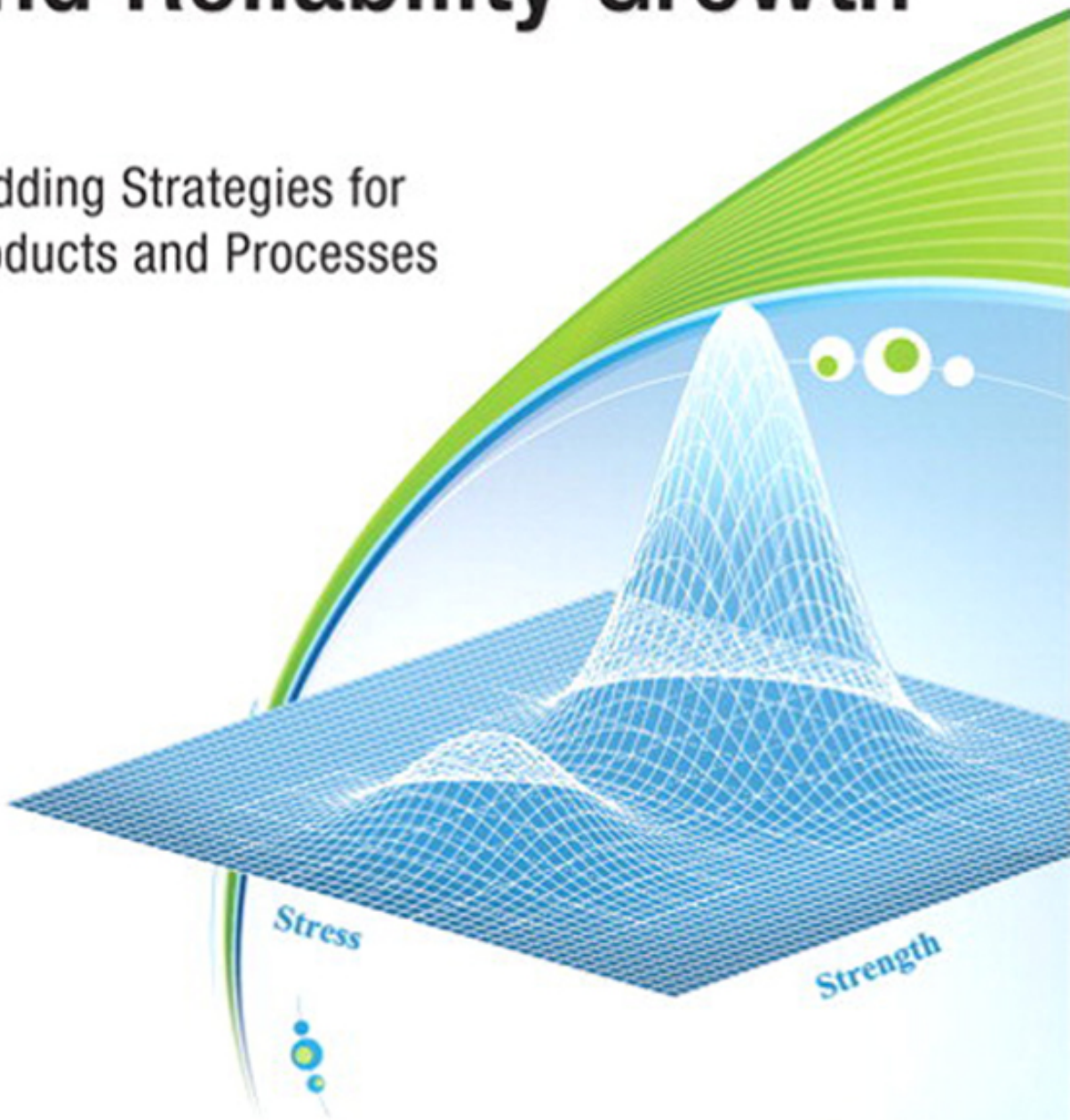
PRENTICE HALL

Robustness Development and Reliability Growth

Value-Adding Strategies for New Products and Processes



John P. King • William S. Jewett

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales
(800) 382-3419
corpsales@pearsontechgroup.com/ph

For sales outside the United States, please contact:

International Sales
international@pearson.com

Visit us on the Web: informit.com

Library of Congress Cataloging-in-Publication Data

King, John P., 1944-
Robustness development and reliability growth: value-adding strategies for new products and processes / John P. King and William. S. Jewett.
p. cm.
Includes bibliographical references and index.
ISBN-10: 0-13-222551-4 (hardcover : alk. paper)
ISBN-13: 978-0-13-222551-9
1. New products—Management. 2. New products—Technological innovations.
3. Reliability (Engineering) 4. Product management. I. Jewett, William S. II. Title.
HF5415.153.K535 2010
658.5'75—dc22
2010003456

ISBN-13: 978-0-13-222551-9
ISBN-10: 0-13-222551-4
Text printed in the United States on recycled paper at Edwards Brothers, Ann Arbor, Michigan.
First printing, March 2010

Preface

To do anything well, repeatedly over time, requires a disciplined process. This is especially true of product development. People come and go, so your organization needs a process that doesn't depend on having specific people who have internalized your process and know how to practice it well. Your good people and your standardized processes are part of your corporate DNA. Of course your people are the most important ingredient, but it's risky to depend entirely on a veteran staff. Great people following a flawed process won't produce the best results. With a clear call to action, processes can be improved and implemented in ways so that they become a source of competitive advantages.

Ultimately the product development process is about risk management. Failing to meet the expectations of customers for product features, functionality, reliability, usage life, and cost of ownership, for example, can have serious consequences for your business. Products that fall short in the marketplace can threaten both your customers and your enterprise in many ways, such as the following:

- Risks to the health and safety of end users can increase product liability costs.
- Your product development teams may continue to work on urgent problems with current products in production rather than developing your next-generation products.
- Your service organization may have continued demands to correct problems that have escaped the factory because of faulty designs or manufacturing processes.
- Management can be preoccupied with correcting past mistakes and not paying enough attention to future developments.
- In the short run, warranty expenses may increase. This can be the most benign result of quality and reliability problems in products' designs.
- In the long run, customers' dissatisfaction will result in lost revenues and market share.
- If the problems with product quality persist, over time it will become increasingly difficult for your company to attract and retain good people. Work will no longer be an enjoyable or rewarding experience. Eventually people will conclude that making avoidable mistakes over and over is evidence of an organization that does not learn how to improve or to compete to win in the marketplace.

Reliability as Quality over Time

Many books on reliability emphasize the mathematical treatment of the subject. They place considerable focus on topics such as probabilities, life distributions, and the analysis of failure data. These are all important subjects. We too have included them. Statistics provides the mathematical tools and rigor that support the analysis and modeling of system reliability. However, the scope of reliability is

broader than statistics. Many reliability problems involve deteriorations in performance over time, not just broken parts. As we will see, robustness is foundational to reliability, and its development should be the early focus of team members responsible for meeting reliability goals.

We have two working definitions of reliability: the traditional definition, focused on the probability of survival, and a more general definition, focused on the stability of performance. Although this second definition is not useful when trying to calculate the percentile for 10% failure, it's a powerful model that is more inclusive and more actionable.

David Garvin's "eight dimensions of quality"[1] include several that are driven by reliability:

1. Performance
2. Features
3. Reliability
4. Conformance
5. Durability
6. Serviceability
7. Aesthetics
8. Perceived quality

When we expand our definition of *reliability* to be "quality over time," reliability has great leverage on customers' satisfaction. Viewed this way, reliability becomes a "quality anchor." If performance drifts off target, customers' perception of quality will follow. When we use the term *reliability* in this book, we are referring to the broader definition of "quality over time."

