Leadership, Teamwork, and Trust

Building a Competitive Software Capability

Watts S. Humphrey & James W. Over
Praise for Leadership, Teamwork, and Trust

“Watts Humphrey has always emphasized the importance of measurement in software development, and this theme has permeated his previous contributions in CMM, TSP, and PSP. Leadership, Teamwork, and Trust continues this mantra and compiles valuable lessons into principles and patterns that are consumable by executives and leaders. Measured improvement is the differentiator of successful projects and market-leading software organizations. If you want to learn to steer such endeavors, this book will provide some valuable insights.”

—Walker Royce
Vice President, Chief Software Economist
IBM

“How to successfully manage knowledge workers is definitely the first of the really big business management challenges of the twenty-first century. Now Watts Humphrey and James Over are able to show how improving leadership, teamwork, and trust are at the heart of what needs to be done and to explain exactly how empowerment, productivity, and profitability are deeply intertwined. This book provides expert guidance on how to reliably bring knowledge work in on time, on budget, and to the correct specification—something that the software engineering industry has been grappling with for decades. There is a better way, and this is it!”

—Mark Smith
Global Director of Quality (2000 to 2010) and former Senior Executive, Global PSQ, and Certifications Director, Accenture

“Read this book if you’re a team leader, manager, or executive responsible for knowledge-working teams. Benchmark your own principles and practices for team motivation, high product quality, and sustained competitive results against industry leaders. Based on their extensive software industry experience, Watts Humphrey and Jim Over present the techniques that empower self-directed knowledge-working teams to produce superior work, both predictably and at the lowest cost. Software organizations will be compelled to try the Team Software Process (TSP), as we did in Microsoft IT with great success.”

—Aiden Wayne
Information Solutions General Manager
Microsoft Entertainment and Devices Division

“I want you to know that TSP is one of the most valuable innovations implemented in the Beckman Coulter product development process since I joined the company in 2002. Software has become increasingly important to the success of our instrument systems. And in our business, quality is the most important factor for success. TSP gives us a path to better development time to market and superior quality. We are true believers.”

—Scott Garrett
Chairman and Chief Executive Officer
Beckman Coulter, Inc.
“Stock exchanges are businesses that have been shaken in recent years by new regulations and unprecedented competition driven by technology. The Mexican Stock Exchange is no exception and is currently immersed in its most important process of business and technological transformation since its creation in the nineteenth century. Understanding that the competitiveness of the exchange will come mostly from its technology platform, we have recognized the value of knowledge work and its management challenges. We adopted TSP/PSP, with coaching from the Software Engineering Institute of Carnegie Mellon, for managing the execution of our most critical software projects. Results so far are very good, and we plan to gradually extend the TSP/PSP practice across the company.”

—Enrique Ibarra
Director, General Adjunto de Tecnologías del Grupo Bolsa Mexicana de Valores (Mexican Stock Exchange)

“Watts Humphrey has done more to advance the science of Software Quality Management than anyone I know. His work has had an immense, positive impact on how I lead software organizations. If you want software that is better quality, faster to the market, and cheaper to build, then Watts Humphrey and Jim Over have a tremendous amount of wisdom to share. Great stuff.”

—Michael J. Cullen
Vice President, Quality
Oracle Communications Global Business Unit

“I’m very impressed with the results of TSP in my organization. It is possible to see the difference made by applying these new knowledge-management methods. With TSP, you can adjust your processes, make them leaner, and obtain high-performance teams. This book is perfect guidance for all executives and managers who want to introduce those methods into their organizations.”

—Joao Barracose
Senior Manager, Development Systems
BBVA BANCOMER (Mexico)

“PSP and TSP have proved to be incredibly successful means for my engineering teams and managers to make and meet their business commitments. Getting high-quality automotive infotainment and head-unit software developed by geographically and culturally separated teams on increasingly tight schedules demands the disciplined engineering and management techniques outlined and referenced in this great new book!”

—Peter Abowd
President, Worldwide Automotive Business
Altia, Inc.
Leadership, Teamwork, and Trust
The SEI Series in Software Engineering represents a collaborative undertaking of the Carnegie Mellon Software Engineering Institute (SEI) and Addison-Wesley to develop and publish books on software engineering and related topics. The common goal of the SEI and Addison-Wesley is to provide the most current information on these topics in a form that is easily usable by practitioners and students.

Books in the series describe frameworks, tools, methods, and technologies designed to help organizations, teams, and individuals improve their technical or management capabilities. Some books describe processes and practices for developing higher-quality software, acquiring programs for complex systems, or delivering services more effectively. Other books focus on software and system architecture and product-line development. Still others, from the SEI’s CERT Program, describe technologies and practices needed to manage software and network security risk. These and all books in the series address critical problems in software engineering for which practical solutions are available.

Visit informit.com/sei for a complete list of available products.
Leadership,
Teamwork,
and Trust

Building a Competitive Software Capability

Watts S. Humphrey
James W. Over

Addison-Wesley
Upper Saddle River, NJ • Boston• Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City
We dedicate this book to our families and to our TSP team at the SEI. Our families, particularly our wives, Barbara Humphrey and Patricia Over, have been wonderfully supportive throughout the many years it has taken us to develop these methods and to gain the experience and understanding required to write this book. Their support has been invaluable.

Our TSP team, many members of which have worked with us for nearly twenty years, has tirelessly and creatively participated in TSP development, taught many courses, and worked with countless organizations to demonstrate the method’s extraordinary effectiveness. We could not have accomplished what we have done without their support.

