigma levels (as well as for the product's life–reliability predictions). Cost estimates, design for X analyses, and product and process FMEAs are completed. For maintainable equipment, preventive maintenance plans are developed, including spare-parts lists.

Develop process control and verification/validation plans—Testing and product verification/validation activities are planned. In preparation for these efforts, the necessary process controls—procedures, protocols, bills of material, device master record, critical parameter identification, measurement, and sampling and reaction plans—are developed. Table 4.4 shows the primary tools used in the Design phase.

Table 4.4 Design phase tools.

Tool	Use
Quality function deployment	Design phase application of QFD is to deploy product requirements to parts and process requirements.
Functional analysis breakdown (tree diagram)	System engineering technique to identify and decompose functions required to be performed by the product and production process. Physical/software/ human architectures are defined to meet these functions.
Creativity techniques	Brainstorming, attribute listing, mind mapping, and other techniques are used to identify parts and process concepts (individually or in a group setting).

Continued

Table 4.4 Design phase tools.

Tool	Use
Benchmarking	Systematic method of identifying technologies that can be applied to the parts and process design.
Intellectual property analysis	Systematic identification of technologies aimed at avoiding use of technologies owned by others (early in concepting) and then protecting company-developed technologies (through patents, trademarks, copyrights, trade secrets, etc.).
Pugh concept convergence	Matrix-based approach to identify, evaluate, and converge on a superior design concept that meets the product requirements (can be used at any design level, including production process concepts).
Cascade/flowdown	Process of deploying and allocating higher-level to lower-level requirements. Design phase cascades product-level requirements to parts and process requirements.
Transfer functions	Quantitative relationships from CTQs (and other important requirements) to lower-level requirements and, eventually, to process variables.
Simulations (e.g., finite element analysis)	Computer representations of the product developed for the purpose of more rapidly, cheaply (and, in many cases with greater insight) predicting product performance than physical prototypes.
Design of experiments (DOE)	Statistical technique to evaluate which design factors affect the CTQs/requirements, optimize the design and ensure its robustness (i.e., Taguchi techniques). Design phase purpose is to optimize the product design (and also production process design).
Industrial design/human factors	Discipline whose purpose is to ensure that the human interface with the product is well designed. Design phase focus will be on usability testing of product.
Design control/ configuration control	Design process to ensure that requirements, features, and design parameters are identified and controlled.
Design for X	Proactive design techniques to consider “X-factors” such as design-to-cost, manufacturability, assembly, environment, maintainability, inspectability, testability, and others.
Reliability methods	Proactive design techniques to ensure that CTQ performance is achieved over the intended product life. Examples include reliability target setting and allocation, reliability prediction, failure modes and effects analysis, fault tree analysis, and reliability testing. If product is maintainable, this includes maintenance plans (reliability-centered maintenance, predictability-centered maintenance) and predictions, spare parts decisions, and others.

Continued

Table 4.4 Design phase tools. *Continued*

Tool	Use
Consumer feedback	Market research methods to obtain feedback on product design.
Lean production principles	Principles applied to production process design to ensure waste, work-in-progress are minimized, that the product is produced "just-in-time" to meet customer demand.
Control planning	Method to control the quality of the product by ensuring that the production process Ys and Xs are identified, control and critical control points are identified and measured, and reaction plans are developed with accountability for action when necessary.
Design reviews	Meetings between the team and peers/experts to review the design and associated analyses. Design phase purpose is to challenge the final product design and associated production process design to identify weak points before moving to verification/validation activities.

Key Outputs

- Functional analyses
- Production-version product design (including parts and assemblies)
- Subassembly, parts, and process requirements (from cascade)
- Design drawings (layouts, flowcharts, schematics)
- Bills of material
- Product specifications
- Production process design (including equipment, tooling, layout, and measurement systems)
- Control plans, procedures, protocols
- Supporting design analyses/test results:

  —Calculations (including transfer functions)

  —Predictive analyses (FMEA, EMEA, FTA, FEA, stress analyses)

  —Experimental results

- —Test plans/results
- —Design for X analyses (manufacturability, assembly, maintainability, environment, human factors, others)
- —Regulatory and environmental impact analyses
- —Cost estimates to produce product
- —Customer/consumer feedback on production version product
- —Production process analyses (capacity, lean, mold fill, environmental impact, others)

- Maintenance plans, spare-parts plans
- Design scorecard (by end of Design, CTQ variation predictions are developed and compared to specifications, predicting design capability)
- Product/process verification and validation plans

Phase Review

The fourth business review occurs when the team completes the Design phase. When the product design is deemed complete, a design review occurs. Following this review, the design is frozen so that process design work can occur with minimal changes/schedule impacts. Sponsors review and approve the team's product and process design. Updates/revisions to the charter, the multigenerational plan, and the project plan are also reviewed and approved. Sponsors review the team's current risk assessment of the project and concur with the team's mitigation actions.

VERIFY AND VALIDATE: OVERVIEW AND TOOLS

Purpose

Verification confirms the product and process meets their requirements; validation confirms the product meets the needs of the customers for the intended use environment. Pilot testing can be a key part of the product's validation. Based on the results of these activities, the decision to scale up to full production is made; implementation and transition plans to support the scale-up are developed.

Key Inputs

- Design outputs from previous phases
- Process control plans
- Product/process verification and validation plans

Steps

Note: The activities are listed in a very general sequence of events—a number of these activities can occur in parallel.

Build production processes—Production facilities, equipment, information systems, and so forth are procured and constructed in preparation for qualification tests. For some businesses, these may be on a pilot scale, with full-scale processes built after successful pilot results.

Verify process readiness—Start-up testing of the production processes is completed. The processes are tested to determine whether they are capable of producing the product. Typically, this involves installation qualification, in which the production equipment is installed and demonstrated to work, followed by operational qualification; here, the equipment is tested to ensure it meets its requirements (as cascaded from customer, business, and other stakeholder requirements).

Perform pilot testing—Typically, this involves performance qualification, in which the process is verified to be able to produce product within all of its requirements. The design scorecard is updated; verification results (Sigma levels) are compared against design predictions. Production version product is produced. The product is offered to customers; validation that the product meets the needs of the users in the intended use environment is performed.

Analyze gaps, determine root causes—Problems experienced by the customer are identified, root causes are determined, and the product/process is revised to eliminate the gaps.

Evaluate scale-up potential—A business decision is made to scale up the product to full scale.

Develop implementation and transition plans—Plans to fully implement the product are developed. Table 4.5 shows the primary tools used in the Verify/Validate phase.

Table 4.5 Verify/validate phase tools.

Tool	Use
Pilot/qualification testing	Various tests are designed to ensure that the production process will produce the product within specifications. Examples include installation qualifications (ensuring the production equipment operates) and operational qualifications.
Design reviews	Meetings between the team and peers/experts to review the design and associated analyses. Verify and Validate phase purpose is to review the results of design verification/validation and process verification activities to ensure the product is approved for launch.
Root cause analysis tools	Methods such as Define–Measure–Analyze–Identify–Execute–Control (DMAIEC), tools such as cause/effect diagrams, fault tree analysis, and five-whys are used to analyze process problems and their root causes so that corrective action can be taken.
Project management	Planning and managing methods to ensure that the production start-up and product launch are performed as quickly and efficiently as possible.

Key Outputs

- Verified production processes
- Verified and validated product
- Process control plans (updated)
- Configuration control plans (products, processes, suppliers)
- Implementation/transition plans

Phase Review

The fifth business review occurs when the team completes the Verification and Validation activities. Sponsors and business leaders approve product launch. Updates/revisions to the charter, multigenerational plan, and project plan are also reviewed and approved. Sponsors review the team's current risk assessment of the project and concur with the team's mitigation actions.

IMPLEMENT: OVERVIEW TOOLS

Purpose

Here the product is launched. The design is transitioned to the operating forces (that is, for products), and the device master record is completed and transferred to production. Further commercialization of the product may occur, and the design of the next product generation begun. The close of this design project is at hand. Lessons learned, as well as the history of the design, are documented. As appropriate, the design team is rewarded and recognized.

Key Inputs

- Verified production processes
- Verified and validated product
- Implementation/transition plans

Steps

Note: The activities are listed in a very general sequence of events—a number of these activities can occur in parallel.

Build full-scale processes, train staff—For many products, existing facilities are adapted to support the new production processes. In some cases, new production facilities/processes will be required.

Perform start-up testing—Necessary testing of the new production processes is performed. Production is ramped up to full scale.

Analyze gaps, determine root causes—Problems noted with early production units/processes are identified, root causes are determined, and appropriate countermeasures are implemented.