Developing reliable products requires a good process that develops data that enable the improvement of product performance and its stability under conditions of stress. Many of the stresses affecting performance have a strong dependence on time and how the product is used. Making product performance less vulnerable to these stresses will improve the stability of performance under those conditions over time.

Six Sigma and Reliability

Six Sigma practices have been used for the improvement of business and technical processes for a couple of decades. Their roots go back much further. There are no "magic bullets" in Six Sigma, but the methodology does offer a disciplined process that can be helpful in solving problems. Most technical people accept the scientific method and do not need to be convinced of its value. The Six Sigma process wraps the tools of engineering statistics around the scientific method. We see a problem, form a hypothesis about its cause, and gather data. By analyzing the data we either accept or reject our hypothesis, or conclude that we do not know enough to move forward with an appropriate solution. When efforts are prioritized, the concept of critical-to-quality (CTQ) is central to everything that makes Six Sigma so useful. It offers a nice model for defining what is important to customers and then mapping that information in a hierarchy. The resulting CTQ tree diagrams the linkages among all of the parameters that the system designers have to specify as well as those product outputs that are important to customers. As we describe the process of critical parameter management (CPM), we use the CTQ tree as a way to characterize the flow down of system requirements and the

1. David A. Garvin, "Competing on the Eight Dimensions of Quality," *Harvard Business Review* (November–December 1987).

flow up of system performance. We build upon that and other tools as we outline the product development process and best practices for the development of robustness and the growth of reliability.

Typically, Six Sigma practitioners differentiate the tools and practices that are included in DMAIC (Define-Measure-Analyze-Improve-Control) processes from the methods of Design for Six Sigma (DFSS). In the DMAIC process, the focus is on solving existing problems with a product or process. It is a systematic approach to correcting problems. On the other hand, DFSS is a proactive, forward-looking process that is focused on the prevention of problems. Similarly, the applications can be both to new products and to technical and business processes. Because DMAIC and DFSS have different objectives, the tool sets tend to be different. Both processes have best practices that add value to the development of robust and reliable products. In the spirit of Six Sigma, we use what works. Here we speak in terms of the traditional tools wherever possible, trying to avoid the temptation to create new names and acronyms. We decode acronyms when we encounter them. You may already be familiar with many of them. We hope to introduce you to a few new ones.

Robustness Development and Reliability Growth

Our title for this book links two important engineering approaches: the development of robustness in a product's design and the growth of its reliability. Robustness development is focused on preventing problems, while reliability growth is focused on correcting problems. Reliability in the market, or as predicted during development, is a lagging indicator. Robustness metrics in development, however, are leading indicators. The development of robustness in subsystem designs enables the optimizing of robustness at the level of the integrated system. Robustness achieved early in development enables shorter cycle times in the later phases. Fixing problems late in a process may threaten delays in market entry and drive up manufacturing costs as tools are modified and inventories are scrapped. More time and resources spent in early phases[2] has paybacks in the later phases. An early focus on clarifying requirements, selecting superior design concepts, analyzing functional models, and planning designed experiments contributes greatly to "getting it right the first time," which you have heard often. Some problem solving late in the game is inevitable. Robustness development doesn't prevent all problems. We're human, prone to making mistakes. In the context of a project schedule, problems have to be identified and understood quickly and corrected gracefully, with prudent judgments of which problems are more important to correct before market entry and the beginning of production.

Managing the Product Development Process

The product development process can be described simply as a time-phased set of activities and decisions. To ensure a good outcome, certain tasks should precede others. The goal of the process can be simply stated: Get to market as efficiently as possible with a product that offers your customers value that is superior to that of your competition while managing risks to an acceptable level. To the extent that your projects do this better than your competitors', you can be more confident that your products and services will both satisfy and, in some ways, delight your customers. There will be positive benefits to your business. In the long run, superior products delivered on time with acceptable costs will drive superior returns to your business. What are some of the more important elements in your process?

2. Sometimes referred to as front-end loading (FEL).

Long-Term: Business and Research Strategies

- Make strategic decisions about your market participation.
- Align your portfolio strategy for new products with your market participation strategy.
- Develop a supporting technology development or acquisition strategy.
- Develop capabilities to deliver superior value to your customers.

These activities are aimed at positioning your company to participate in selected markets and to support this decision with the required development initiatives. Many considerations go into making these decisions. They can include the trends in your business, your core competencies, emerging technological advantages and disadvantages, and the difficulties in repositioning any of your products. Decisions such as these set the foundations under your product development projects.

Short- to Medium-Term: Product Development Process

- Understand the needs of your customers.
- Translate customers' needs into technical requirements.
- Develop better design concepts that satisfy your customers' needs.
- Choose the best concepts from available alternatives.
- Develop the selected concepts into production designs.
- Make the designs robust against sources of performance variation.
- Employ accelerated methods to identify and correct problems to grow reliability.
- Develop robustness in manufacturing, service, and other business processes.
- Prepare for the scale-up of manufacturing, service, and customer support.
- Launch the product into selected markets.
- Provide marketing, sales, and customer support.
- Provide maintenance and repair services.
- Discontinue production at the end of the product's life cycle.

Of course these accomplishments are neither simple nor serial. Development phases overlap. Some activities start as early as possible, while other activities have dependencies on the completion and freezing of earlier work. Still other activities may continue into later phases and impose risks due to their unfinished work. Preparations for manufacturing start long before manufacturing begins. Verification of the product's design is a process that begins while obtaining the validation of requirements from customers and their feedback from their testing of prototypes. Effective organizations seek feedback continually during their progress, from their initial fuzzy ideas through market entry into production. Overlaid on this set of activities are processes to manage the project and to ensure that the business manages its risks.

Key Features of This Book

- Strategies to improve the effectiveness and efficiency of product development, with particular attention to those influencing product reliability
- A comprehensive process for developing new products, with details relevant to reliability development
- Distinctions between the processes for robustness development and reliability growth

- Methods for the development of information critical to the achievement of higher reliability
- Detailed guidance for the planning of experiments and the analysis of data from laboratory tests and field failures
- Methods to optimize performance using both first-principles[3] and empirical system models

In the spirit of *Good to Great*,[4] improvements aggregate over time. Usually we don't make major improvements in anything overnight. Doing things better usually requires working and thinking differently, expecting you to modify your behaviors and eliminate your bad habits. Discipline, persistence, and corporate motivation are essential ingredients to enable positive changes and their implementation. The better process and tools will not help if management lacks the resolve and the desire for better business outcomes. Your "call to action" should be derived from your recent experiences with business results. Ultimately your people, processes, and capabilities make the difference and determine the fortunes of your company.

Topics of Interest versus Functional Responsibility

Of course we believe that our book has value to all who work for the development of new products. We encourage you to read it cover to cover. The book is organized into sections.

The first section, Critical Drivers of Value, provides broad insights about development strategies and tactics. It explores concerns for the consequences of not delivering superior value to your customers.

The second section, Framework for Reliability Development, describes approaches for the development of robustness and the growth of reliability in the context of a product development process with well-functioning governance and decision processes.

The third section, Tools and Methods Supporting Reliability Development, describes selected methodologies with reasonable depth that may enable managers to institutionalize these practices, if they are absent, and to have value-adding conversations with tool specialists. Our intention is that managers set clear expectations for the consistent application of better practices that deliver data with higher integrity to decision processes. For those readers who are not familiar with these tools, we encourage reading further and a pursuit of training to build new capabilities.

In the last section, Integration of Framework and Methods, we discuss the management of critical parameters. An example of a development project ties the strategies, decisions, and methods together in a useful illustration.

With all the demands on your time, we recognize that, depending on your responsibilities, some topics will be of more immediate interest and utility than others. We offer the following suggestions about what to read first.

General Managers: Chapters 1–6, 25

These early chapters should be helpful for understanding the model for product development and how reliability is directly affected by the quality of its implementation. We have given significant

3. A first-principles model is one that uses the established laws of science to predict the behavior of a physical system, rather than an empirical model derived from experimental data.