—Watts S. Humphrey and James W. Over
Preface xv
Acknowledgments xxi

Chapter 1 Creative Destruction 1
  Corporate Churn 1
  Knowledge Work 3
  The Urgency of Change 4
  The Softtek Story 8
  The Softtek Experience 9
  What Next? 11
  Summary and Conclusions 12
  References 13

Chapter 2 The Bureaucracy 15
  Why Organizations Need a Bureaucracy 15
  The Software Crisis 16
  The Quarksoft Story 18
  The Quarksoft Management System 20
  The Quarksoft Executive Team 23
  Managing the Bureaucracy 26
  Summary and Conclusions 27

Chapter 3 Knowledge Work 29
  The Nature of Knowledge Work 30
  Why Knowledge Work Is Troublesome 31
Why Customers Tolerate Shoddy Software Work 32
Why Software’s Problems Persist 34
Is There a Better Way? 34
A Knowledge-Working Team 35
Team Accomplishments 40
The Future of Knowledge Work 42
Summary and Conclusions 43
References 44

Chapter 4 Managing Knowledge Work 45
Taylor’s Management Principles 46
The Modern Technical Workplace 48
Modern Technical Work 49
Modern Technical Workers 50
The Principles of Managing Knowledge Work 51
Trusting Knowledge Workers 53
The Blame Culture 56
The Need for Trust 57
Trustworthy Knowledge-Working Teams 58
Using Facts and Data 59
Quality Must Be the Top Priority 60
Team Leadership and Support 61
Summary and Conclusions 61
References 62

Chapter 5 Motivating Knowledge Workers 65
Management and Worker Objectives 65
The Nature of Team Motivation 66
The Knowledge-Working Culture 68
The Elements of Trust 69
The Start-Up Problem 70
Self-Management Tasks 71
Making Cost, Schedule, and Quality Plans 72
Recording Data 75
Using an Operational Process 76
Tracking and Reporting Progress 79
Self-Management Training 84
Overcoming Skepticism 85
Summary and Conclusions 86
References 87

Chapter 6 Motivating Knowledge-Working Teams 89
Beckman Coulter 89
Beckman Coulter’s First TSP Team 90
Team Commitment 92
Management Behavior 95
Building Self-Directed Teams 97
Management Issues 98
Management Style 100
Summary and Conclusions 104

Chapter 7 Managing with Facts and Data 107
Auditable Data 107
Auditing TSP Data 111
Using TSP Data 112
Communicating with Data 120
Summary and Conclusions 122
References 124

Chapter 8 Managing Quality 125
Make Quality the Top Priority 125
The Software Quality Problem 128
The Testing Problem 132
Software Quality Economics 136
The Quality Transformation 139
Establishing the TSP Implementation Team 194
Building a Strong Coaching Team 199
The TSP Pilot Programs 201
Implementing the TSP for a Project Team 205
Training 211
The TSP Launch Process 214
Management’s Role in the TSP Process 218
Summary 229
References 230

Appendix C Expanding TSP Use 231
The Overall Implementation Strategy 232
The Overall Rollout Plan 232
Building Local Sponsorship 237
Developing the Local Implementation Plan 239
Building Coaching Capability 240
Other Capability Requirements 244
When and Where to Use the TSP 245
Summary 255
References 255

Appendix D Using the TSP to Manage Programs 257
The Program Management Problem 258
Establishing Aggressive but Realistic Plans 259
Monitoring Program Status 266
Identifying and Resolving Issues 270
Managing Quality 275
Dealing with the Customer 281
Management’s Continuing Responsibilities 283
Summary 285
References 286
Appendix E  Sustaining the TSP    287

   Why Continuous Improvement Is Important   287
   Improvement Examples   288
   Improvement Risks   291
   The Principles of Lasting Improvement   293
   Executive Financial Reviews   295
   The Executive Quality Review   298
   The Executive Role in Continuous Improvement   301
   References   305

About the Authors   307

Index   311
Preface

The problems of managing software have probably been annoying, but they may not have seemed fundamental to your business. You may also have noticed, however, that the amount of software work in your business has been growing and that more and more of your people’s work now looks like software work. This means that, like many other senior executives and managers, you will soon find that an increasing amount of your people’s work will become as hard to manage as software. If that prospect doesn’t frighten you, it should.

This book is for those senior executives and managers who run modern technology-intensive businesses. It describes why software work has always been hard to manage, why more and more work will soon be as hard to manage as software, and what you can do about it. As Robert Frost once said, “The best way out is always through.”

What Frost meant is that dodging problems doesn’t work. You must dig into them, understand them, and then address them. The time has come to address these software management problems. That is the premise on which this book is based: Software work is manageable. To manage it, however, we must first understand why it has always been so hard to manage.

The fundamental reason that software has been hard to manage is that it is a new and different kind of work. The management principles of the past are not suitable for software development. Software engineering is knowledge work, which is nothing like the traditional kinds of labor for which today’s traditional management methods were developed. This book describes the knowledge-based management system that this
kind of work requires, the principles upon which it is based, how this new management system works, and how to introduce it into your organization.

The management challenges of the future concern knowledge work and knowledge workers. In the past, your knowledge workers were primarily doing software engineering work, but now knowledge workers are increasingly involved in all aspects of your business. Software was the first large-scale knowledge-based industry, and its management problems are well known. Now, with knowledge work and knowledge workers pervading modern business, problems that used to be restricted to software groups are becoming common in most parts of every technology-based business.

Knowledge work is not like other kinds of work, and its management problems are unique. Recent research into management methods for knowledge workers has provided new insight into why software work has been so hard to manage. In fact, it is the special characteristics of software work, and in fact of all knowledge work and knowledge workers, that have made software projects so hard to manage. This new understanding has led to a new management system specifically tailored for knowledge work and knowledge workers. When organizations use these new methods, they find that their software work is predictable and manageable and that their people have more rewarding and satisfying work lives. In addition, the quality and profitability of their products are greatly improved.

Modern work in almost all technical fields now involves software, and it also involves design and development practices that are much like those used by software people. When the management methods described in this book are used, all forms of creative work become more manageable. In addition, employee turnover drops, customers are more satisfied, and your people
are more creative. This, in turn, leads to more efficient ways of working, as well as to more attractive and profitable product offerings. This book describes these changes, gives examples of organizations that are working this way today, and outlines some of the benefits they have obtained.

**BOOK ORGANIZATION**

This book is composed of nine chapters and five appendices. The nine chapters describe the new knowledge-working management methods, why they are required, and the principles that guide their introduction and use. The book’s five appendices address many of the questions that executives and managers have raised as they explore these concepts, conduct trial studies, and introduce the methods into their organizations.