Transition to process owners—As the new product enters production, the design team performs a turnover to operating forces. Bills of material, device master records, process procedures, and control plans are transitioned. Design history files are updated.

Evaluate and close design project—Before the design team disbands and begins to work on the next products, lessons

Table 4.6 Implement phase tools.

Tool	Use
Project management	Planning and managing methods to ensure the production start-up and product launch are performed as quickly and efficiently as possible.
Process control methods	Method to control the quality of the product by ensuring that the production process Ys and Xs are identified, control and critical control points are identified and measured, and reaction plans are developed with accountability for action when necessary.
Root cause analysis methods	Methods such as DMAIEC are used to analyze process problems and their root causes so that corrective action can be taken.
Configuration/change management	Methods ensure that the as-designed product configuration is maintained and that those changes to materials, processes, or supplier-provided parts are evaluated for their effect on the product and its requirements. For example, the device master record contains all the routine instructions required to manufacture the product. The design history file contains or references the records that demonstrate that the design was developed in accordance with the approved design plan and the requirements of this part.

learned are generated, good practices are recognized, and improvement opportunities are identified. All of these should be fed back to the owners of the design process to improve the overall design processes. Table 4.6 shows the primary tools used in the Implement phase.

Key Outputs

- Production processes operating at desired quality and capacity levels
- Commercialized product
- Device master record
- Design history files
- Lessons learned/best practices
- Updated multigenerational plans

Phase Review

The last business review occurs when the organization defines the design effort to be complete (for instance, six months postlaunch or upon achievement of specified performance goals). This last review is intended to capture lessons learned (identifying both problems that occurred and the best or improved practices), determine next steps for the product line, and reward and recognize the team for their efforts.

KEY POINTS

You should walk away from reading this chapter with:

- Understanding that DMADVI is an *end-to-end* design process and that DFSS is adaptable to many product development challenges in many industries. In addition, although our focus has been on physical products, DFSS has been applied to business process/service designs.
- Understanding that the core of DMADVI includes gathering voice of customer information to identify product requirements/CTQs, developing design concepts to meet the requirements/CTQs, cascading requirements from product to process and predicting/verifying that the design meets these requirements—both targets and specification limits.
- A choice for you to consider: Bring the DMADVI framework into your organization as the DFSS method or use your existing framework and incorporate DMADVI's tools and methods.

5

Gap Analysis and Readiness for the DFSS Journey

INTRODUCTION

After the first four chapters, we're sure you agree that DFSS makes sense and that DFSS could help improve your organization's new product development. That's the easy part. On the other hand, the hard part is figuring out how to make it work for *your* organization. Everybody wants to win the game, but doing it is hard!

We've been easy on you so far in this book. You've had the chance to understand a bit about DFSS. We hope you understand why companies have found it valuable to integrate DFSS in their product development processes.

In the next four chapters, we're going to turn the spotlight on your organization's new product development function. We have seen companies try to implement DFSS, only to have it become another "program of the month." We don't want that to happen to you.

We've mentioned Six Sigma's improvement method, DMAIEC, throughout this book. To help ensure that you get it right, we're going to use DMAIEC thinking to help you plan and execute your DFSS implementation. This chapter will address most of the Define, Measure, and Analyze step activities; the next chapters will take you through Identify, Execute, and Control.

You'll find that *change management* is critical in driving and sustaining a DFSS culture. With resources stretched in all areas of organizations, leaders are finding it critical to make effective, sustainable changes with the least amount of resistance. DFSS is more than a set of processes and tools. It truly represents a cultural shift in the way you do business. We will share ways with you to assess your organization and determine whether it is truly ready for a change like DFSS.

In this chapter, you are going to figure out whether DFSS is for you by answering three basic questions:

- Why do you think DFSS is needed in your organization; that is, why do you want to do this?
- Do you understand the gaps between your current product development process (if you can even call it that) and *what good looks like?*
- Is your organization ready for the changes that DFSS demands? How much of a change can your organization take now?

WHY DO YOU WANT TO DO DFSS?

By 1985, Florida Power & Light (FPL) had been playing with its quality improvement program for about four years. While we had seen a good bit of local improvement, our execs weren't happy—the program wasn't making as big an impact on the company as they'd expected. A group of 12 respected middle managers (informally known as the *quality apostles*) were charged to benchmark quality programs around the world. They learned about Crosby, Deming, and Feigenbaum. They went to Detroit, to Europe, and finally to Japan. Here, they happened to find the Kansai Electric Power Company (KEPCO).

KEPCO had just won the 1984 Deming Prize, and its overall performance was excellent. Because it was similar to FPL in size and characteristics, KEPCO was judged to be a good model from which FPL might learn. The apostles met with KEPCO representatives, discussed their approach, and asked how they did it—who helped them, and so forth.

In due time, we were introduced to Dr. Tetsuichi Asaka, the lead Union of Japanese Scientists and Engineers (JUSE) counselor for KEPCO. Our senior leadership arranged a meeting in Japan with Dr. Asaka and several other counselors. We thought at first that these counselors were like typical Western consultants: throw money at them and they'd do whatever we wanted.

That was a big mistake, and it was our first of many in dealing with these strict teachers. The first question Dr. Asaka asked our leadership was "Why do you want to do this?" We gave him such reasons as we wanted to get better and we think this will be good for our reputation. He wasn't happy with our answers. We were sent home (without a commitment!) to figure out *why*? Six months later, we went back with an answer that satisfied the counselors. It was only then that Dr. Asaka agreed to work with us.

What made the difference? Well, from several fuzzy, feel-good reasons, we had been able to articulate a clear business reason to change our management systems. These reasons included: (a) concern that the *economic* disaster that General Public Utilities experienced after Three Mile Island could happen to us (FPL owned two nuclear plants at that time), (b) concern over our customers' *satisfaction* with our electric service and its price (this, after the two oil shocks of the 1970s; the rest of our generating capacity was oil fired), and (c) concern that the *regulatory environment* could shift, deregulating the electric utility industry, because we had just seen the telephone industry deregulated and Bell Telephone broken into pieces. So we're going to apply the same hard question to you that Dr. Asaka posed over 20 years ago: Why do you want to adopt the discipline of DFSS?

Chapter 2 gave you some reasons why others have done so. So here's your first homework—building the business case for DFSS. Answer the following questions. Go out and get the data. Table 5.1 shows a sample of possible pieces of the business case for DFSS.

Table 5.1 Examples of DFSS business case elements.

Audience	Candidate reasons	My company's situation
Senior leadership (language = *money*)	New product failure rate: The percentage of your new products that make it to full market commercialization but that fail to achieve the objectives set for them Cost elements: • Development resources • Plant and equipment investment • Raw materials and purchased parts/assemblies • Sales and advertising	

Continued

Table 5.1 Examples of DFSS business case elements.

Audience	**Candidate reasons**	**My company's situation**
Senior leadership (language = *money*)	• Warranty costs • Intangible: —Reputation —Difficulty placing new products with distributors • Product recalls: —Cost of recalling product, —Product replacement —Legal liability —Stock price decline (loss of capitalization) • Excess cost of production (because of off-target performance): —Scrap and rework —Cost inside the spec limits • Wasted development resources: —Rework in product development —Stabilization of new product —Projects proceeding past their "kill-by" date • Lost revenue opportunities: —Successful product with no follow-on —Product launched too late (competitive product appeared first) —Product overwhelmed by superior competitive product —Inadequate development pipeline	

Continued

Table 5.1 Examples of DFSS business case elements. *Continued*

Audience	Candidate reasons	My company's situation
Middle management (bilingual)	All of the above money reasons All of the below technical reasons	
Development staff (language = technical)	• Frustration with: —Poorly scoped development project —Lack of understanding what the customer really wants —Irrational product requirements —Inability to develop a product concept that focused on meeting customer needs —Having to invent a new technology in the middle of a development project —Being whipsawed with changing requirements or a product with ever-changing features —Inability to predict how well the product will perform in the face of process and supplier variation —Irrational risk management approach —Lack of knowledge and assurance that the product will be producible —Nights away from home solving production problems —Continually changing product requirements/product features list	

What did this research tell you? Can you put a case together for improving your product development process? Does this case convince you? How about others in your organization (try it out with a few friends first)? Will this case be good enough to sell your senior leadership? If the answer to these questions is yes, move on to the gap analysis. If not, stop reading this book and move to the next book on your list!

GAP ANALYSIS

The *why* questions tell you that something's wrong and/or that there's an opportunity to improve your current development process. The next key question is: What are the gaps between your current process and what "good" looks like?

General Electric's change management approach starts with a simple but powerful equation:

$$Q \times A = E$$

Here, Q stands for the quality of a change or decision, A for the acceptance of the decision, and E for the effectiveness of the change or how well it will "stick" in the organization. We refer to "Q-side" issues as the *hard stuff* and "A-side" issues as the *soft stuff.* Your gap analysis must include both sides.