4. Jim Collins, *Good to Great* (New York: HarperCollins, 2001).

emphasis to risk management and decision making in the face of uncertainty. Ultimately, decisions to proceed, as a project moves from phase to phase, are business decisions requiring risks to be identified and evaluated. In the course of running an enterprise, risks are assumed on a regular basis. The likelihood and potential impact of things going wrong in the future have to be understood, and appropriate actions selected and implemented.

Project Leaders: Chapters 1–8, 24, 25

Project leaders are responsible for delivering a quality product on schedule with the right cost structure. There are strong interactions among quality, costs, and schedule. Decisions must be made to select the most appropriate implementation strategies and solution concepts for the product's development. Project management plans must then provide the right resources at the right time to implement the strategies and develop the solutions. Usually projects that are in trouble with their product's performance and reliability are also in trouble with their schedule and resources. The decision that budgets and schedules are all-important should not be made without understanding the associated risks and consequences to product quality. We offer some methods to understand risks.

Product Development Engineers and Scientists: Chapters 2–25

Discovery and understanding are essential to building successful products. As technical people we have a need first to understand how things work and then to make them work better. Usually pride of ownership and an understanding of the impact of shortfalls are drivers in this process. To do this we build both virtual and physical models through thoughtful experiments and first-principles analyses. The models can then be used to improve performance and reduce its variability. In later chapters we offer a collection of tools and practices that, when combined with a well-thought-out process, will help improve your chances of success.

Reliability Engineers: Chapters 2–8, 14–25

Traditional reliability engineering has focused on identifying and correcting failed components. Failed parts that cause the shutdown of a product can be dramatic, and shifts and deteriorations in performance are also a major source of customer dissatisfaction. For this reason we recommend a more holistic approach that broadens the focus to include performance variability and drift. It pulls more involvement by reliability engineers who should be a part of any product development project. Reliability engineers need to be there at the beginning, when specifications are derived from customer requirements, and to be part of the product development team throughout the project.

Quality Professionals: Chapters 2–25

Quality must be designed into a product as well as into its manufacturing and service processes, not achieved by inspection and repair. Quality engineering is a function that can help the organization be proactive in achieving higher levels of quality and reliability. The involvement of quality engineering at one time did not start until a design had been completed. It focused mainly on supply chain activities. This limitation has been recognized by many organizations and corrected. Today, quality engineering has become an important partner in the product development process.

The input of the quality organization is valuable to the planning and execution of reliability design and its verification. It should start early in the process.

Change Begins with You

We intend that our book will provide you with valuable insights. We expect that some will be new and enlightening. It's been an enjoyable project for us to describe the many paradigms and principles that have worked well in our own experience. Time has proven that these strategies, tools, and practices are not just a fad or a theme of the year. They have endured well and work nicely with current initiatives to make product development more lean, flexible, and agile. Certainly their implementation evolves with use. It takes time for new methods to become integrated into a larger set of practices as organizations learn better approaches that can provide major benefits with immediate and lasting value for their companies. It does no good for us to keep them to ourselves. Now it is your job to read, think, adapt, and practice so that you can achieve sustainable competitive advantages. We encourage you and your teams to be students of better processes and methods. Experiment and allow the demonstrated benefits to provide the encouragement that often is needed for institutional changes.

If you are an individual contributor, remember that it is possible to make tangible improvements in the quality of your work without having a formal initiative prescribed by management. If everyone on a team embraced that idea, the organization would reap many benefits in improved product quality and organizational efficiency.

If you are a manager, you can accomplish a lot by setting higher expectations for better information to enable intelligent risk management. By better information we mean developing data that are useful for informed decision making. This means understanding what's important, and developing the relevant performance metrics that must be tracked. Knowing where you are relative to your goals and understanding the problems that must be solved are key to assessing risks and making effective decisions.

The marketplace rewards those who are excellent at execution. Strive to be better than your competitors and serve your customers well. Good luck in your journey!

CHAPTER 1

Time, Money, and Risks

The capabilities, performance, and reliability that are achieved in a product prior to market entry are functions of many activities and decisions during the development process. Physical laws drive product failures. They are also symptoms of implementation problems with plans, resources, methods, and tools. This chapter discusses typical drivers of reliability that are inherent in the way product development is practiced. Later chapters address technical concerns.

Inherent in discussions about product development are recommendations for process improvements. Assessment models such ISO and CMMI are effective at driving organizations to define their processes and to follow them well. Benchmarking can help even more by enabling the people in your organization to learn firsthand about better methods that they would not have known from internal experiences. You compete with your processes as well as your products and services.

During our years of working in product development we have experienced many projects that had elements that were executed very well. Excellent design concepts were developed. Manufacturing processes were controlled to be stable and aligned with the specifications of design parameters. Cross-functional teams managed parallel work efficiently across all organizations and partner companies. Customers and suppliers were involved early and throughout the process. Timely decisions enabled the critical path, addressed project risks, and managed cross-project conflicts.

However, few projects were managed to execute all of the necessary elements well at the same time. More often than not, the consequences of good work in one domain were compromised by handicaps from other domains. For example, sound designs of robustness experiments were cut short by schedule reductions. The involvement of engineers with customers was canceled on the assumption that they already knew what customers needed. The transfer of new technology into product development was premature and incomplete, imposing risks on product development. Few projects learned how to repeat the successes of previous projects. As new products approached their market entry, crisis management became the norm. Schedules slipped, costs rose, designs were changed, tools were modified, parts inventories were scrapped, resources were kept on the job. Heroes may have saved the product launch date, but often the

product itself limped into the market with less-than-acceptable quality, reliability, and costs. Design changes in production were common. These stories are not good!

We expect that your own experiences confirm this concern about the difficulties of executing product development well and consistently. We've known project teams that proclaimed themselves to be best in class without merit. From the viewpoint of both the business and the technologies, their projects were disasters. Often there was great technical work, but with poor business results. Certainly you have experienced examples when everything important worked well once in a while, but not consistently.

A major lesson is that failures become the call to action for major improvements in the way that product development is practiced. One result is a commitment to a development project that has superior strategies, that stays on its plan, and that adapts easily to inevitable changes. It delivers designs that achieve their requirements and are robust. It delivers the new product system to its market gracefully. The product is viewed to be superior to its competition. What a great experience! The challenge is to repeat those successes in follow-on projects. That's when standardized processes employing better practices and well-managed teamwork need to be institutionalized so that they have lasting value for your corporation.

Quality, Costs, and Schedules

"Bottom-line" business metrics usually relate to accounting parameters such as the costs of poor quality, the costs of development or manufacturing, and the costs of schedule delays. "Top-line" growth metrics usually focus on revenues generated from new products, with insights from customers' satisfaction indices and competitive positions that contribute to pricing. Their forecasts are predictors of revenues due to sales and usage volumes. Broadly, they apply to the value generated by the portfolio of new products evaluated over time.

Higher reliability developed in a product may be perceived as a competitive differentiator. It may also be a significant contributor to reduced service and warranty costs. Higher levels of robustness contribute not only to higher reliability but also to reduced manufacturing costs. To the extent that they are achieved earlier in a development project, they contribute to shorter, more predictable development schedules and reduced development costs.

Many contributors to reliability are within the control of the project teams prior to market entry. During this time, the reliability metrics are forecasts of the level and stability of future performance in the hands of customers. Of course, the ultimate measures of success are derived from the actual performance of the new products in the market, the degree to which higher reliability delivers value to customers, and the new product's contributions to your overall business. For customers of non-repairable products, reliability is perceived as the product's usage life. For repairable systems, it is evaluated by metrics such as failure rate, service frequency, availability when needed, percentage of downtime, or other measures relevant to your customers' business or activities.