Chapter 1 describes the increasing pace of change in the modern marketplace and the threats that all organizations now face from an aggressive set of worldwide competitors. Chapter 2 addresses the bureaucratic problems of increasing corporate size and how modern knowledge-management methods can help to address them. Chapter 3 describes the nature of knowledge work and how properly led knowledge-working teams can resolve many of the intractable problems businesses face today. Chapter 4 covers the principles and methods involved in managing knowledge work, and it explains how these management practices differ from those traditionally followed.

With Chapter 5, the book switches from explaining what knowledge-work management methods are to discussing the challenges of introducing them. Chapter 5 addresses how to motivate knowledge workers to use these new management methods personally. Chapter 6 discusses how to build and maintain the disciplined and collaborative environment that knowledge-working teams need to consistently follow these practices. Chapter 7 covers
the new opportunities for dynamic decision making that become available with these data-driven knowledge-management methods, and Chapter 8 describes the critical nature of quality for knowledge work. Finally, Chapter 9 describes how to make this vision of the future come true in your organization.

Following these nine chapters, the book’s five appendices provide more detailed guidance on introducing and using these management methods. Hundreds of organizations have now used them, and the appendices encapsulate the guidance we have found most effective in making their efforts successful. Because all organizations are different, and because many new and unfamiliar situations will arise, your management team will need substantial guidance in following the steps outlined in this book. The appendices contain answers to the most common questions and step-by-step guidelines on how to proceed.

READING THIS BOOK

We suggest that you read the nine chapters in order. They present a logical sequence of concepts that builds a complete high-level overview of these new management methods. Building on this conceptual background, the appendices provide a deeper level of information. They can be used as a reference or guideline for introducing and using the methods. The five appendices start by covering common questions, then trial introduction and piloting, and finally the broader issues of adoption and use. We suggest using this book in one of the following ways:

- You can read this book’s nine chapters to gain an overview understanding of these knowledge-management methods, why they are needed, how they could help improve your organization’s performance, and how to introduce and use them.
• You can use the book, particularly its appendices, to answer questions you might have before deciding to introduce and use these methods.

• Once you have decided to introduce these methods, your people can follow the guidance in the appendices to conduct pilot studies and to then introduce the methods broadly in your organization.

WHY WE WROTE THIS BOOK
We first tried using the management methods described in this book in 1996. Since then, we have introduced them to hundreds of organizations with thousands of managers and developers. During this time, we have learned a great deal about how these methods can be used to address the many situations that arise in modern technology-intensive businesses. We have worked with organizations of every imaginable variety, and while no single book could cover all of the issues of introducing and using these methods, we were able to encapsulate the most frequently encountered issues into a single moderate-size volume. This book is that volume.
First and foremost, we must thank three organizations and their executives for letting us describe their experiences with these methods. Blanca Treviño and her people at Softtek have been instrumental in getting these new knowledge-management methods more widely recognized, particularly in Mexico, where Softtek is a leading supplier of software and software services. Cesar and Carlos Montes de Oca and Ricardo Delgadillo of Quarksoft were among the first users of these methods, and their experiences have helped to get these methods better known and to build the experience base needed for their widespread adoption.

We also thank the people at Beckman Coulter for allowing us to describe their work. Humayun Qureshi kindly allowed us to quote his early reactions to these teamwork methods, and Rick Marshall, Carl Wyrwa, John Hetzler, and Larry Whitford have also been instrumental in getting the methods properly introduced in their company. Tim Lancaster and his team also took the time to review our description of their experiences and to ensure that what we have said properly reflects what happened. We thank them all.

Unfortunately, the team we described in Chapter 3 was and still is doing highly classified work for the U.S. Department of Defense, and we were unable to get their story cleared for release. We have worked with them for several years, however, and they have pioneered many of the practices that have turned out to be most successful, particularly for large, distributed, and multidisciplined knowledge-working teams. We salute them for
their groundbreaking achievements and regret that we cannot recognize them by name.

We also thank Jason Ziemer of NAVOCEANO (NAVO) for telling us about his team and letting us use his story to illustrate some key points in the appendices. The members of his team were Carissa Bedford, Lleo Garner, Brook Bell, and Bobby Roots, and they did an extraordinary job of improving the services their group provides to U.S. naval forces in battle zones throughout the world.

In addition to getting help and support from many of our users, we have also been fortunate in having a large group of reviewers who have kindly given their time to review drafts of the book’s chapters and to provide helpful comments and suggestions. These reviewers are Daniel Burton, Bob Cannon, David Carrington, Noopur Davis, Agustín De La Maza, Carlos Montes de Oca, Julia Mullaney, Jan Philpot, Marsha Pomeroy-Huff, Jim Sartain, David Scherb, Gregory Such, and Alan Willett. We thank them all for their help.

We also thank the management team at the Software Engineering Institute (SEI) for their support of our work. Anita Carleton, Clyde Chittister, and Paul Nielsen have supported us over the many years it has taken to develop, refine, and transition these methods into increasingly widespread use. We cannot thank them enough for their support.

Bill Thomas, who manages technical communications at the SEI, kindly helped with the final editing and release of the manuscript for publication, and Peter Gordon and his staff at Addison-Wesley have done a masterful job of producing this volume on an aggressive schedule. Without their help and support, this book would not have been possible.
Change is a fact of life. The world is changing faster than ever before, and the challenges of tomorrow will almost certainly be different from and more demanding than those of today. While no one can say precisely what these challenges will be or how to prepare for them, some things are pretty obvious from our recent history. This book describes the nature of these challenges and a strategy that will help you to address them.

CORPORATE CHURN

The first new phenomenon that is obvious from our recent history is corporate churn. Industry leaders always fail, and sometimes they fail surprisingly quickly. Consider, for example, what has happened to the largest and most successful U.S. businesses. In the 24 years from 1956 to 1980, 24 firms dropped off the Fortune 500 list every year. However, in the 24 years from 1982 to 2006, that rate increased to 40 firms every year [Economist 2009]. That comes to 960 seemingly successful firms switching from being winners to being losers in 24 years.