QUALITY-SIDE GAPS

We'll start your gap analysis on the Q-side of the equation. You want the highest-quality new product development system you can create, so we'll ask you to start by identifying gaps in your current development system. (We'll define *gap* as simply a difference between what *is* and what *should be.*) What are the biggest issues with your organization's current product development system? Why are they issues?

There are two main factors you'll consider here: design and development processes and organizational structures. We'll focus first on identifying development processes that need an overhaul and then organizational structures that are impeding you from developing a world-class Six Sigma product development process.

Prepare to take a deep dive into your development processes. Don't just get a copy of your product development procedure; look at the methods each product development team actually uses. Look closely at each step of the product development process, identifying the gaps between current practice and best practice (review Chapter 4 for what "good" looks like). This is important. Be sure to identify true *current practice,* not what is documented as your process. We've found too often that the documented new product development process and the one being followed can be very different. We mentioned earlier how one company had pictures of their process posted in all the hallways, but the reality was that *nobody* followed that process. Your success in sustaining DFSS is to make sure in the end that the process you put in place and document is a process your teams *can* and *will* follow. It really helps to be a good detective here. Adopt the Missourians' show-me attitude and look for data-based evidence, even if that evidence isn't numerical.

For example, one team we worked with built a cardboard-and-string mobile that listed the critical customer requirements at the top and strung the cascading critical requirements level by level. The mobile hung over their team table, so every team member saw it every day, reminding them that every level of detail linked back to the customer. This team *got it.*

Other teams could show management their "progress" with fancy PowerPoint presentations, but these were often cobbled together the night before the briefing. These teams were just putting up a *storefront*.

GAP ANALYSIS: CURRENT NEW PRODUCT DEVELOPMENT PROCESSES

Before we do a deep dive into your development process, here are a few basic questions. Do you have a defined product development process? Do your product development teams do things differently, without defined steps, gates, or milestones? Is that OK or even *condoned* and *encouraged*?

Think about the main functions associated with each step of DMADVI:

- *Define*—Organizing the development project
- *Measure*—Obtaining VOC and developing product requirements
- *Analyze*—Developing the feasible concept
- *Design*—Developing the production product and production processes

- *Verify and Validate*—Verifying and validating product and process performance
- *Implement*—Launching and stabilizing the product

Even if you're not using a DMADVI-like process, how are the teams performing these basic functions today? Can you build a map of the process (a flowchart, a schematic, whatever your favorite process picture is)?

If you've got major gaps here, your DFSS strategy and implementation plan (see Chapter 6) will start lower on the maturity scale. Your first steps may be to install a basic development process with decision gates. Go ahead and use DMADVI then as your basic development process.

Gaps I've Found:

Now let's look at your product development process step by step to find your detailed gaps. Although we'll walk you through all the design functions covered by DMADVI, you should focus on the most important gaps. We'll mix metaphors here and suggest that you use the Pareto principle and not try to eat the elephant in one sitting.

How Robust Is Your Define, or Chartering, Phase?

Based on the DFSS roadmap in Chapter 4, how far off are your teams' processes from the "what good looks like" for the Define phase? Do your teams use a chartering process to get the development project going? Are they reviewed and approved by management? Do all key stakeholders (at least marketing, R&D, and operations) sign off on the charters?

Are some teams chartered to develop a product with a wink and a nod from a manager, while others in your organization clearly define each goal, deliverable, and a project timetable? Although this question relates more to your overall development portfolio, how big are the development projects (balancing the project's risk with its reward)? Are the development projects linked to your business strategy? Is that evident in the charters?

Go out and sample a few of your teams' charters. What do the charters look like? Do the charters include the detailed product features the teams are tasked with designing (red flag here)? Or are the charters focused on the unmet customer needs, allowing the design teams the space they need for true ideation (green flag here)? Are the charters up to date, or can you see a bit of mold around the edges? For example, how does the scope defined in the charter match up with what the team is working on? Take a charter and, without letting them see it, ask team members a few questions. Make them obvious, such as what are the goals, scope, and deliverables of the project?

Gaps I've Found:

How Robust Is Your Voice of the Customer Collection?

To assess your gaps here, start with the product of the VOC process—the team's product criteria or design specifications. First, do your teams refer to a list of criteria during product design? Are these criteria simply a list of features that are to go into the product, or are they real requirements? Here's the linkage—the customer has *needs,* the product has *requirements,* and the resulting design has *features*. If your customer would like to dig a hole quickly, the *requirement* for the product could be "customer can remove X amount of soil from the ground per minute." This statement of a requirement allows your development team the opportunity to consider different solutions. If they listed the requirement as "surface area of the shovel blade," they've (a) just defined a feature and (b) have locked themselves into a solution.

Is this list of customer requirements referred to throughout the design and validation/verification cycles? Does anyone check to see whether the requirements were met by the design? Are there data to back up these claims? How much confidence do you have in these data? What is the business decision-making process if there are gaps here? Do the teams go back and redesign the product or do they shrug their shoulders and say, "Oh, what the heck, we'll launch it anyway?"

Now, work backward to the source of the product's requirements. What process was used to develop the requirements? Do the requirements come from customer visits, interviews, and confirmation surveys that are traceable to the appropriate customer segments? How well have your past launches met your customer expectations? How do you measure that satisfaction? It is very important to take a deep dive into your product development teams and take an honest look into their approaches to VOC.

Third, look for how teams are prioritizing these requirements. Is there evidence that the Pareto principle has been applied; that is, has the team selected those few critical drivers/customer differentiators? Or are all requirements "created equal"? We have seen teams list all of their requirements equally and then go through endless redesigns until they "meet" them all. Teams waste months, if not years, designing the product to hit every requirement perfectly. With DFSS, a team can learn to differentiate between requirements, from the many that have to perform adequately to satisfy the customer to the critical few that need to outperform everyone in the market. The time saved by applying this differentiation can be the difference between being first to market with the next iPod and fighting to convert customers to a me-too product.

Finally, look at the customer requirements' stability from the start to the finish of projects. Do you see the criteria change as new players come on to the team? As we stressed in Chapter 4, a robust VOC collection method will stand the test of time through product development, and it will stand the test of team-member shifts as well.

Gaps I've Found:

How Robust Is Your Concept Development (Analyze) Process?

Do you see evidence that the teams are evaluating their concepts objectively, according to how well each concept will deliver on the CTQs? Do you see evidence that more than one concept is being evaluated? Can you see that the concepts have been refined and combined, torn apart and put back together again, until the team can show that they have the best combination of features rolled up into one superior concept? Or are teams jumping to one solution too quickly? Ask them to prove that they've evaluated other possibilities.

What feasibility studies are performed to assess the concepts? Do your teams use a build-test-fix approach or a design-analyze-confirm approach? Do you see the teams wasting time with unnecessary prototypes that don't answer key questions? Are your teams using modern feasibility techniques such as simulation methods like finite element analysis or the SPICE circuit simulator? How do your teams seek and incorporate customer input to the design concept?

How many of your teams are stuck in *science projects,* trying to invent the technology needed for the product?

Are your teams thinking ahead to the production processes and other Design for X considerations (manufacturability, assembly, environment, reliability, human factors, testability, inspectability, maintainability, and other -ilities)?

How are your teams identifying risks associated with the concept? Who does the failure modes and effects analysis (FEMA)? When? Are the risks prioritized? Is there an opportunity for the results of the FMEA to be fed back into the design? Ask the teams for evidence where one or more high-risk failure modes (identified from the FMEA) were then mitigated through a design change.

Is the end of the concepting phase defined clearly for each team, marked by the milestone of a single concept selected, risk analysis performed around that concept, and mitigation of those risks?

Gaps I've Found:

How Robust Is Your Design Cascade Process?

Take a look at your design teams' outputs. Walk through their team rooms or meeting areas. Do you see solid evidence that the team members keep their eyes on the customer? Or do you see evidence that each member is approaching his or her subsystem with a narrow focus, suboptimizing the parts instead of optimizing the system?

You may gather evidence about this gap further downstream in the design process; how many times have problems come up when subsystems are brought together during systems integration?

Do you see engineering analysis that starts with manufacturing and assembly variation and predicts the resulting system capability? This is the second-largest gap we see in organizations. In many organizations, the engineers develop flawless performance calculations of the most complex systems and subsystems, but they are calculated only to the nominal. Many teams then turned to the quickest method for building safety into it, the 2X safety factor. Review your teams' approach to safety factors. If you see safety-factor-of-two calculations, then that team may not really understand their manufacturing capabilities.