With product reliability having the potential for being a significant contributor to business success, your product development teams have to align their technical achievements with those elements of value to be returned to the corporation. To the extent that teams are excellent in the execution of robustness development and reliability growth, they may achieve competitive advantages relevant to their rivals in the market. The more that product development teams work in a productive environment, using efficient methods and benefiting from constructive teamwork, the higher will be their probability of success.

Product Development

Product development is not just a process for engineering teams. "Concurrent engineering" has focused attention on cross-functional activities that develop the product and its manufacturing processes. In addition to engineering leadership teams, effective work groups include representatives of other involved functions to manage the development of their capabilities to launch the product on time, to sell the product, and to provide service and support to their customers. So the management of product development activities must integrate these organizations along with their customers, suppliers, and partner companies.

Product development is a very competitive business. It can be difficult and highly complex. Unfortunately, well-intended processes can be constrained, if not sabotaged, by shortened schedules, late project starts, dysfunctional decisions, inadequate reserves of resources, midcourse changes in portfolio plans, and unresolved conflicts with competing projects. The market rewards good performance but penalizes poor performance, without mercy. If you win, you get to keep your job. If you lose, bad things can happen.

Who differentiates the winners from the losers? Customers do! They vote with their purchases of products and services over time, showing preferences for those companies whose offerings provide superior value in the context of their business, activities, and environments. Customers have options. They choose those products that best meet their needs. Generally, that means those products that provide necessary benefits, in relationship to their costs, better than available alternatives. Everything else follows, such as price, revenues, volumes, competitive position, reputation, and shareholder value.

Companies develop products and services for those business opportunities for which they have competitive advantages. The job of product development is to provide superior solutions to those problems for which customers in target markets are willing to spend their money. Perfection can be elusive, expensive, and time-consuming, but where compromises have to be made, "good enough" must be judged from the viewpoint of customers. Trade-offs represent potential risks for product development, so they need to be made with accurate knowledge of the consequences for customers' value drivers.

The choices that customers have include doing nothing. So there is little forgiveness for not achieving excellence in execution for every functional discipline involved in the project. Customers may be loyal to your company based on a historical relationship. They may cut you some slack, knowing that the products or services are new and understanding that you need to work out some bugs. They may accept your less-than-superior product, knowing that your superior service will compensate, or vice versa. But how long will that last?

The "early bird gets the worm" may be a useful metaphor. Often being "first to market" has its value, since early adopters can set precedents. Products or contracts may commit customers to a long life cycle. Often, however, it's being "right to market" that wins in the end, that is, having the right quality (features, functionality, reliability, usage life) with the right costs at the right time. So if your product is first to market but at a high price or with compromised quality, can your company win in the long run? An old but telling attitude toward product development is "Quality, cost, and delivery schedule—pick two." That implies defeat, a compromise that serves neither your customers nor your company. The business challenge is then to work with a strategy that satisfies all three criteria.

This book does not propose to compromise quality by some clever approach in order to satisfy imposed constraints on development costs or schedule. Customers are not willing to make that trade-off. Likewise, we do not propose that the best product development methods take too

long or cost too much for you to adopt them. Companies that are good at product development are also efficient and flexible. They treat product development as an investment in the design of a new or extended business to generate returns that are expected to be superior to alternative investments for the corporation. Otherwise, why would they do it?

There are no magic tricks here. If you lag behind your competitors in a substantial way, we doubt that there are quick fixes that will be adequate. It takes hard work sustained over time with effective implementation of better practices. It takes the right resources at the right time working on technical concepts that are superior to their competition. It takes management working to enable development projects to be successful, having a constancy of purpose over time.

The process does not focus on a specific improvement theme, but rather on the integration of many practices that "winners" do to be successful. Over the past few decades, leading companies have implemented approaches to make their work more efficient, to enable their processes to achieve shorter cycle time, and to develop their technologies to be more robust. Most of these improvements have provided benefits, but none is sufficient alone. They must be integrated into the natural way that people and teams do their work.

The Quality Improvement Movement

Quality training has provided many helpful strategies and methods, from quality circles to an emphasis on analytical problem solving. Many of the approaches use tools that were learned in the past but have been forgotten or were not well applied. Improvement initiatives have encouraged engineers to get closer to their customers and to develop technologies, designs, and processes that are more robust. Project management professionals have pushed sound methods for managing resources and their progress in performing activities and achieving deliverables on predictable schedules. Improved computer-based tools have been implemented to assist analysis, experimentation, and design documentation. Software development has adopted agile methods to embrace late changes that often are imposed by rapidly changing markets. Organizations for software engineering and systems engineering have developed "capability maturity models." Organizational development initiatives have improved the leadership and practices of cross-functional teamwork and of their support from functional management.

Industry studies, conferences, and publications continue to push for improvements. Their agendas have themes such as Design for Six Sigma, lean product development, agile development, risk management, high-performance teamwork, collaborative innovation, and portfolio balance. All of these are important and complementary. The trick is to integrate them in a manner that makes them appropriate for your business model, with implementation that makes them systemic, to become just the way you do business, not a collection of "themes of the month."

It takes "top-down" implementation and "bottom-up" practice. It takes impatience with poor performance and a sense of urgency to improve, with an imperative to achieve the benefits quickly and to remain consistently good over time. It takes creativity to develop new techniques and a flexibility to adapt to changing conditions. It's an ongoing process of improvement.

Companies that "beat" the competition in the market tend to have a knack for integrating resources, processes, and technologies to achieve results that customers judge to be superior. They do it consistently, product after product, to reinforce a reputation in the market for delivering superior value. Often it's that reputation that provides a shortcut to the value judgment, rather than a side-by-side comparison of features or a competitive analysis from an independent assessment. How can the implementation of fundamental improvements be managed?

Management of Process Improvements

The process of change for an organization can be difficult, with nontrivial risks for achieving the intended benefits. Probably you have experienced this yourself. There can be many sources of resistance. The investments that are required may be beyond those that are thought to be affordable or worthwhile. The challenges to learn new methods may be viewed as not contributing to short-term imperatives. People in power may not believe that the problems being experienced are severe enough to justify the changes. Paradigm changes may not be understood easily. We expect that you can add to this identification of barriers easily.

Certain factors have been found to be critical to success in changing the approach to product development. Examples include the following:

- The reasons for the change must be clear and compelling, both for major transformations and for reforms in the context of continuous improvement.
- The adoption of new methods must be developed internally so that ownership is with the people who must practice them.
- The implementation process must be embraced and enabled from the top down. The improvement plans must have objectives that harness the corporate energies.
- The change process must achieve measurable results early to demonstrate the benefits and to build confidence and momentum within the organization.
- The timing of implementation must not be so disruptive that the organization resists it as a "theme of the month." Remember this advice: "The best time to fix the roof is when the sun is shining."

There are several good sources for useful insights and practices for change management. You may find the work of John Kotter[1] and others to be interesting and applicable.

Clear and Compelling Call to Action

With probing, open-ended questions your evaluations of recent development projects can identify those practices that provide competitive advantages and those that consistently put your projects at a disadvantage. Where should you focus? Clearly your business achievements and disappointments will set your priorities. Capability assessment tools can identify the gaps relative to "best in class." Cause-and-effect diagramming can establish probing questions to identify those factors that contribute a larger impact and have a higher probability of occurrence.

For example, suppose your projects miss schedule milestones consistently, with market entries being late. What are root causes that can be addressed? Certainly your projects will have their own specific factors that contribute to these difficulties. If it seems evident that project management practices are deficient, the Project Management Body of Knowledge (PMBOK)[2] can guide probing questions about practices and their integration. If it seems that engineering practices are sound, but cross-functional interactions are too iterative and contentious, the principles of effective product development and of high-performance teamwork[3] can provide probing questions. Clearly there can be complex interactions among the many factors.