Joseph Schumpeter studied the reasons for corporate churn and, in 1942, published a book called Capitalism, Socialism and Democracy [Schumpeter 1942]. In this book, he explains why organizations grow, prosper, and die. He called this concept “creative destruction.” While his ideas were not well accepted at
the time, they are now widely recognized as perceptive and pre-
scient. He describes why economies are in constant flux in the
following way:

The fundamental impulse that sets and keeps the capitalist engine
in motion comes from the new consumer goods, the new meth-
ods of production or transportation, the new markets, the new
forms of industrial organization that capitalist enterprise creates.

Schumpeter’s work also suggests why market leaders are so
often surprised by their newer and more agile competitors. It is
because the rules of the game keep changing.

As soon as quality competition and sales effort are admitted into
the sacred precincts of (economic) theory, the price variable is
ousted from its dominant position.

While Schumpeter’s ideas sound reasonable and are now
widely accepted, he does not say where these giant-killing new
competitors come from. Just who are they and why are they able
to topple large and established businesses?

Just as they have in the past, the challengers to industry lead-
ers will come from unexpected quarters. These newcomers will
be entrepreneurs who have found some innovative new way to
make themselves unique. This has been true in a great many
industries, and as indicated by the relative vigor and productivity
of small businesses, it is likely to remain true in the future. The
innovative advantages of small businesses are indicated by the
fact that in the United States, small businesses produce many
more new jobs and grow much faster in percentage terms than
their larger competitors.

Consider, for example, a recent Small Business Administration
study [Terleckyj 1999]. Over a three-year period, new and small
companies accounted for only 25% of employment but for 39% of job growth. This means that, on average, small businesses grew nearly 60% faster per capita than their larger competitors. Clearly, being a small business has had some pretty significant advantages. In Chapter 2, we discuss how small businesses operate and examine some of the reasons for their superior performance.

Change is the name of the game for modern industry. Those organizations that do not recognize and plan for the often obvious future trends of their industries will almost certainly be replaced, and it could happen very quickly.

**KNOWLEDGE WORK**

Assuming that Schumpeter was correct and that the rules of the game are continuously changing, a high priority for executives should be identifying those changes that will impact their businesses. Once these changes have been identified, executives and senior managers can better judge how and when to make the adjustments needed to capitalize upon them. The key challenge, of course, is determining what these future changes are likely to be and how to take advantage of them before your competitors do.

Peter Drucker devoted much thought to the analysis of corporate management. More than 50 years ago, in his 1957 book *Landmarks of Tomorrow*, he outlined the key challenges he saw for future managers and executives [Drucker 1957]. He concluded that learning how to manage knowledge work would be the key management challenge of the next century. He described knowledge work as work that is done in the workers’ heads instead of with their hands. He concluded that knowledge work would soon be the most critical and the highest-valued form of labor. Later, in his book *The Age of Discontinuity* [Drucker 1969], Drucker wrote:
To make knowledge work productive will be the greatest management task of this century, just as to make manual labor productive was the great management task of the last century.

More recently, in an article in the *Harvard Business Review* [Drucker 1997], he also said:

> The productivity of knowledge workers will not be the only competitive factor in the world economy. It is, however, likely to become the decisive factor, at least for most industries in developed countries.

Drucker was the premier management thinker of the twentieth century, and it behooves us to take his views seriously. This book, in fact, does just that. It starts from the premise that knowledge work is the work of the future, and that the organizations that first recognize and capitalize on this fact will be the industrial leaders of tomorrow. Ask yourself this question: “If Drucker and Schumpeter were right, what should I do to capitalize on the opportunities of the knowledge-working age?”

This book answers that question.

**THE URGENCY OF CHANGE**

As is clear, both from Schumpeter’s and Drucker’s views and from the current rate of industrial churn, change is a fact of competitive life. Furthermore, the rate of change is accelerating. While this is not particularly surprising, what is surprising is how often changes have come as surprises, even to very successful firms. Many of these surprises, however, happened not because the new ideas were unknown in advance to the leading firms. In fact, many of these companies actually *invented* the new methods that ultimately destroyed them.
Eastman Kodak still survives, and it invented many of the technologies in modern digital photography. Eastman Kodak, however, is not a market leader in digital photography. Similarly, Texas Instruments invented many of the methods used in developing and manufacturing integrated circuits, but TI no longer leads the semiconductor industry. The reason organizations are often surprised by technologies they already know is that they refuse to accept the implications of what they know.

For example, IBM management knew very well that the personal computer was coming and that it would be big business. They also knew that programming was an increasingly important part of the computer business. However, because IBM never put these two facts together, the company literally gave the PC programming business to Microsoft. Within a few years, Microsoft jumped from being a small start-up to being a major corporation with a market value even greater than that of the once-mighty IBM.

IBM management’s lack of vision probably can be attributed to the fact that its executives and senior managers had long thought of programming as an expense. Until 1968, IBM had always given its software and systems engineering services to its customers as a part of its hardware support. Even 13 years later, when IBM introduced the PC in August 1981, its executives could not visualize software as a potentially profitable business opportunity. Today, IBM’s software and systems engineering services generate more revenue and are more profitable than its hardware businesses. In fact, only a few years ago, IBM actually spun off its printer and disk drive hardware businesses. Old attitudes are hard to change, and IBM management’s outdated attitudes were nearly fatal for the company.

The problem in large corporations is not a lack of vision; it is a lack of courage—the courage to recognize that the world is
changing. Leaders must recognize that the things that made them successful in the past are not likely to be the things that will keep them successful in the future. The question, of course, is: “What will make organizations successful in the future?” The answer is that nobody really knows, and those who say that they do will almost certainly be proven wrong. What we do know, however, is that the problems that both large and small businesses will soon face will be different from those of today, and they will principally concern management. We also know that these problems will likely be of two types.