We found another curious practice. One team of electrical designers told us they used worst-case analysis to ensure that their circuits performed within specs. First, a worst-case analysis of more than a few components is very conservative. Second, the designers told us that it wasn't component variation that caused problems, but variation in assembly processes such as soldering! So they were doing a "useless" prediction that ignored the important sources of product variation!

A team that understands the manufacturing capabilities will be able to use the predicted variation in their calculations instead of simple safety factors or worst-case analyses that may or may not truly be "safe" or "worst."

Also look at prototypes built during DMADVI's Design phase. Why were the prototypes built? How many versions of prototypes were built? Ask the team what questions they were answering when they built each set of prototypes. If the answer is always "to see if it works the way we want," throw up a red flag. The DFSS approach will help teams learn to evaluate and predict "how well it will work" by variation analysis and deep-dive into the cascade. Are teams building each version they design in order to check the functionality? If so, the chances are pretty high that the team doesn't truly understand

Gaps I've Found:

the design cascade from system down to component and feature. They are "building and testing," a costly and time-consuming practice that DFSS will help minimize. On the other hand, a team that builds a prototype to verify an analytical model will get the knowledge they need up front, requiring fewer prototypes through their project.

How Robust Is Your Verification and Validation Process?

We defined *verification* as the ability of the product to meet its requirements. What process do the teams use to verify their designs? Can they show you evidence of verification activities for product-level requirements? If design predictions were performed, how does the data from the verification activities compare; are there large gaps?

How has the production process been verified to perform its fabrication and assembly functions? Are sample products produced with just nominal settings, or does verification also include testing at the process extremes?

How is the final product, produced under production conditions, validated with the customer? Can the teams show you evidence that the product meets the customers' needs?

In today's resource-strapped world, it is understandable to want to reassign the development team to a new project the minute the first commercial product hits the shipping docks. But the best thing you can do for your organization is to allow the team to evaluate their results and identify what they could have done better. The whole team needs to be held accountable to deliver the quality product and manufacturing process that they promised. If you see teams being disbanded before the launch and subsequent product validation, consider that gap a significant obstacle to improving your product development process. The feedback cycle should identify both problems that occurred during this particular development effort as well as any good practices that could be replicated in future efforts.

Gaps I've Found:

GAP ANALYSIS: CURRENT NEW PRODUCT DEVELOPMENT ORGANIZATIONAL STRUCTURE

As you've performed the process gap analysis, you've probably identified a number of organizational practices that will also require change. You should look for structures, policies, and practices that have outworn their usefulness. Here are a few structural issues you should consider in your gap analysis:

- *Development team organization*—Do you use dedicated development teams or are development resources shared across teams? (Note that we're not suggesting either of these approaches is wrong, but rather that you consider team organization as a potential gap.)
- *Development resource utilization*—How are your development resources being used? How long do they stay in their roles (too short or too long)? DFSS emphasizes a design-analyze-confirm approach. Do your development resources have time and skills to perform this work? Have a few developers log how they spend their time for a few days (first, explain why you are doing this). How much of this time is non-value-added—that is, time they spend in meetings, chasing information, making PowerPoint presentations, and so forth? Because you will be asking them to do additional work (at least, up front), what can you tell them they can stop doing?
- *Reward and recognition systems*—This can be one of the most important factors either helping or hindering your new product development improvement initiative. What practices, behaviors, and so forth are rewarded? For example, are bonuses based on achieving time-to-market goals or time-to-profitability goals? Do the reward and recognition systems suboptimize your development process? Take a close look at what motivates each player, especially marketing, R&D, and operations. When GE wants to change something in its organization, it seems very good at aligning its systems to reward the right behaviors.

Which of the current structures, policies, and practices will hinder or prevent you from driving DFSS through your organization?

For example, in one organization, marketing staff members are rotated on an 18-month cycle. To make his or her mark, the marketer needs to quickly come up with a product idea and drive that idea through the development process to launch. This behavior leads to product development projects that are started with little voice of customer input and that are mainly variations of existing product lines. The company suffers from a plethora of line extensions being launched with few truly innovative (and hence higher revenue/profitable) products. Until this rotation policy is changed, the company will have little success in improving its new product development process.

What structures need to be in place to allow robust product development, strategic decision making, and business-driven portfolio management? For instance, one company put into place a phase-gate system for business checks and balances two years before deploying DFSS. This proved to be a valuable first step in the success of DFSS. The phase-gate process provided that organization with a strong, cross-functional project management approach that held teams accountable throughout the development cycle.

Because this was a large development organization, it took two years to get used to the phase-gate process. Some organizations may be able to implement a phase-gate process at the same time as the DFSS process, but every company's tolerance for change is different. Evaluate the tolerance for change within your organization before you decide whether to deploy the phase-gate process and DFSS simultaneously.

How robust is your decision-making process? Do products get launched because the organization is too weak to kill a bad idea? Do teams feel penalized if they find that a product is not worth pursuing? Do you practice killing the messenger? Do products get launched because teams and sponsors have pet projects and don't want to stop driving the product even though the data point toward killing the project? The DFSS process works if only teams and sponsors let data and strategy drive decisions. If your organization has weaknesses in the decision-making process within product development, stop where you are and work toward fixing this problem before you spend any time on DFSS implementation.

What about the portfolio management process? How do projects enter the pipeline? Is there a robust strategic alignment around new product ideas or unmet needs that allows leaders to use data to determine which projects feed the product pipeline? Do the teams use multigenerational product planning to manage their pipeline? Without a robust portfolio management process, DFSS may greatly improve your design robustness, but your company will feel a loss of morale as you fail to see the market growth you expect from DFSS. DFSS will improve your product development process only from charter to launch. Getting the right product ideas or unmet needs fed into the pipeline is as important as DFSS itself.

Gaps I've Found:

TEAM-BY-TEAM GAP ANALYSIS

So far, we've focused on the common cause sources of variation in your new product development process. These sources will generally affect all of your teams. We've also found that there can be a lot of variation across each development team. We've found that one of the most effective DFSS implementation approaches is to *make it personal.* If one team has done an outstanding job of collecting voice of customer information but is struggling with concepting or cascade/prediction work, you should recognize and celebrate the former and then focus your help on the latter.

To help you examine your individual teams' performance and processes objectively, develop and apply an evaluation tool like the one shown in Table 5.2 for your organization. This tool will help you do an individual team gap analysis by assessing the maturity of each team relative to what "good looks like."

In our example tool, we've included three levels of maturity: *unacceptable performance, watch for trends,* and *best practice*. You may decide to be more or less granular, but we would suggest you don't go above five levels of differentiation.

Be careful here and assess your culture's tolerance for *labeling*. Some cultures may be fine with a 1-through-3 rating, or 0 to 100%. Some may be tolerant of red–yellow–green coding. Don't underestimate the importance of your organization's cultural norms before using this assessment. The goal of this assessment is to gain the alignment of the teams, agreeing that there is improvement to be made. Your goal is to simply hold up a mirror for these teams to look objectively at themselves. Let the data do the talking.

Share the results with each team immediately. This sharing is the real heart of the A-side of the equation. Once teams see how they look in the mirror realistically, they usually want to change that reality. If you are objective in your evaluation of each team, you may find teams actually coming to you to request that you evaluate them next.

PRIORITIZE THE DEVELOPMENT PROCESS IMPROVEMENTS

Some organizations may have the willpower and resources to fix many of the DFSS gaps at once. In our experience, however, most companies do not have the resources or the patience for an extreme makeover. Instead, we've found that focusing on the most pressing process failures and structural gaps is the most effective way to proceed. This will allow the DFSS deployment team to fix the biggest gaps first and show the organization early and significant improvements. From a $Q \times A = E$ approach, you'll gain traction with the organization as you fix the biggest "pains" experienced by your new product development teams.

In Chapter 6, we'll ask you to create a DFSS implementation charter and deployment plan. By now, you've gathered just about all the information you need to do this, including those gaps you are going to fill in the organization first. This will help you maintain the quality of your change. You can now focus on the critical elements to fix: those elements that the organization can't help but agree are broken. With the individual team gap analysis tool, you also have a mirror that you can hold up to any team or department to help them see what they look like.

"A-SIDE" GAPS

Having defined and prioritized the "Q-side" gaps, you, the development leader also need to consider "A-side" issues as you plan implementation of DFSS-driven improvements. Don't do a "metric system." Most people recognize that the metric system has a lot of benefits over the English system used in the United States. However, general acceptance of the metric system stalled in the United States a

Table 5.2 Example of DFSS preevaluation criteria.