1. John P. Kotter, *Leading Change* (Boston: Harvard Business School Press, 1996).
2. Project Management Institute, *A Guide to the Project Management Body of Knowledge* (Newtown Square, PA: Project Management Institute, 2000).

What does not work well? What are their causes and effects? Here are some thought provokers that are relevant to our topic of reliability development:

- **Weak design concepts:** Reliability problems caused by immature or inferior design concepts that cannot be controlled well enough can force too many design changes late in the development process. Those that have consequences for tooling and parts inventories, for example, can lead to schedule slips and cost overruns.
- **Delays in market entry:** Major schedule slips increase development costs, threaten revenues, and have serious consequences for those organizations mobilized to launch the product. Long, repeated delays can jeopardize the competitive product positioning and threaten the reputation of the company.
- **Escaping reliability problems:** In addition to dissatisfying customers, design and manufacturing problems that are not prevented prior to production impose higher service costs and the potential need to retrofit design modifications or to recall products.
- **Constrained human resources:** Not having enough of the right skills or experience working on specific problems can cause inferior or delayed solutions. This problem raises the question of whether to hire, to develop internally, or to outsource the missing skills.
- **Constrained prototypes:** Additional prototypes may be necessary to develop robustness and grow reliability, or to obtain customer feedback. That may require additional funding and development time, which management may not tolerate.
- **Overloaded resources:** Centralized functions may be staffed only to maintain a high level of utilization, leaving no reserve capacity for variations in demand. The lack of capacity when needed may be a root cause of conflicts among projects and queues that steal development time. For example, the lack of access to an overloaded testing facility may cause important tests to be skipped in deference to the schedule or delayed to the extent of jeopardizing the product launch date. This problem can be anticipated if there is an integrated approach to project planning when using shared resources rather than a culture of competition, where the project manager having the dominant personality gets satisfied first and everyone else fights for the leftovers. Typically, overloaded resources and top-down pressures lead to multitasking, which actually tends to reduce efficiency and further aggravate the resource problem. For a more detailed discussion, in the context of Critical Chain Project Management, see Leach.[4]
- **Lack of attention to risk management:** Project teams may not be diligent at identifying risks or reducing them. Certain risks may become actual problems that must be resolved, often an expensive process that can delay the product launch. The management of risks takes resources, time, and management attention, all of which may be in short supply.
- **Bureaucratic decision processes:** Management gate reviews and other decision processes may be difficult to schedule or may demand iterations of reviews to answer unexpected questions. Those iterative reviews may actually dilute responsibilities rather than reduce risks. They may tend to add little value for customers and cause significant delays along the project's critical path. They often fail to draw sufficient attention to the rate of reliability growth that lags behind expectations and needs management intervention.

3. Jon R. Katzenbach and Douglas K. Smith, *The Wisdom of Teams: Creating the High-Performance Organization* (New York: HarperCollins, 2003).

4. Lawrence P. Leach, *Critical Chain Project Management* (Norwood, MA: Artech House, 2005).

- **Lack of flexibility in a design:** The system architecture of a product may depend upon the robustness of a specific design concept. If that is not achieved early in the project, the inability to change to another, more robust design concept may cause serious consequences for schedules and costs. The capabilities of the entire product and its development schedule can be in jeopardy.
- **Late changes in requirements:** The development of complex, heavy metal products can benefit from requirements being defined and frozen early. However, many products are aimed at markets that can change rapidly, leading to either late-arriving feedback from customers or new expectations set by competitive entries. Design architectures or development processes that cannot accommodate late changes gracefully can be plagued by the consequences to their schedules and costs.
- **Weak investment in early development:** Design concepts that are chosen to be baseline may be found to lack sufficient robustness in the early development phases. This can lead to numerous problems that can force costly corrective actions in the later development phases.

Of course, the list of possible problem-cause scenarios is endless. The challenges for your capability assessment are to identify the business results that are not acceptable and to establish their priorities and their root causes. Process solutions can then be devised, often with assistance from outside specialists who can bring wisdom from their experiences with other projects or companies. The business case for improvement initiatives can then argue for resources, justified by a "call to action" that is derived from the consequences experienced.

Internal Ownership

When new "best practices" and methodologies are introduced, they rarely become "common practices" until lead users inside your projects demonstrate their benefits. It takes learning and experience for people to become believers. With internal experts coaching project work, the consequences of the learning curve for project cycle times can be reduced. Once the benefits are seen to be important to a project's objectives, the new practices can be built into the project's plans. When management understands the intent of the new practices and their technical or business advantages, they can set expectations for their appropriate use. This can be very powerful. Without reinforcement by project leaders and management, new practices with high potential value can be set aside in favor of the more familiar "way we always have done work."

Top-Down Enablement

People in power positions may feel threatened by suggestions that current strategies and capabilities do not work well enough. They may see a recommendation as a reflection on their own personal competence rather than a concern for a corporate process or method. They may not appreciate the long-term value of time off the job for training or of the investment in new software tools. For example, getting customers on your project team may be viewed as too troublesome or a violation of project security. However, pilot projects can show how substantial the benefits can be. The emphasis on the development of superior robustness may take a clear understanding of why a design-build-test-analyze-redesign strategy under nominal conditions is not sufficient. Learning from other companies can help a lot.

It should be in the interest of management to promote changes that will add value. Without their participation, bottom-up initiatives tend to be short-lived, doomed to falling short of their

potential impact. It helps greatly to have a high-level champion whose commitment is based on painful experiences. Members of the management team need to be the suppliers of resources, the stakeholders in the expected benefits, and the drivers in the implementation of changes. An effective tactic is for management to teach and coach the new principles and methods. This has a powerful influence on how well new methodologies are understood and integrated. It also ensures that individuals and teams recognize them as being expected in how they do their work.

There is major value in the expectations and rationale being set at the top of the organization. Subsequently, these expectations must cascade throughout the organization and be reflected in individuals' performance. Clear expectations drive behavior changes if they are reinforced in top-down communications and applied in the probing questions and acceptance criteria for project reviews. It is management that enables major changes through strategies, resources, motivation, and a constancy of purpose. It's also management that can sabotage major changes by not making their support deliberate, visible, and consistent.

Bottom-Up Improvement

What if there is no upper-management support for a major overhaul of the product development process? Should you assume that there is no point in trying to improve? Fortunately, even with management reluctance, it is still possible to make improvements. Usually management recognizes when things are not optimal. They would like capabilities to improve, so it is very unlikely that they will prohibit the use of better methods and processes as long as improvements don't jeopardize schedules, budgets, or product quality. If all you can get is indifference, take it. It is better than active interference or prohibition. The problem is that everyone is tired of the "program of the month." A major process overhaul requires the commitment of time and money, making it very difficult to argue that there will be an acceptable payoff. This is especially true if people have been disappointed with past initiatives. Under these conditions, a better way to approach the problem is to scale back from trying to make the "big win" and concentrate on demonstrating success on small subprojects. Shoot for small initiatives that will be "under the radar." It is easier to get buy-in after you have demonstrated improvements. When success happens, publicize it. Before long there will be supporters.

Objectives with Substantial Impacts

What are the objectives of your major improvement initiatives? Examples may include

- Higher levels of customers' satisfaction
- Higher quality in new products and services
- Higher product reliability and longer usage life
- Lower manufacturing and service costs
- Lower development costs
- Shorter, more predictable time to market

These objectives may be obvious. What else applies to your situation? All can have a direct impact on the profitability of the portfolio of new products. They are measurable, but what are their criteria for acceptance? How well are these objectives related to the success of your business? How well can they be reflected in the performance expectations for individuals, teams, and projects? Development time can be measured for a project, but is the real objective a shorter

time or the more efficient use of time? Is it increased "leanness" or higher productivity? Possibly a better objective is to start on time.