The first type of future problem concerns questions of scale. Small businesses typically grow faster and are more dynamic than larger ones, at least in part because they are not burdened by the problems of size. The question, then, is how a business can grow and be successful without being choked by its own size. Businesses have long faced this problem, but with the Internet and the new flexibility of the “flat world,” these size problems are now quite different from what they were just a few years ago [Friedman 2005]. For example, in the past, the big issue was numbers of people and spans of control. Today, while we still have the span-of-control issue, the scale problems also include managing geographically distributed groups, mixed cultures, and heterogeneous technical teams. Clearly, mastering the problems of size in this increasingly complex environment will be more challenging and more important than ever before.

The second set of future problems has been with us for some time but has largely been confined to the specialized field of software. These problems concern knowledge work and knowledge workers. As Drucker pointed out, knowledge work is work that is done in the workers’ heads rather than with their hands. While we have long had knowledge workers, traditionally there have been only a few of them on most projects. The vast bulk of
the work has been done by technicians and less skilled laborers or factory hands. Today, most technical work looks more like software engineering, where the workers make creative decisions and produce work products on computers. Knowledge work is the key to the future, and those who master this discipline will be the industry leaders of the twenty-first century.

The fact that knowledge work requires a new management strategy and style is obvious from the history of the software business. Software projects have always been hard to manage, and few software groups, even today, can consistently deliver quality products on committed schedules or for anywhere near their planned costs. Software development was the first technology to involve large-scale knowledge work, and while software work has always been a management problem, traditionally it has involved only a small part of most businesses.

As knowledge work becomes pervasive, new corporate management strategies will be needed. Software and other forms of knowledge work are becoming increasingly important as they involve a greater proportion of business operations and more executives and senior managers recognize that software is now the controlling element of their operations. Software controls production schedules, optimizes prices, manages costs, and calculates profits. When new business strategies are implemented, software is the gating element, and when products are late, the software work is usually furthest behind schedule. In almost all areas of modern science and industry, products are developed with methods that look very much like software development. Just about all future systems and product development work will have to be managed as knowledge work. As noted in the next chapter, many aspects of the corporate world could benefit by being managed with these methods today.

The methods described in this book are designed for knowledge work and knowledge workers. These same methods will
also help you to address other key issues, including those of size. The next example shows how the new knowledge-management methods can help executives and senior managers manage their businesses.

THE SOFTTEK STORY
Blanca Treviño is CEO of Softtek, a Mexican company with headquarters in Monterrey. Softtek is the largest independent IT service provider in Latin America with almost 6,000 employees and offices in 13 countries. The company was founded in 1982 and grew steadily until 2000 when Ms. Treviño became the CEO. Since then, the corporate growth rate has exceeded 30% per year. Softtek has long operated in North and South America and Europe, but it recently opened operations in five more countries: Venezuela, Chile, Paraguay, England, and China.

From the outset, Softtek focused on quality as a key marketing discriminator. It was one of the first Latin American companies to implement the Software Engineering Institute’s (SEI’s) software development practices and was assessed at CMMI\(^1\) level 3 in 2000. In 2004, Softtek’s development groups achieved the SEI’s coveted level 5 rating, the highest CMMI level. While this was an important achievement, it was not unique to Softtek. In fact, many of its competitors had also achieved the same high rating. To compete and be successful, the Softtek managers had to maintain their quality rating and also devise some new way to provide their customers with unique products and services.

Ms. Treviño knew that Softtek had to be unique to stay competitive and continue to grow. Trying to compete on price alone would be a losing game because, although Mexican labor costs

\(^1\) CMMI, Capability Maturity Model Integration, is an evaluation method and model devised and supported by the SEI to rate the process capability of technical organizations.
were below those in the United States, Canada, and Europe, companies from India and China had an even lower cost structure. She therefore established a corporate goal of being the highest-quality software provider in Latin America and among the best in the world. Her objective was to offer such high-quality products and to provide such predictable and responsive services that Softtek’s customers would make it their preferred supplier.

To achieve her corporate goal, Ms. Treviño knew that Softtek had to make some significant changes to both maintain and improve the productivity and quality of its engineering work and to differentiate itself from the competition. She therefore had her technical groups introduce the method the SEI had developed for knowledge work, the Team Software Process (TSP). The success of the TSP in improving engineering performance, coupled with Softtek’s “near-shore” advantages in the U.S. marketplace, has enabled the company to attract a growing volume of profitable business and to expand its IT services business rapidly.

**THE SOFTEK EXPERIENCE**

Starting in 2007, Softtek has been introducing the methods described in this book. The company’s early TSP pilot projects were highly manageable, its people had more rewarding and satisfying work lives, and its customers were increasingly satisfied. In those parts of the business that used these new methods, Softtek improved its project performance, enhanced its product offerings, and improved its employees’ quality of work life. It also improved profitability and accelerated corporate growth.

**Project Performance**

The new TSP knowledge-working methods also helped Softtek’s technical teams improve their record of on-time and within-cost development performance. One large global financial institution
even challenged Softtek to become its highest-performing software supplier in Latin America. After introducing the methods described in this book, and for almost a full year to date, Softtek’s development teams have not missed any of this customer’s quality or on-time delivery goals. This performance has earned them their customer’s highest rating as a services provider, and for the next year, Softtek became that customer’s IT services vendor with the highest proportion of the customer’s business. In fact, the customer has even asked its other software vendors to consider using the TSP methods described in this book.

**Product Offering**

While Softtek had previously had an excellent record of delivering products on schedule and within contracted costs, its development performance has recently improved to such an extent that it can offer more development contracts on a fixed-price basis. This convinced many clients to move from a cost-plus to a managed-services delivery model for their projects. The customers are happy to have a more predictable cost structure, and Softtek has a higher-valued set of customer contracts. While some of this improvement was due to improved project management, a major part was a direct result of Ms. Treviño’s drive for superior quality.