	Define	Measure	Analyze		Design	Verify/validate and implement
	Preliminary	CTQs	Concept selection	Cascade to transfer	Scorecard	Scorecard
	Charter	Defined	Re: CTQs, cascade	Functions	Predicted capability	Actual capability
Best practice	Charter aligns unmet customer needs, strategic plan, and market size.	CTQs are aligned with unmet customer needs and are new, unique, or difficult to fulfill. CTQs are measurable with targets and ranges.	Multiple concepts are evaluated. Diligence is shown around risk analysis, concept feasibility, and prediction around the ability to perform to the CTQs. Trade-offs can be articulated to show the optimal concept was selected.	Transfer functions have been developed that link all the CTQs to the critical parameters (i.e., subassembly, components). Every CTQ is cascaded through mathematical analysis.	Transfer functions and variation analysis are performed to predict the capabilities of the CTQs. Components can be clearly linked as critical drivers based on the variation analysis and transfer functions.	CTQs are tested to verify the actual performance capability. Critical components tested to verify the actual capability of each critical feature. Any gap is identified and accounted for, and design is reoptimized if necessary, with clear under-standing of what the drivers were to optimize the design.

Continued

Table 5.2 Example of DFSS preevaluation criteria. *Continued*

	Define	**Measure**	**Analyze**		**Design**	**Verify/validate and implement**
	Preliminary	CTQs	Concept selection	Cascade to transfer	Scorecard	Scorecard
	Charter	Defined	Re: CTQs, cascade	Functions	Predicted capability	Actual capability
Watch for trends	Mature technology to be used is defined. The value proposition points to market research and/or a multi-generational product plan.	All of the CTQs are solution-free.	Only two or three concepts were evaluated, with little or no difference between the concepts.	Cascade is shown, but links are weak and anecdotal.	CTQ scorecard is complete with predictions.	Scorecard is complete with capability analysis, measurement system analysis for each CTQ, and each critical component. Verification plan shows capable measurement system.
Unacceptable	Team is chartered to solution, with no real design/ innovation room allowed.	No CTQs are identified. All requirements are treated equally.	Concept shown is the only concept the team can define. No proof of other concepts at all.	No cascade can be shown.	Component score-card is complete with predictions for critical component features that have been identified by the transfer functions and variation analysis.	No scorecard is shown.

number of years ago. As you create the structure and implement the change to DFSS, gauge the acceptance level in your organization.

There are two main elements to consider when predicting the acceptance of change. Leadership commitment is the most critical element, followed by the tolerance of and motivation for change.

LEADERSHIP COMMITMENT

It takes more than one senior leader to drive DFSS (even if that senior leader is someone in a position like Jack Welch at General Electric). You must gauge the commitment of each leader in the organization. Some leaders may just need to be verbally supportive of DFSS, while others will need to actually make things happen in their departments.

Note that we're using the term *leader* here, not *manager.* Not all managers are leaders, and not all leaders are managers. Who do your people really look to when leadership is needed? Of approximately 20 managers in FPL's Power Plant Engineering department during the 1980s, we'd count only two as leaders. Some of these managers were even anti-leaders, resisting change from any direction. There were, however, another six or seven engineers in the department who were truly leaders, even though they did not have the word *manager* in their titles.

Deploying DFSS takes strong leadership at all levels of the organization. You should assess each leader's position as a stakeholder and determine whether they are 100% behind the drive for DFSS, whether they are merely passively supportive of the change, or whether they may work to torpedo DFSS. In addition, you should consider where they need to be for the change to be successful.

Table 5.3 is an example of a *stakeholder analysis*. Your DFSS implementation team can use it to analyze and track the commitment of your leadership. They can also use it to develop a plan to move each leader to the necessary level of commitment.

This example shows just a subset of stakeholders you will actually need to evaluate. Presidents, vice presidents, directors, and other managers who have strong influence across the business should be interviewed to explore their concerns and commitment to this important change. Also include the informal leaders in your business. Are there external stakeholders you need to include? For example, some regulatory bodies, such as the Food and Drug Administration, are very interested in the new product development process. Some of their regulations promote DFSS thinking; others do not. After an

Table 5.3 Example of stakeholder analysis.

Key: X = Stakeholder is here O = Stakeholder needs to be here						
Stakeholder VPs	Against	Neutral	For	Concerns/Details	Actions	Responsible
Mark Martin, VP of R&D		**X**	**O**	Understands the worth of DFSS, but is concerned about the resources/commitment needed.	Share success stories from other organizations. Let Mary Jones tell the story herself. Calculate the cost to deploy DFSS and estimate the savings to show return on investment.	Mathews
Kim Reynolds, VP of marketing		**X**	**O**	Concerned that it is just another “flavor of the month.”	Share success stories from other organizations. Let Mary Jones tell the story herself.	Chen
Mary Jones, VP of operations			**X O**	Has seen it work in prior companies.	Get details/success stories from Mary Jones to share.	Connor
Joe Patel, director of X franchise	**X**		**O**	Thinks his organization is already stretched with the recent changes and upcoming launches.	Share success stories from other organizations, and estimate the cost of not deploying DFSS, in both time and money.	Thomas

introduction to Six Sigma, a chemist from a pharmaceutical company told us point-blank that the organization would never be able to achieve Six Sigma performance. When we asked a cautious *why,* she explained that they set their specification limits for the process by producing three registration batches of product, calculating the mean and standard deviation, and then setting the specs at +/− 3 standard deviations. If they reduced the variation in the process, that would result only in a tightening of the spec limits! She stated that this was the way the FDA forced them to do business!

Keep in mind that not every leader will need to be moved to the "for" box on the stakeholder chart. Some leaders with minimal influence may be fine in the "neutral" box. Others may not need to be moved from the "against" box.

For example, one vice president might be assessed as "against" the change right now, but you know that the vice president is retiring in three months. You know you will take the next four to six months creating the DFSS structure before you even roll out the change. While this organizational role might be critical, your energy will best be used in communicating with the executive who will take that vice president's place.

If you complete your stakeholder analysis and find that fewer than half of the leaders are "for" the DFSS drive, you may want to rethink your timeline for implementation. As you will read in the "Don'ts to Avoid" section of Chapter 6, a grass-roots approach to a change this great in scale is not likely to work. Your team's time would be better spent bringing the critical subset of leaders further up the commitment curve before embarking on the wider DFSS implementation strategy.

Make sure you spend time understanding why people are for or against the change represented by DFSS. During Florida Power & Light's quality journey, John went through his own roller-coaster-like change experience. In late 1985, he was sent to FPL's team leader training, a one-week course that today would be equivalent to a Green Belt Lite course. Leaving the course, he was excited by the idea of using facts and data to solve business problems. If you asked him to help you implement DFSS when the course ended that Friday, he would have given you an enthusiastic "yes!"

The following Monday, though, he was assigned by his manager to lead a team. Unfortunately, instead of providing a *problem* to solve, his manager gave their team a *solution* to investigate. The team was quickly frustrated; they didn't even know what the problem was! And John was caught in the middle. The manager kept demanding progress on the solution while the quality improvement folks kept demanding that he and his team follow the problem-solving method—a no-win situation. Eventually, the team was disbanded as a failure. After this experience, John wanted nothing to do with the quality improvement program. If you asked him to help you implement DFSS then, you would have gotten a very negative response.

A turnaround occurred when he was selected to be one of FPL's 13 statistical specialists (today, a Master Black Belt). At first he resisted, but fortunately (for John), there was no choice in the matter. He was then exposed to people in FPL who were using the quality process to solve problems and achieve better business results. Slowly, his attitude changed and he became a born-again promoter of the quality process, and today it is his passion. Today, once again, your request for help would get an enthusiastic "Yes!" So, we hope the lesson is clear: Make sure you understand "why."

What should you expect in terms of leadership behaviors? Here are some examples of good behaviors:

- *Resourcing the DFSS implementation*—Leadership will send a strong signal about their DFSS support when they select the DFSS implementation team members and resource this team. This behavior applies in general to how the DFSS initiative is resourced. For example, when one team leader became excited about the benefits of variation prediction, he ordered 15 copies of a Monte Carlo simulation software package, one for each of his development engineers.

- *Personal involvement in the change*—This can take many forms. We've mentioned the R&D leader who scheduled charter review sessions that represented about one day per month of involvement for him and his team. One R&D leadership team conducted periodic DFSS assessments of each development team; this helped them both gauge team progress and learn the DMADVI process. Even having a senior leader kick off DFSS training sessions is appreciated.