Is a lower failure rate the objective, or should the emphasis be on reducing those failures with the higher severity levels? Should the project teams care more about constraining development costs to comply with budgets or about getting the right functionality, quality, and reliability to the market on time? Are all development projects comparable, or is each one unique in its challenges? The setting of objectives is not easy, particularly since if it is not done well it can create unintended incentives for the wrong behaviors.

Are these objectives independent? Probably not! Higher product reliability and durability will contribute to higher customer satisfaction. So will features and functionality that are not provided by competitive products. More customer benefits can enable higher prices without jeopardizing the perceived value. Lower manufacturing and service costs can enable lower prices. Lower prices can contribute to higher perceived value and higher manufacturing volumes, which in turn can decrease unit manufacturing costs and increase profit margins. A more efficient product development process should reduce development costs and enable on-time market entry with a more predictable beginning of the revenue stream.

Early Measurable Results

It is through performance expectations, cascaded throughout the organization, that the necessary actions and behaviors can be driven. This is particularly effective if people's paychecks are affected. For example, if engineers are required to shorten product development time, but managers are not expected to start development projects early enough, it will be clear that the engineers are to be punished for a failure in management. There is a powerful message sent when development teams share performance expectations with upper management. Balanced scorecard methods[5] provide a way to make the change process important to everyone's wallet. People then work to the same set of goals. Here are a few suggestions that may be helpful:

- Know what excellence looks like. It will give you benchmarks against which to establish your capability gaps and important projects for corrective actions.
- Evaluate your current experiences with an eye to the business consequences, not just to excellence in the scientific or engineering results.
- Identify lessons learned from past projects and from benchmarking. Incorporate them into the functional and project plans and the standardized business processes.
- Allocate funding, labor, and management bandwidth to the development and implementation of process improvements. Otherwise, not much good will be achieved and the past will also be the future.
- Do not try to fix everything at once. Focus on the "high bang for your buck" opportunities. Prioritize your improvement activities to achieve quick, visible results that build enthusiasm and confidence, and then form the basic foundations for longer-term improvements.
- Keep it simple. That doesn't mean that learning sophisticated methods is to be avoided. However, if a collection of improved engineering strategies and tools is the answer,

5. Robert S. Kaplan and David P. Norton, *The Balanced Scorecard: Translating Strategy into Action* (Boston: Harvard Business School Press, 1996).

their direct relevance to "bottom-line" benefits needs to be demonstrated for and understood by those who provide the funding and resources.

Soft Causes of Poor Reliability and Durability

What is reliability? What is durability? Here are useful definitions:

> ***Reliability:*** *Acceptable quality delivered consistently over time, under specified operating conditions*
> ***Durability:*** *Acceptable product life, under specified operating conditions*

So we focus on achieving the required quality, as perceived by customers, and on keeping it stable under the operating conditions of the various product applications.

These definitions don't have the rigor associated with being testable and measurable, but they are useful when thinking about reliability and durability at a high level. In Chapter 2 we present a definition for reliability that has statistical rigor.

Fundamentally there are two categories of root causes of reliability problems: "hard" causes and "soft" causes. Hard causes are in the technical elements of the product's design and of its manufacturing and service processes. Soft causes are in the human behaviors of project teams and in the management of the development process.

Many of the tools and processes in this book address the hard causes. Soft causes focus on opportunities for project leaders and management to enable excellence in execution and to avoid being the root causes of failures in project support, cross-functional integration, and teamwork.

Your process capabilities assessment may identify soft causes such as

- Project plans that are not based on sound strategies or cross-functional dependencies
- Production-intent designs specified before robustness and integration have been developed
- Product requirements that are neither complete nor stable (e.g., "feature creep")
- "Seat of the pants" decisions that increase project risks
- Selection of design concepts that lack sufficient capabilities or initial robustness
- Technical deliverables that lack clear expectations at a project milestone
- Development teams that lack process discipline or are complacent about their capabilities
- Project teams that are not able to speak truth to power or ask for help
- Project plans that do not apply lessons learned from past projects
- Development budgets that are reduced by ad hoc decisions external to the project

To the extent that your capability assessment recognizes these or similar problems, effective solutions can be developed. The literature on product development includes references that can provide additional insight into soft causes. For example, Robert Cooper[6] has characterized many things that successful companies do well. An interpretation of these, in the context of your

6. Robert G. Cooper, *Winning at New Products: Accelerating the Process from Idea to Launch, 3rd ed.* (Cambridge, MA: Perseus, 2001).

business model, can help you design probing questions for your assessment. Don Clausing[7] describes many fundamental engineering methods and echoes the concern about soft causes in his description of "The 10 Cash Drains." You may recognize many of them as root causes in your own business. Let's look at our list of soft causes in a little detail.

Inadequate Project Management Plans

A development process to achieve higher reliability should follow strategies chosen by the project to have the highest probability of success. It may leverage mature designs. It may be dependent on the development of specific technical concepts achieving prerequisite robustness. It may collaborate with other companies that have necessary and complementary technical capabilities. It may depend on extensive modeling or experimentation in the early development phases. In whatever way the strategies are intended, the project management plans must then reflect their relevant activities, dependencies, timelines, milestones, and resources. The plans have to show *how* the strategies will be implemented. If the resources or funding is not available, the plans may not be achievable and the strategies may not work.

Another concern is that the project management plans must deliver key parameters to the project's business plan. For example, if the plan to achieve the required reliability is not acceptable or achievable, the initiative to achieve the reliability requirement is set up to fail. So if the project plans are not achievable, the business plan also is not achievable and the project should not be chartered.

Inadequate Development of System Integration and Robustness

This is a critical concern for reliability development, being part of the foundation for the familiar guidance to "achieve quality early." To the extent that these criteria are not satisfied, additional risks are incurred and the objectives for quality, reliability, and durability are in jeopardy, as are those for schedules and cost structures.

The evidence of robustness is the specification of critical design parameters that have been demonstrated to control the mean and standard deviation of performance under expected stressful conditions. The specification and configuration control of these set points are the objectives of Critical Parameter Management (CPM). During the product design phase these functional parameters are converted into design parameters suitable for product manufacturing and service. Without them, there is no sound basis for the specifications of a design.

Important actions include these:

- Technical concepts should be developed to be robust for the application prior to becoming baseline for the product development project.
- Subsystems should be developed to be robust within the system architecture prior to being integrated with other subsystems.
- Robustness should be optimized at the system level in order to specify the parameters that must be controlled by the production processes.

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

The strategy for robustness development must incorporate an understanding of the roles and benefits of specific methodologies, many of which are discussed in later chapters. From a project management viewpoint, sufficient time needs to be allocated for the selection of the best design concepts and for the planning of experiments, avoiding a rush to cut metal and to build prototypes prematurely. Too many prototypes, built without the learning from previous generations, contribute to Don Clausing's concern for "hardware swamps." The rush to build-test-fix may satisfy management's desire to see things happen but may be contrary to sound engineering.

Effective product development is analogous to applied systems engineering. It follows a disciplined process of developing linked information and decomposing it to be useful at the various levels in the system hierarchy. Manufacturing and service functions are considered to be part of the system, as are the procedures for users of the product. The development process includes the flexibility to make changes easily within the system and to freeze elements upon which other developments are dependent. Although it is a disciplined process, it should be very adaptable to the technical and business characteristics of a project. The leveraging of existing information and designs is prudent and efficient, while ill-advised shortcuts introduce risks and can sabotage the project.

Changing Product Requirements

Your engineers create solutions to requirements that are understood in the technical language of their design concepts. The origins of these requirements fall into three fundamental categories:

1. The needs and preferences of your customers in selected market segments
2. The standards and mandates imposed internally by your company
3. The government regulations and industry standards for your markets and countries

Ideally these requirements are defined and clarified in the initial development phases and, in the ideal case, approved and frozen. In practice, life is not that easy. Requirements are vulnerable to misinterpretation, internal biases, lack of insight, disagreements, conflicts with each other, risks for achievability, and changing market situations. Feedback from customers may prove the initial requirements to be wrong or ambiguous. Your experience adds to this list. The consequences can range from being trivial to disastrous. If the requirements are vulnerable to changes, engineering is shooting at moving targets with an increased probability of major consequences for their delivered quality, costs, and schedules. If the requirements lack insight into the real intentions of their application, or lack clear differentiation from competitive products, the product and price positioning are in jeopardy. Failure to comply with regulations or industry standards can be a barrier to market entry. There is very little forgiveness.