Product quality became so predictable and Softtek’s products had so few defects that the company decided to offer quality guarantees. In selected cases, Softtek even included quality warranties in development contracts and promised to refund the customer’s money for every defect found in customer acceptance testing or use. Initially, Softtek set its warranty budget at one-tenth of what such a warranty would have cost the company historically, but with its TSP teams, its costs have typically run well below that. While Softtek did make a few refunds, the custom-
ers liked the guarantees, and Softtek’s competitors were unable to match them without losing money. To date, the guaranteed Softtek products have had so few defects that the company has decided to offer quality guarantees on a more regular basis.

**People**

With the TSP methods, the developers’ quality of work life has greatly improved. Now, instead of suffering through all-nighters during final test, the developers can balance their workload and recover from schedule problems without impacting their projects. They are now home for dinner nearly every night, and they can take scheduled vacations without fear of unanticipated project crises.

This work-life improvement also has had important company benefits. For the TSP teams, turnover has decreased to one-quarter the turnover rate of non-TSP teams, and the company is better able to retain its most experienced and productive talent. This maintains team stability, saves recruiting costs, and improves the company’s return on its personnel investment. With its improved corporate image and reputation, particularly among students, Softtek also finds it easier to recruit and retain the best engineering talent.

**WHAT NEXT?**

Judging by the abominable history of the software industry, Softtek’s accomplishments might seem extraordinary. But the fact is, in any other industry this level of performance would earn only a passing grade. Just delivering quality products on schedule and within planned costs is what product developers are paid to do. For software work, however, this is a significant achievement, and it is one that all businesses must soon learn to achieve with all types of knowledge work.
As you read this book, remember that the problems you face in the future will be different from and more challenging than those you face today. Chapter 2 describes the issues of managing organizations as they grow. It also discusses small businesses, bureaucracy, and the problems of managing and controlling a growing business. Chapter 3 gives an overview of knowledge work, how the knowledge-working teams of the future will work, and the issues those teams must address to be successful. Starting with Chapter 4, we describe how to change your organization to better utilize your knowledge-working people and to capitalize on the enormous potential opportunities of the knowledge-working economy.

SUMMARY AND CONCLUSIONS

This chapter describes why even very successful organizations face an urgent need to change, why these changes must be a high priority, and the issues to be addressed in making the changes. The chapter makes four principal points:

1. The corporate leaders of today are not likely to be the leaders of tomorrow because of the accelerating pace of industrial change, a phenomenon that Joseph Schumpeter called “creative destruction.”

2. As Schumpeter also pointed out, the rules of successful competition are changing, and the things that made today’s businesses successful are not the things that will make businesses successful in the future.

3. Businesses face two principal challenges in making the management changes that will allow them to remain competitive in the marketplace. The first challenge is managing the problems of size; this problem is discussed more fully in Chapter 2.
4. The second challenge is making knowledge work productive. As Peter Drucker said, this will be the decisive factor for future success. This topic is discussed further in Chapter 3, and the balance of this book outlines a strategy for addressing the challenges of managing knowledge work and knowledge workers.

REFERENCES


This page intentionally left blank
Index

A
accelerating projects, 149–152
acceptance phase, Conner’s change model, 192–193
acceptance tests
  benefit of implementing TSP, 170
evaluating pilot projects, 204
productivity gain with TSP, 177–178
as quality-cost element, 299
Softtek product offering and, 10
accuracy
  communications, 120
  limitations of EV measures, 269–270
as performance indicator for TSP team, 40–41, 107, 111, 122
ACM (Association for Computing Machinery), 307–308
acquisition phase, overcoming resistance of suppliers in, 234
Acquisition Support Program (ASP), SEI, 165
Activision, using TSP, 166
Adobe, using TSP, 164–165
A/DR (appraisal-to-development ratio) measure, quality
executive review of, 300–301
overview of, 276–279
program reviews, 280–281
Aerospace Software Engineering Award, 308
aircraft
  managing quality of safety on, 127–128
  operational procedures example, pilot preflight checks, 78
Allied Signal, 113–114, 281–282
American Express, 19
Anderson, Art, 293
appendices, roadmap to, 162–164
application software, developed with TSP, 167–168
appraisals
  cost-of-quality measure for, 126–127
  management conducting performance, 180
  managing quality of, 276
  appraisal-to-development ratio. See A/DR (appraisal-to-development ratio) measure, quality
ASP (Acquisition Support Program), SEI, 165
assessment
  management not using personal data for, 108–109
  of program plan, 261–263
  reviewing program plan for risk, 265–266
  risk, during TSP launch, 215
auditable data
  accuracy of TSP, 120
  handling poor performers, 109–111
auditable data (continued)
  overview of, 107–108
  personal data, 108–109
auditing
  for process fidelity problem, 245
  TSP data, 111–112
automobiles
  blame culture and defects in, 56–57
  safety quality of, 127–128
  Toyota stuck-accelerator problems, 128
autonomy, of knowledge workers, 52

B
Bartko, Peter, 261
Beckman Coulter
  first TSP team example, 90–92
  leadership goal of, 147
  management behavior, 95–97
  overview of, 89–90
  quality management example, 141–142
  team commitment, 92–94
  training managers in TSP, 100
Beckman Instruments, 89
benchmarks
  establishing at team level, 113–114
  motivating continuous improvement with, 203–204
  tracking quality plan using, 264
beta testing, development of, 135–136
biomedical testing, at Coulter Corporation, 89
blame culture, 56–57
Boeing Corporation, 308–309

bonus pay
  Quarksoft people evaluation and, 26
  Taylor’s management principles, 47
  trusting knowledge workers, 54–55
bottom-up strategy,
  implementation, 232
broad implementation phase. See extending TSP use
building motivation, 155–156
building trust, problems with knowledge workers, 58–59
bureaucracy
  fundamental purpose of, 15–16
  managing, 26–27
  Quarksoft executive team, 23–26
  Quarksoft management system, 20–23
  Quarksoft story, 18–20
  software crisis, 16–18
  summary review, 27–28
  why organizations need, 15–16
business manager, TSP launch process, 215