- *Willingness to change their habits*—Let's face it, many of the *gaps* you're trying to improve were created because of the design system set up by current and past leaders. A big A-side win will involve leaders changing their expectations. For example, one organization's leadership expected to see early physical prototypes of new products. The DMADVI method adopted by this company now focuses on obtaining product requirements first, and then developing and optimizing virtual concepts. This organization's leaders fortunately developed the patience to let the teams follow the process. Now the team can show up at the charter review without a prototype. In another case, an organization overloaded itself with product development projects numbered in the hundreds! Here, a new leader diagnosed this as a key gap. He soon decreed that the organization was not allowed to have more than 30 development projects on the books at any one time. He also was clear about his strategy for the organization, including revenue and profit goals for new product launches. This resulted in a portfolio of projects that had a much higher value per project. Many of the low-revenue line-extension projects were canceled, freeing up resources to work on fewer but better projects. Although this action could be considered a bit draconian, the message was certainly clear.

- *Willingness to lead the change*—It's often perceived as risky to change what we're asking subordinates to do, especially if we're not quite sure of the benefits. A while back, John was the guinea pig for a change to Florida Power & Light's engineering scoping process. Sid, the project manager, came to John's office, explained the change and why he was asking for the change. Although John was a bit dubious of the benefits of the change, Sid convinced him to give it a try. John struggled a bit with the new scope form but completed a draft. Sid then helped John make a few corrections, approved the form, and sent it along to the customer. A few days later, the customer gave Sid positive feedback on the new scoping form, which Sid passed on to John. Word got around, and pretty soon other engineers were stopping by John's office to get help completing the form. In this case, Sid had not only personally led the change, but he had also co-opted John for help and support in rolling the change out to the whole organization. Sid could have saved some of his time by just writing a memo telling the organization that the new form was required, but that would have been an ineffective change method (remember, $Q \times A = E$).

- *Playing nicely together*—We've seen situations where the heads of R&D and manufacturing are solidly behind the DFSS initiative, but marketing is missing in action. This may be the most difficult good behavior, but these three functions should all be supporting DFSS. Leadership signing up to review and agree on the results teams achieve in each DMADVI phase can be evidence of good behavior. At the beginning of each phase, the triad of marketing, R&D, and operations agree on the development team's deliverables and also on the path going forward.

TOLERANCE AND MOTIVATION FOR CHANGE

Your organization's tolerance and motivation for change is the other key A-side issue you will have to face. One of your key risks to address is *low commitment* to your DFSS initiative. This can occur either because of flaws inherent in the DFSS implementation plan or as a side effect of other poorly planned or executed changes. We'll help you as much as we can with the former, but, whatever you do, don't underestimate the latter! It may not be your initiative the organization is apathetic toward, or fighting against, it's that DFSS is *another* change in an already change-overloaded world.

We'll freely admit that, when younger and a bit more immature in the ways of organizations, we either ignored or didn't pay appropriate attention to the tolerance and motivation factor. Even Dr. Joseph Juran notes that he didn't understand this factor until he read Margaret Mead's *Cultural Pattern and Technical Change.* He drew a parallel between the resistance that United Nations teams faced working with developing countries and the resistance he saw between management and labor, or between consultants and clients.

Has your organization been "doing all right" for a number of years? Is there a motivation to change? Or is your organization so threatened by competition that everyone is running scared seven days a week? What is the fear level in your group? The motivation may be there, but fear can paralyze the very people you need to help implement DFSS. These are just a few of the many questions you need to ask as you assess the readiness and tolerance for change in your organization.

Start your assessment of this factor by looking back at some of the change efforts that your organization has promoted over the last few years. We know that with all the stresses to improve the bottom line, almost all companies have been going through a tremendous amount of change turmoil.

Total quality, reengineering, outsourcing, Y2K, downsizing, SAP implementations, Six Sigma, and never-ending reorganizations are a few to which we've been exposed. What were your last few major changes? How were they implemented? How were they received? Has your organization been through a major change every year for several years? Are the associates still complaining about the past changes? Do their heads still spin from the last change?

Some of these changes may have gone well, others not. Question and research how well the changes were planned and implemented. Survey the associates in your organization. Ask them what they thought of a few of the most recent changes. Ask them whether they think the changes were critical to the success of the business, or just change for the sake of change (we know one R&D organization that seems to reorganize about every six months, whether they need it or not). Ask them whether they think the changes were well planned and implemented, or just hurriedly implemented as if to check a to-do box.

Take a close look at the associates in your organization. Are they exhausted from their efforts to follow the latest change? Have the changes in the past been followed up with data to show their successes? Have these changes even stuck? Or did the organization revert back to the old way after facing too much resistance in the last attempts to change the organization? Ask the leaders of the previous changes to articulate their successes, as well as their lessons learned. Take advice from the old song and accentuate the positive learnings and eliminate the negative learnings. Table 5.4 shows a simple template you can use to evaluate past initiatives.

Try to gauge whether your organization is ready for the changes associated with DFSS. We worked with one company whose board brought in a slash and burn CEO. In his first year, all employees were told that they were fired (the only one who had a job was the CEO). The CEO then picked his direct reports, who picked their staff and so on based on a new organizational structure designed by the CEO's consultant. The remaining employees were put in a pool, where they were given a month to see if they could find a job in the company. If they couldn't, they were laid off. One of the managers told us that it took two years before his engineers got the courage to try new things; they had all become extremely risk averse. These two years would not have been a good time to try to implement DFSS.

Learn from the mistakes of previous changes so that your implementation team can create a change model that really works.

Table 5.4 Change initiative assessment form.

Change Initiative	
Purpose of Initiative	
What Worked	What Didn't Work
Why?	Why?
Learnings for DFSS Initiative	

Evaluate the weaknesses of your organization and either strengthen them or build around them to ensure that your change doesn't crumble atop those weak pillars. The effort needed to implement and sustain a culture of DFSS is much too great to ignore the resistance.

PLANNING FOR CHANGE

Once you learn how your organization has adjusted (or not) to recent changes, you can make the plans appropriate for your DFSS implementation. If you ignore the history of the organization, the pain it has seen in recent years of change, you are destined to repeat the same mistakes. Take the time to interview a cross-section of folks in the organization. Ask them about recent changes they have seen or implemented (be sure to ask leaders of previously implemented changes for their honest hindsight). Take note of the reasons behind any changes that didn't stick. Then make a plan for your project to counter those influences. Table 5.5 is an example of a change-management analysis. It shows how recent changes failed and what the implementation team will plan to do to make sure this DFSS deployment doesn't fail.

Once you interview a representative cross-section of your organization, you can complete a similar change-management analysis and make the appropriate plans to make the new DFSS process truly stick. A bonus you will get from interviewing people from across your organization is that these folks will get a positive feeling because you are listening to them. They will talk within their subgroups about the impending change, and you get a lot of free advertisement.

Table 5.5 Change-management analysis example.

Top reasons for recent failures (from interviews across the organization)	Steps to avoid failure in this implementation	Responsible team member
It wasn't on my performance plan to change how I did things.	Work with leaders to include new behaviors on performance plans.	Mathews
Some sponsors still wanted to see things the old way.	Educate sponsors first; show them what to expect differently, and show the benefits of the change.	Chen
The policies tell me to do things the old way, and they *have* to be followed, so I don't do things the new way.	Change the policies and procedures to reflect the new process we're putting in place. Train all levels on the new procedures.	Chen
It is just extra work for me, and I don't see any benefit! So why would I waste my time?	Communicate each team's success and benefit (for example, one team that saves $70,000 on prototyping costs; another saves two months in development time).	Connor
I did not have the right people on my team to truly make the change stick. I just did the best I could before I closed out the project.	Ensure a project charter with timeline and team members who will be on the project through completion.	Thomas
My leaders were eager to move me on to the next project, so I left as soon as I could.	Identify metrics that will indicate when the DFSS implementation project can be considered complete.	Mathews

KEY POINTS: DFSS GAP ANALYSIS

This chapter takes you through DMAIEC's Define, Measure, and Analyze steps for your DFSS implementation. By now, you should understand why you want to pursue DFSS, you understand the current development processes in use, and you have performed a gap analysis that tells you which are the most pressing gaps in your organizational structure or your product development process and why these exist.

Your next steps will be to identify, plan, and execute process/structural changes to close these important gaps. Chapter 6 will guide you through the Identify and Execute steps, where you can develop the best DFSS deployment plan and design it to fit your culture. Chapters 7 and 8 will help you complete the Execute step and move into Control phase so you can measure your success and transfer the processes to a sustaining team or structure.