Good practices are well developed to enable engineering and customers to work together so that designs not only solve the actual problems but also delight customers in ways that establish competitive advantages. Useful tools can enable the thorough and accurate translation of requirements from the language of customers into the technical language of engineering. They can then be decomposed to be applicable at the levels of subsystems, components, and manufacturing processes. Agile strategies for product development enable teams to react to late-arriving changes in requirements and to feedback from customers' tests of prototypes without dramatic consequences for schedules and costs.

It is the job of development teams to design the basic functions of the product, specifying the controllable parameters so that the product transforms the customers' inputs into desirable outputs, in spite of stressful factors that can cause deterioration or failure. Figure 1.1 illustrates

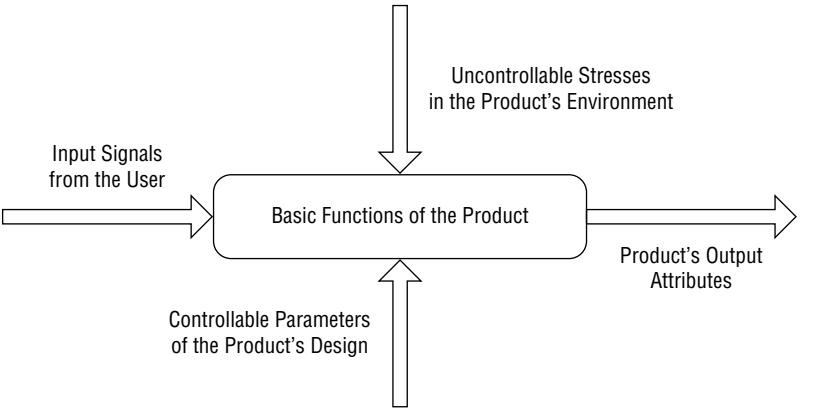


Figure 1.1 The requirements for a product's output attributes are driven by customers (VOC), as well as by internal mandates (VOB, VOT) and regulatory standards (VOM).

this simple viewpoint. It's then essential for development teams to understand the input/output relationships in the applications and the uncontrollable stresses that can affect them.

There are two fundamental concerns that project leaders need to address:

1. There is a tendency for engineers and management to believe that
   a. They already know more than their customers
   b. The process for understanding the voice of the customer (VOC) is too expensive and time-consuming, or it places the confidentiality of the project in jeopardy
2. Requirements can also be driven by "voices" other than those of customers (VOC):
   a. There is the "voice of the business" (VOB), which tends to demand lower development costs, shorter schedules, and lower production costs, but often without comparable concerns for satisfying customers. Lean product development and sound strategic decisions early in the process align with this mandate. Start the project earlier, instead of depending on schedule pressures to compensate for delayed decisions.
   b. There's the "voice of technology" (VOT), which tends to modify requirements to reflect those capabilities that can be delivered within the cost and schedule constraints. Another version of VOT is the "technology push" scenario that assumes, often naively, that customers will benefit from a new technology because it's new.
   c. There's the "voice of marketing" (VOM), which tends to push a wide range of requirements focused on opportunities for current sales and market share. These can be a reaction to the latest market information and to sales challenges with current products.

These additional sources of requirements are not to be ignored. The worst case of technology push assumes that the benefits of the product are needed, that customers will accept and pay for whatever capabilities you deliver, and that there are no competitors. This is often called a "WYSIWYG" product, that is, "What you see is what you get." Of course, in a competitive market that strategy doesn't work.

In many cases there are valid and reasonable strategies to shortcut the VOC process. It may be that a few key people have more insights about future markets than do current customers.

That's a gamble that may pay off. The challenge is to know when a rigorous VOC process must be included in the project plan and how to get more benefits from the investment in its process.

Risk Management

Many decisions are made during a product development project. Good ones clarify the direction of the project, enable the progressive freezing of information, and reduce risks. However, business and technical pressures can promote risk taking, even reward it. Risks are problems that have not yet occurred but might occur in the future. The probability of their occurrence is uncertain, as are their expected consequences for customers or for the company. When problems actually do occur, projects know that they have to be dealt with in a manner appropriate to their consequences. However, potential problems that have not yet occurred tend not to get the same deliberate attention.

Another way to say this is that problem correction is expected and rewarded, whereas problem prevention is a quiet competency that can go unnoticed. Problem correction late in the process tends to cost more than problem prevention. Excellence in achieving schedule milestones and the objectives for reliability and durability, for example, often is rooted in the capabilities of reducing risks, that is, avoiding the problems that consume valuable time and resources when they can be least afforded.

Good practices for managing risks reserve some capacity in resources to identify risks and to implement preventive actions, where justified. Contingency plans are the alternative. These are preplanned actions that will be taken to compensate for the consequences of the problem, when the problem does occur.

Collective wisdom and functional understanding are organized to identify risks and to characterize their expected consequences and probabilities of occurrence. This understanding can be based on experiences with previous projects, on specific technical or business insights, on scenario planning, or on other techniques. Often the foundations for a lower-risk project are put in place with decisions that are made in the early development phases, such as the selections of the development strategies, the system architecture, its design concepts, and their enabling technologies. The concerns increase when significant risks are ignored and risk taking is rewarded. But what are the costs? And who pays for them? For example, decisions based solely on reducing manufacturing costs can result in poor product quality or reliability, higher life-cycle costs, or a loss in market reputation.

Imagine a development project that achieved its requirements for quality, reliability, durability, production costs, and schedule, with no late crises that jeopardized its graceful market entry. Calculate the costs of delayed market entry, and then ask how much you could afford to invest in identifying risks and preventive actions. Remember the wisdom in "Pay me now, or pay me later."

Immature Baseline Technologies

Baseline design concepts are chosen in the early development phases. What criteria are used for their selection? How dependent on them is the product architecture? For new or adapted design concepts, what happens if their capabilities fall short of their expectations for the application?

The concern here is that new technical concepts may be chosen for reasons that are independent of the value they can deliver to customers. It may be that engineers favor a particular design concept for purely technical reasons. Possibly it enables lower costs or an attractive

architecture. It may be one for which they have personal preference. It might be the key to major increases in reliability and durability. Fine, but is it ready to be commercialized by a product development project? Does it have functional parameters that control its performance and variability? A disregard for this question tends to place technology development within a scheduled product development project, imposing additional risks. A very difficult situation is created when the product's architecture is entirely dependent on a design concept whose technologies are immature, that is, neither superior to alternatives nor robust for the application.

We remember a disastrous project. The product's architecture was entirely dependent on a design concept that enabled a rather compact system configuration with lower power consumption and reduced acoustical noise. However, the control parameters of the design could not be replicated by available manufacturing processes. In robustness terminology, the process could not be maintained "on target with minimum variability." In addition, the environmental stresses from the product's applications caused shifts in the functional response that customers would not tolerate. Fighting these problems continued well into production. No alternative designs were available, primarily because the system architecture could not be adapted. The root cause of the problem was a decision made very early that "bet the farm" on the hope that clever engineering could compensate for a failure of technology development. It could not. The costs were enormous!