C
C1105 program, IBM, 290–291
capability
  building coaching, 176, 240–244
  building internal, 163, 177, 235–236
  building TSP management, 190
  ensuring role-manager team, 254–255
  other requirements, 244–245
Capability Maturity Model (CMM), 291
Capability Maturity Model Integration. See CMMI (Capability Maturity Model Integration)

*Capitalism, Socialism and Democracy* (Schumpeter), 1–2
career path, coaching as, 241–242
Carnegie Mellon University, 18
on applying TSP principles to project, 36
Humphrey joining SEI at, 307
SEI established at, 165
study on effectiveness of bonus payments, 54–55
CERT (Networked Systems Survivability Program), 165
certification
coach, 243
SEI services, 165
TSP, 303–304
change
establishing energetic implementation team for, 190
taking advantage of future, 3–4
urgency of, 4–8
change management principles
commitment, 191–194
executive sponsorship, 194
overview of, 190–191
when TSP might not make sense, 245
*Chaos Reports* (Standish Group), 220–221
Clemens, Samuel (Mark Twain), 108
CMM (Capability Maturity Model), 291
CMMI (Capability Maturity Model Integration)
definition, 8
evolution of, 307
improvement risk and, 291–293
organizational processes satisfying, 76
on pitfalls of taking shortcuts, 77
Quarksoft R&D improvement, 25–26
Softtek story, 8–9
Softtek’s objectives for TSP implementation, 166
who is using TSP, 166
*CNN Expansion*, and Quarksoft, 19
coaching
analyzing value of, 114–115
building capability for, 176, 240–244
as career path, 241–242
certification, 303–304
costs of implementing, 169–170
as critical success factor, 184–185
development strategy, 200–201
going started with TSP, 180–181
house building in record-time example, 152
identifying candidates, 243
identifying initial TSP coaches, 199–200
monitoring transition to TSP, 245
return on investment analysis, 171–172, 174
reviewing long-term plan for, 197
status review of, 229, 243–244
as support, 153–154
training, 236, 243
TSP launch process and, 215–218
coding work
defects injected/removed in, 136–137
measuring project status, 81–82
software quality transformation, 139–140
cohesion, team, 101–103
commitment
achieving during launch, 215–216
benefits achieved with TSP, 177
example, Beckman Coulter’s first TSP team, 90–92
example, Quarksoft team, 21
management behavior after seeing team, 95–96
motivating team, 92–94
team, 92–94
team accomplishments, 40–42
troublesome nature of knowledge work, 32
commitment phase, Conner’s change model, 192–193
communication
accuracy, 120
establishing trust with management, 80–84
getting started with TSP, 198–199
hierarchy, 120–122
maintaining program-wide, 284–285
comparative measures
evaluating pilot projects, 203–204
launching TSP implementation team, 196
compatibility, and TSP launch, 209
competition
changing rules of successful, 2, 12
factors changing nature of, 146
complexity, and future of knowledge work, 42
confidence
building self-directed teams, 97
of management in knowledge-working teams, 96–97, 104
safety-critical systems and restoring public, 128
using TSP data with, 122
confidence intervals, product size estimates, 263
configuration testing problem, 133
conflict, between workers and managers, 54
Conner, Darryl, 192
Conner’s change model, 191–194
continuous improvement, sustaining
avoiding dead-end improvements, 292–293
avoiding simplistic measures, 291–292
examples of, 288–291
executive financial reviews, 295–298
executive quality reviews, 298–301
executive role in, 301–305
importance of, 287–288
motivational strategy of, 198
overview of, 287
principles of, 293–295
references, 305
COQ (cost-of-quality) measure
as controllable cost in development, 297–298
executive review of, 298–300
overview of, 126–127
using TSP data, 115–116
corporate churn, 1–3
costs
confidence in knowledge-working team and, 96–97
cost-of-quality (COQ) measure, 126–127
determining job, 265
executive financial reviews of, 295–298
executive quality review of, 298–301
expected benefits of TSP implementation, 177
future of knowledge work and, 42
doing the job, 169–170
making plan for, 72–73
management vs. knowledge worker objectives, 66
objectives TSP will not help you achieve, 178
Quarksoft management system, 20, 22–23
reasons to manage quality, 125–126
recovering training, 212
reviewing TSP implementation plan, 197
Softtek experience, 9–11
Softtek story, 8–9
software quality, 128
troublesome nature of knowledge work, 7, 32
TSP return on investment analysis, 171–174
Coulter Corporation, 89
courses, TSP and PSP, 213
Covey, Stephen, 120–121
creative destruction
corporate churn, 1–3
future of, 11–12
importance of continuous improvement, 287–288
knowledge work, 3–4
overview of, 1
references, 13
shoddy software work and, 33
Softtek experience, 9–11
Softtek story, 8–9
summary review, 12–13
urgency of change and, 4–8
creativity
fostering in knowledge workers, 146
of modern technical workers, 51
Quarksoft management system, 21
credibility
hierarchical communication problems concerning, 120–121
teams establishing with management, 92, 94, 104
crisis prevention
ensuring effective program-wide communication, 284
identifying and resolving issues, 270
crisis-driven organizations, 295
critical success factors, 184–185, 228–229
culture, knowledge-working, 68–69
customer interface manager, role of, 253
customer satisfaction analysis
evaluating pilot projects, 204–205
maintaining priority for, 301–302
using TSP data, 116
customers
  overcoming resistance to TSP, 235

program management
  problems, 259, 281–283
  status reports, 226
  tolerating shoddy software work, 32–34
  TSP launch process, 215