Glossary

analysis of variance (ANOVA)—An analytical process that identifies the contribution of each individual input factor to the overall output response. (Chapter 2)

axiomatic design—A theory by Dr. Nam Pyo Suh that stresses that each functional requirement be designed to achieve robustness without affecting other functional requirements. (Chapter 10)

benchmarking—The act of seeking out relative processes to compare your methods against. The goal of benchmarking is to find a new standard to live up to, whether it is inside your industry or not. For example, a laundry detergent manufacturer might benchmark other consumer products for advertising best practices or other chemical companies to benchmark manufacturing and quality-control processes. (Chapter 5)

best practice—The "new standard" to live up to. The best current set of processes or products you have found to compare to, measure against, and learn from. For example, Motorola is known as a best practice in Six Sigma deployment. And Toyota is known to have best practices in automotive manufacturing. (Chapter 5)

Black Belt—Typically a person assigned full-time to lead Six Sigma (DMAIC, lean, or DFSS) projects that reach out past his or her field of expertise. Black Belts are also typically assigned to mentor Green Belts. (Chapter 2)

build-and-test—A traditional method of designing products that results in multiple iterations, low levels of actual learning from each iteration, and wasted design time. Build-and-test methods can be replaced with predictive design and software-based simulation in best-practice design organizations. (Chapter 4)

cascade—A flow-down of requirements from the product level down through the subsystem level and component level to the process level. The cascade allows us to link requirements from the top down, as well as from the bottom up.

Typical cascade: Performance requirement → subsystem requirement → parts requirement → process requirement → process variable. (Chapter 2)

cause-and-effect analysis—The process of identifying the likely causes of any outcome. Some cause-and-effect analyses produce an output like a fishbone diagram. Others produce a cause-and-effect tree, with multiple sub-branches. (Chapter 4)

concept feasibility—Predicted ability to fulfill performance level requirements with considerations of variations of customer use, balancing of performance-level requirements, and manufacturability. (Chapter 4)

concept selection—The selection of the concept that best fulfills the performance-level requirements. Clamping concepts could include mechanical (spring-force, linkage), electrical (electromagnetic), and biologic (finger compression). (Chapter 4)

control charts—Charts that track a process input (or output) over time, showing the trends and tracking the tendency to produce a failure. Control charts use statistical methods to predict the trend to failure in time to fix the process, minimizing the traditional desire to tweak the process. (Chapter 4)

control plans—Principle: If you keep the critical X-parameters under control, the outputs (Ys) will also be under control. Follow-up of Ys outcomes: Identify the difference between signal and noise. Define actions: When to take action and what actions you should take in case of parameters moving out of control. (Chapter 4)

cost-benefit analysis—An analysis of any change to identify whether the costs outweigh the benefits. All costs, both financial and intangible, and all benefits are weighted in a cost-benefit analysis. (Chapter 4)

Cpk (short-term capability index)—A measure of short-term process capability that takes into account both the process center as well as the variation. It is the minimum of {CPU,CPL}. (Chapter 2)

critical parts factors (CPD)—Dimensions, material, or surface properties of parts responsible for achieving the target and/or allowable variation associated with critical-to-quality product requirements. (Chapter 4)

critical process variable (CPV)—A variable associated with the manufacture or assembly of the product that is responsible for achieving the target and/or allowable variation associated with critical-to-quality product requirements. (Chapter 4)

critical to quality (CTQ)—The select few requirements that your team can focus on to deliver a differentiated and superior product for your customers. Must be measurable, with targets and ranges. The recommended method of determining CTQs is to identify performance-level requirements that are linked to VOC and are new, unique, or difficult to fulfill (NUD). For example, a vehicle must provide 30 miles per gallon of regular gasoline or better. Target 35 miles per gallon, no upper limit. Or a drill must provide 50 inch-pounds of torque or higher. Target 60 inch-pounds, upper limit 80 inch-pounds. Drill requires less than 40 pounds of force to actuate. Target 30 pounds, upper spec 40 pounds, lower spec 15 pounds. (Chapter 2)

defect—Any event that does not meet a customer specification (must be measurable). (Chapter 2)

defect opportunity—A measurable chance for a defect to occur. For example, an assembly with three components and two connections might have five opportunities for defect (one for each component, and one for each connection/assembly step). (Chapter 2)

design of experiments (DOE)—A structured approach to experimenting, effectively and efficiently exploring the cause-and-effect relationship between numerous process variables (Xs) and the output or process performance variables (Ys). Designed experiments can produce a useful transfer function to identify the relationship between inputs and outputs. (Chapter 4)

design freeze—A situation that is declared when:

- Engineering analysis is complete.
- Tolerance analysis budgeting is complete.
- Product testing specifications are identified and testing methods are defined.
- Design cascade on CTQs is complete with transfer functions.
- Tunable features are identified.
- Risk analysis is complete. (Chapter 4)

design for manufacturability and assembly (DFM/A)—A collection of principles and design analysis techniques aimed at ensuring the product can be produced at a minimum of cost, complexity, and defects. (Chapter 4)

DMADV—Phases of DFSS used by GE and many other companies. Phases are: Define, Measure, Analyze, Design, and Verify. Variations of DMADV include DMADVI, DMADI, and I2DOV. (Chapters 4 and 11)

failure modes and effects analysis (FMEA)—A process used to identify the potential failures of a design, process, or equipment. By identifying all the ways something can go wrong, the team can create a robust design with all misuses and other failure modes accounted for. (Chapter 4)

fault tree analysis (FTA)—Systematic logic diagram used to explore causes of failure or to identify factors (CPF and/or CPV) responsible for achieving critical-to-quality product requirements. (Chapter 4)

finite element analysis (FEA)—A design tool to model stresses, heat flow, or other physical phenomena in a design. (Chapter 10)

flow-down—The design act of allocating targets and specifications associated with product requirements/CTQs to lower-level (such as subsystem or parts) requirements. (Chapter 4)

flow-up—The design act of predicting the capability of a design to achieve product requirement targets and specifications as a function of parts/process variation. (Chapter 4)

gage R&R (repeatability and reproducibility)—A method to assess the repeatability and reproducibility of a measurement system. Measurement methods can contribute a large percent of the observed variation on quality characteristics. Gage R&R allows us to measure the extent of this variation and alerts us if we need to take action to reduce variation caused by measurement methods. Typical gage repeatability and reproducibility $< 10\%$ of process variation. Gage R&R $< 10\%$ of process tolerance. (Chapter 4)

gemba visit—"Place of work." A gemba visit is a method of obtaining voice of customer information that requires the design team to watch customers use the product in their environment. (Chapter 4)

Green Belt—Typically a person assigned part-time to lead Six Sigma (DMAIC, lean, or DFSS) projects that are fairly narrowly scoped to match the Green Belt's field of expertise. (Chapter 2)

histogram—A graphical display of the frequency distribution. The height of each bar represents the number of respondents in each category. The histogram shows whether the distribution is approximately bell-shaped, U-shaped, or skewed. (Chapter 2)

house of quality—The name given to any of several matrix-based analyses performed to translate customer requirements into product requirements or to lower-level requirements. (Chapter 2)

hypothesis testing—A statistical test that quantifies the validity of a statement. A statistical method to test a hypothesis by comparing the data to values predicted by the hypothesis. It may be used to detect differences between groups. For example, a hypothesis test may be used in consumer testing to determine whether there is a significant difference in consumer preference between two or more products. (Chapter 4)

independence day—The day (or week or month) on which you can verify that your deployment team has succeeded in implementing DFSS into your organization and you can transfer to a sustaining mode. (Chapter 8)

Kano survey—A voice of customer method used to distinguish which customer needs are must-be, one-dimensional (or "more is better"), or delights. (Chapter 4)

Master Black Belt—Typically a person assigned full-time to train and mentor Black Belts as well as lead the strategy to ensure the projects chartered are the right strategic projects for the organization. Master Black Belts are usually the authorizing body that certifies Green and Black Belts. (Chapter 2)

mean—The average value of a sample of data. (Chapter 2)

measurement system analysis—An analysis that determines the dependability of your measurement system, revealing how much variation exists in the measurement system, either from the measurement device itself, from the humans operating the device, or from a random effect. This ensures that the conclusions drawn from the data are valid. A typical measurement system analysis is the Gage R&R. (Chapter 4)

Monte Carlo simulation—The predictive simulation of an outcome created by identifying the variation and distribution of each of its inputs. A Monte Carlo simulation predicts the variation you would expect to see in your output. (Chapter 4)

multigenerational plan (MGP)—A planning tool to determine a series of product releases and associated technology requirements based on a market- and business-driven vision of the product's "direction." (Chapter 4)

new product development (NPD) process—The process of developing products or processes: Defining the opportunity; Collecting and Measuring the voice of the customer; Creating, Analyzing, and Selecting the best concept; Designing; and Verifying that all requirements and CTQs have been met. (Chapter 3)