In preparation for its commercialization, a new technical concept must be developed to be superior to available alternatives and to be robust for the range of intended applications. These criteria say a lot. Products often compete with their technologies. What makes them superior to available alternatives? What does it mean to be robust for the application? Good practices include two basic deliverables:

1. A stream of alternative technologies developed in anticipation of the needs of future product development projects
2. An unbiased selection process that places available alternative concepts in competition, with selection criteria that include demonstrated performance and robustness relevant to the intended product applications

Examples of "available alternatives" can include a current design already in production, an analogous mature approach that can be adapted to a new application, a design available from a partner or competitor, or an alternative new technical concept developed in parallel. A selection process that is vulnerable to the loud, bully voice of a concept advocate will lack the objectivity that is necessary.

Lack of Clear Expectations

Management has substantial power to guide a development project and affect its outcome. Certainly they charter projects, review their progress, provide resources, and make decisions. They also have the role of setting clear expectations. These expectations may focus on functional excellence. They may demand rigorous integrity in data that are needed to support decisions. They may express a bias for costs or schedule compliance. They may emphasize concerns for risks, that is, reductions in uncertainty. In the absence of clear expectations, development teams are left to define their own acceptance criteria.

Routine conversations with management, as well as those during project gate reviews, are opportunities for these expectations to be clarified and reinforced. If the discussions are entirely business-oriented, the technical expectations are at risk. If the discussions are entirely technical,

the development teams can tend to forget that they are managing an investment that is the generator of a new value stream with expected returns to the business.

The challenges for management are substantial. For the purpose of this discussion, they can include the need to

- Maintain an accessible and interactive relationship with development teams
- Ask probing questions with both technical and business insight
- Recognize good answers and understand their metrics
- Make data-driven decisions so that their customers win
- Set clear expectations for the effectiveness and efficiency of future project work
- Focus on the "critical few" risks that are most important to the business, rather than on the many potentially trivial "issues" that people tend to push on management
- Accept action items to enable the predictable progress by development teams
- Reinforce both flexibility and discipline in standardized development processes

Certainly there are other items for your list, but the important point is that your development teams listen to management. So management needs to be clear about the signals they send.

Lack of a Disciplined Process

Complacency can be a handicap for those companies that have dominant technologies, a high market share, admirable profitability, or a history of successful product launches. With deep pockets, money can be thrown at problems. An unfortunate side effect is that the realities of inefficient processes, growing competitive disadvantages, and customers choosing other companies' value propositions, among others, can remain outside a company's consciousness. Fundamental changes in the marketplace can be overlooked. Your company can become vulnerable to its own successes.

In these situations, the problems that have soft causes become much more fundamental than just product performance or reliability. Your basic business model may no longer be viable. Instead of your capability assessment providing guidance for continuous improvements, it may identify the need for major reengineering projects. Companies that survive over the long run often practice an ongoing reinvention of themselves. Those that do not often find themselves no longer being significant players in the market, or in some cases no longer existing. Although this concern is far beyond the scope of this book, to the extent that it exists it can be a major handicap to the development of reliable new products.

Inability to Speak Truth to Power

In the preceding discussion about the setting of expectations, the advice has been for management to have an "accessible and interactive" relationship with development teams. The absence of this relationship can easily jeopardize the ability of teams to deliver on their value proposition.

Suppose, for example, that management gave clear expectations that they did not want to hear bad news, particularly at gate reviews. This is a "shoot the messenger" scenario. How would development teams behave? Management may intend sternness and ridicule to motivate improved deliverables. On the other hand, such a policy may also prevent bona fide risks from being revealed or significant decisions from being based on the correct data. Project teams may not be willing to acknowledge problems, hoping to correct them before the next gate review. Worse yet, the development teams would be set up to be scapegoats for resulting failures. That situation is neither constructive nor efficient.

The quality leader Dr. W. Edwards Deming[8] was prescient when he urged management to "drive out fear." Rewards should go to those who are open and honest about the status of their project. Management needs to understand the truth early, when there's still time to react without jeopardizing the project's success.

This same intimidation may motivate project leaders to withhold any requests for additional resources or guidance. They may be convinced that it would reflect poorly on their own leadership capabilities. But who pays for this mistake? Imagine how constructive would be the process that brought the right resources to the right problem at the right time, regardless of a naive plan that was established sometime in the past. Independence and self-confidence are healthy attributes, but they can work against you. Certainly it's better to address problems when they are small, when their corrective actions can be managed easily. In his memoir, Harold Geneen,[9] the executive who built ITT into a multinational conglomerate, described his expectations on being open about problems. He felt that hiding problems was a waste of time, time being the one irreplaceable resource that his organization had. He expected his staff, when faced with difficult problems, to be open and to seek help early in the belief that the collective experience, wisdom, and skills of the organization usually trumped those of the individual.

When we work to solve problems, time is our most important asset. Letting too much time pass before getting help on a tough problem makes it more likely that schedules will suffer and that the delivered product performance and reliability will fall short of expectations.

Failure to Implement Lessons Learned

Many organizations engage in the struggle to get a new product to market faster but do not allocate time afterward to learn about what went well and what did not go well. These lessons learned are important contributors to continuous improvement. Without organizational learning, the mistakes of the past tend to be repeated. This can generate inadequate technical and business results across the portfolio. It can also demoralize the workforce who see the same mistakes being made over and over, and who will wonder if product development work will ever become less stressful.

A key element of organizational learning is the clear intention that the evaluation process not be a negative one, that is, one that is perceived to place blame on individuals or teams. To the extent that it is managed to be blameless and clearly intended to build wisdom into follow-on projects, with a top-down commitment to act on the results, it will be a healthy and anticipated process.

Inadequate Development Budgets

If teams are required to shorten development time and also to reduce their resource budgets, they may have objectives that are overly constrained. For example, it is often the case that an overrun in the project's development budget will have only a small impact on a new product's profitability. However, spending more money on resources to commercialize the product earlier, with higher quality at launch, can enable higher revenues and contribute to lower manufacturing and service costs. Unfortunately, budget compliance often gets more attention, probably because it is more easily measured and is a familiar measure of management performance.

This discussion of common soft causes of problems reinforces the notion that product development, although often perceived to be a set of technical activities, is managed in the

8. Mary Walton, *The Deming Management Method* (New York: Perigee Books, 1986).

9. Harold Geneen and Alan Moscow, *Managing* (New York: Doubleday, 1984).

context of a wide range of processes and behaviors that have the potential for either positive or negative consequences. That depends on how they are managed. The assessment of your company's capabilities and of the related needs for their improvement must include these soft concerns as well as those more technical hard causes.

Throughout this book we intend to shed light on both categories for improvements, particularly to the extent that they can contribute to the development of improved product reliability.

Key Points

1. The technical development of new products must enable the success of the business.
2. Developing new products is not just a challenge for engineering. Many functions of your enterprise must be involved deliberately in value-adding ways that are appropriate.
3. Ongoing research of markets and technical concepts must identify advantageous business opportunities and evaluate competitive threats.
4. Product development must implement superior design concepts that can be manufactured and serviced economically, with constant attention to risks.
5. Excellence in teamwork has to be achieved across organizations, between teams and management, extending along the entire value chain to suppliers and customers.
6. The processes and practices that are inherent in the way that work is done must deliver effective results in an efficient manner consistently from one project to the next.
7. With the investment in product development being an investment in developing a new or extended business, better methodologies, with excellence in their execution, can deliver substantial competitive advantages in the market and much-improved returns to the company.

Discussion Questions

1. How well do your product development teams understand your customers' value drivers?
2. How well do your new products, after market entry, return value to your company as expected by their business plan?
3. How have risks affected the predictability of your development projects?
4. What types of risks have materialized as problems? What were the consequences?
5. What successes have you had in developing and implementing process improvements?
6. How well have process improvements provided competitive advantages for your projects?
7. Give some examples that show how your development teams practice problem prevention.
8. What are the major difficulties faced by your development projects?
9. How well do your development projects learn from past successes and difficulties?
10. How do you think your product development process compares to those of world-class companies?

Index

Page numbers followed by f and t indicate figures and tables