NPD project charter—A charter that clearly defines what the new product development (NPD) team will be charged to accomplish, which customers will be satisfied by the new product, and what those customers are missing from today's options. (Chapter 4)

optimization of product and process—Adjusting the inputs to where the performance requirement targets are met and are least sensitive to variation. (Chapter 4)

parameter estimation—The process of estimating the variation of each factor in a design. (Chapter 4)

Pareto analysis—A process used to identify the factors that are most responsible for the outcome. A Pareto analysis will show the largest contributors so you can focus on those factors. (Chapter 4)

phase—A period in new product development that specifies certain activities. In the process we describe in this book, the phases are Define, Measure, Analyze, Design, and Verify. (Chapter 2)

prediction—The design act of calculating the capability of a design to meet the targets and specification limits as a function of parts and process characteristics. (Chapter 2)

probability distributions—The histograms showing the distributions of a population or sample and the probability of delivering outside of the upper or lower specification limit. (Chapter 2)

process capability—The ability of a process to deliver according to the requirements. Often measured in terms of Ppk or Cpk, process capability can also be measured in terms of Sigma level. In a normal process, the process capability can easily be converted from Cpk to Sigma level (Z).

$$Z = 3\ \mathrm{Cpk}$$

Example: A process running at a Cpk of 2 can be said to be producing at Six Sigma. (Chapter 4)

product-level requirement (or simply product requirement)—A characteristic that can be directly traced to customer and procedural requirements; it also includes requirements necessary for developing and delivering the product. (Includes performance measures, appearance, packaging requirements, human factors, and so forth.) Examples include vehicle mileage, torque delivered through a tool, and force to actuate a tool. (Chapter 2)

production process design—Preparing the product for commercialization. This includes DFM/A analysis, capacity analysis, and tooling plan. (Chapter 4)

Pugh matrix—A qualitative decision-making tool used to assess the strengths and weaknesses of design concept relative to customer and business criteria. A Pugh concept selection matrix can help a team select the best ideas from among many alternatives, using objective criteria. (Chapter 4)

quality function deployment (QFD)—A method used to translate voice of customer information into product requirements/CTQs and to continue deployment (for instance, cascading) of requirements to parts and process requirements. (Chapter 4)

radar charts (spider diagrams)—Graphical outputs typically used to represent the gaps between importance and satisfaction of any requirement. (Chapter 4)

reliability testing/accelerated testing—A set of tests designed to understand and predict the failure rate of a system or component. There are many ways to perform reliability testing, including accelerated testing and worse-conditions testing. (Chapter 4)

response surface methods (RSM)—A collection of statistical techniques used to optimize a dependent variable (for instance, a product requirement/CTQ) as a function of one or more independent variables. (Chapter 4)

sampling—When collecting data, it is a good idea to have a sampling plan. A good sampling plan includes a properly estimated sample size and steps to ensure that the sample is representative. There are several approaches to sampling, including random sampling, stratified random sampling, systematic sampling, and subgroup sampling. (Chapter 4)

scorecard—A summary of the predicted and measured capability/reliability of critical requirements at all levels defined within the cascade. Examples include CTQ: Force to actuate $<$ 40 lb (USL) @ 4 Sigma; capability: Mean force $=$ 30 lb; standard deviation $=$ 2 lb; predicted Sigma level $=$ 5. (Chapter 4)

sigma(s)—Standard deviation. The English letter s is the standard deviation of a sample. The Greek character sigma (σ) is reserved for the standard deviation of the entire population. (Chapter 4)

Sigma level (Z)—A measure of the capability of a design. Specifically, the Sigma level is the number of standard deviations that fit between the sample mean and the specification limit. *Six Sigma* is the term used in a centered normal process when 6 standard deviations fit between the process mean and the specification limit. (Chapter 4)

Defects per million opportunities	Sigma level
66,807	3
6,210	4
233	5
3.4	6

Six Sigma—A term used when a normal process is performing centered at 6 standard deviations from the specification limit. A Six Sigma process can be predicted to produce 3.4 defects per million opportunities. (Chapter 2)

standard deviation—A statistical measure of the spread of the data. It is the square root of the variance. Symbols to represent the standard deviation are *s* for the sample standard deviation and *s* for the population standard deviation. (Chapter 4)

$$S^2 = \sum_{i=1}^{n} \frac{(x_i - \overline{x})^2}{n-1}$$

supplier-input-process-output-customer (SIPOC)—A high-level process map that depicts the suppliers that provide the input, the input they are providing, how the process operates, the output from the process, and the customer of the output. It is used as a communication/visualization tool so that everyone has the same understanding of the process and a key analysis tool that is a starting point for risk analysis, identifying sources of variation, simulation, and process improvement. (Chapter 4)

stakeholder analysis—Understanding the key stakeholders in your organization and identifying what information they need and "what is in it for them" if you implement a change. Making an action plan to bring each stakeholder up to the level of support needed for a successful change. (Chapter 5)

systematic innovation technique (SIT)—A technique that helps teams solve design problems or create ideas by using a systematic process designed to build the maximum amount of useful ideas. (Chapter 1)

Taguchi robust design—Setting appropriate parameters in the design in order to optimize not only the output, but also the variation. (Chapter 10)

tolerance design—Setting appropriate tolerances on process parameters and critical characteristics of raw materials to achieve a good process capability on the quality characteristics. (Chapter 4)

transfer function—A quantifiable relationship between inputs and outputs. There are three methods of developing transfer functions: (a) explicit mathematical equation, (b) computer simulation modeling, and (c) experimentation, or combinations of (a), (b), and (c). A transfer function is also used to predict the variability of the outputs. (Chapter 4)

For example:

Sleeve Bend Strength

$$M = Ultimate\ Stress \times \frac{l}{C}$$

$$I = \frac{\pi}{4}(OD^4 - ID^4),\ C = 1/2 \times OD$$

TRIZ—Russian acronym for theory of inventive problem solving, a systematic means of inventing and solving design conflicts. (Chapter 4)

voice of the customer process—A broadly defined process for collecting the customers' needs and reactions. For detailed process to collect and confirm the VOC, see *Voices into Choices* by Christina Hepner Brodie or *Customer-Centric Product Definition* by Sheila Mello. (Chapter 2)

voices of the customer—Things you hear directly from the customers about their needs or actual measurable requirements. For instance, "I don't want to go broke driving the car!" or "I want the tool to do the tough jobs for me." (Chapter 4)

References

Bertels, Thomas (Ed.). (2003). *Rath & Strong's Six Sigma Leadership Handbook*. New York: Wiley & Sons.

Breyfogle, Forrest W., III. (1999). *Implementing Design for Six Sigma*. New York: John Wiley & Sons.

Brodie, Christina Hepner, and Gary Burchill. (1997). *Voices into Choices*. Madison, WI: Joiner.

Clausing, Don. (1994). *Total Quality Development—A Step-by-Step Guide to World-Class Concurrent Engineering*. New York: ASME Press.

Cohen, Lou. (1995). *Quality Function Deployment—How to Make QFD Work for You*. New York: Addison-Wesley.

Cooper, Alan. (2004). *The Inmates Are Running the Asylum—Why High-Tech Products Drive Us Crazy and How to Restore the Sanity*. Indianapolis, IN: SAMS Publishing.

Cooper, Alan, and Robert Reimann. (2003). *About Face 2.0—The Essentials of Interaction Design*. New York: John Wiley & Sons.

Cooper, Robert G. (2001). *Winning at New Products*. New York: Perseus.

Creveling, C. M., J. L. Slutsky, and D. Antis Jr. (2003). *Design for Six Sigma*. Upper Saddle River, NJ: *Pearson.*

Drucker, Peter F. (1985). *Innovation and Entrepreneurship—Practice and Principles*. New York: Harper & Row.

Goldenberg, Jacob, and David Mazursky. (2002). *Creativity in Product Innovation*. Cambridge, UK: Cambridge University Press.

Goldratt, Eliyahu M., and Jeff Cox. (2004). *The Goal—A Process of Ongoing Improvement*. Croton-on-Hudson, NY: North River Press.

Huthwaite, Bart. (2004). *The Lean Design Solution*. Macinac Island, MI: Institute for Lean Design.

Mello, Sheila. (2003). *Customer-Centric Product Definition*. New York: AMACOM, American Management Association.

Rantanen, Kalevi, and Ellen Domb. (2002). *Simplified TRIZ—New Problem-Solving Applications for Engineers and Manufacturing Professionals*. Boca Raton, FL: St. Lucie Press.

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

Tull, Jethro. (1731). *The New Horse-Houghing Husbandry, or, an Essay on the Principles of Tillage and Vegetation*. Dublin: Cyber Library, College of Bioresource Science, Nihun University.

Yang, Kai, and Basem El-Haik. (2003). *Design for Six Sigma*. New York: McGraw-Hill.

Index

D

Q

R

T

U

V