TSP launch process, 215 cycle postmortem. See postmortems

cyclic development strategy, TSP, 224–225

D

data, gathering. See also facts and data, management with error problem in testing, 133
  knowledge workers recording, 75–76
management rewards for using, 228
managing quality at beginning using, 276
objectives TSP will not help you achieve, 178
precise data, 20
prioritizing quality using, 284
resource estimates in aggressive program plan, 263–264
results of TSP in other organizations, 71
tool support for, 119
dead-end improvements, risk of, 292–293
design
  defects injected in, 136–137
  measuring project status, 81–82
  team accomplishments, 41
team example, 40–42
  TSP quality process, 139–141
design manager
  launch management review meeting, 222
  maintaining program-wide communication, 284
  role of, 253
evaluating pilot projects, 204
executive quality review of costs for, 298–301
gains when implementing TSP, 170, 178
injecting and removing in development, 136–139
problem of, 130–131
quality management finding and fixing, 276
in safety-critical devices, 127–128
software quality facts, 129
software quality transformation, 139–140
testing finding only fraction of, 130–131
testing problem, 132–134
TSP return on investment analysis, 173
why defective systems work, 135–136

Delgadillo, Ricardo Vidrio, 18–19, 23–24
Dell, 19
Deming, W. Edwards, 33
Department of Defense. See DoD (Department of Defense)
role-manager team for, 253–254
TSP weekly meeting topics, 217
development
defect problem in, 130–131
injecting and removing defects in, 136–137
pilot project selection criteria, 208
as quality-cost element, 299
removing most defects before testing in, 131
Softtek performance during, 9–11
software quality transformation, 139–140
development organizations, 250–251
Diffusion of Innovation (Rogers), 193–194
dimensions of excellence, defining, 156–157
discipline, team-based, 21
distributed teams, using TSP, 248
DoD (Department of Defense)
improvement risk, dead-end improvements, 292–293
improvement risk, simplistic measurement problem, 291–292
operational process and, 77
organizations using TSP, 166
SEI Acquisition Support Program working with, 165
team accomplishments, 40–42
team example, 35–40
Drucker, Peter
definition of knowledge workers, 6, 30
on knowledge worker management, 52
on knowledge worker productivity, 3–4, 13, 160
on making commitment, 215
on Taylor’s management principles, 45
duration, project
assessing plan feasibility, 224–225
characteristics of TSP project teams, 169
cost of implementation and, 169–170
pilot project selection criteria, 209
E
ey early warning of problems, identifying and resolving issues, 270
earned-value. See EV (earned-value)
Eastman Kodak, 5
EBITA, financial management, 24–26
economics
of coaching, 154
reasons for corporate churn, 2
reasons to manage quality, 126–127
software quality, 136–139
ECRs (engineering change requests), NAVO functional teams example, 249
edicts, conditions for, 274–275
education, of modern technical workers, 50
efficiency, executive financial reviews of job, 297–298
Embry-Riddle Aeronautical University, 308
employee satisfaction analysis
  evaluating pilot projects, 205
  using TSP data, 116–117
empowerment
  of knowledge workers, 42
  motivating team to meet goals by, 67
engineering change requests (ECRs), NAVO functional teams example, 249
engineering staff
  executive financial reviews of, 296
  TSP return on investment analysis, 171–172
environments, TSP independence from, 209
ERP systems, built with TSP, 167–168
estimates, of product size, 73–74
EV (earned-value)
  definition, 25
  example, describing project status, 82–83
  example, status report, 267–269
  financial management, 25
  identifying and resolving issues. See issues, identifying and resolving limitations of, 269–270
  monitoring program status, 266–267
  reporting project status, 83–84
evaluation measures, TSP teams
  evaluating pilot projects, 202–204
  SEI services, 165
  for team plan, 221–223
example
  airline pilot preflight checks, operational procedures, 78
automobile defects, the blame culture, 56–57
Beckman Coulter quality management, 141–142
Beckman Coulter’s first TSP team, 90–92
building team motivation, 261
cutting response time from U.S. Navy’s fleet, motivation, 248–249, 260
default improvements, improvement risk, 292–293
defect rates of experienced developers, 129–130
earned value (EV), describing project status, 82–83
earned value (EV), financial management, 25
failed program, continuous improvement, 289–290
hierarchical communications,
  communication hierarchy, 120–121
hospital procedures, operational processes, 78–79
house building, breaking world records, 149–152
IBM, customer satisfaction analysis, 116
IBM beta testing, quality, 135–136
knowledge-working team, 35–40
lighthouse, communication hierarchy, 120–121
long-pole report, monitoring program status, 267–269
Microsoft training of personnel in India, overcoming resistance, 234
NASA goals—faster, better, cheaper, 147–149
NAVO, using TSP, 166
priority of customer satisfaction, 301–302
Quarksoft. See Quarksoft example
Quarksoft financial analysis, using TSP data, 117–118
recognition of quality work, support, 154–155
safety-critical devices, quality, 127–128
simplistic measurement problem, improvement risk, 291–292
Softtek experience, 9–11
Softtek story, 8–9
software crisis, bureaucratic procedures, 16–18
successful program, continuous improvement, 290–291
team accomplishments, 40–42
Toyota stuck-accelerator problems, quality, 128
Xerox, quality, 136–137, 139–140
excellence, dimensions of, 156–157
execution as element of leadership, 157–158
of TSP launch, 216
Executive Seminar, for TSP adoption, 194
executive team broadening TSP use, 237–238
building sponsorship, 190, 194 as critical success factor, 184–185
financial reviews by, 295–298
implementation strategy, 232 maintaining priority by reporting to, 237
overcoming resistance to TSP, 234–235
quality reviews by, 298–301
Quarksoft, 23–26 resolving productivity issues, 29, 35
role of, 301–305
training, 212–214
 extending TSP use coaching capability, 240–244
defined, 190
distributed teams, 248
functional teams, 248–249
implementation strategy, 232
large-scale programs, 249–253
local implementation plan, 239–240
local sponsorship, 237–238
multidisciplined teams, 247
other capability requirements, 244–245
overview of, 231–232
references, 255
role-manager teams, 253–255
rollout plan, 232–237
summary review, 255
when and where to use, 245–246
external resistance, overcoming, 234
external resources, getting started using, 176–177, 184
F
facilities analysis, 117
facts and data, management with auditable data, 105–106
auditing TSP data, 111–112
communication accuracy, 120
facts and data, management with
(continued)
communication hierarchy,
120–122
handling poor performers,
109–111
overview of, 59–60
personal data, 108–109
productivity, 124
references, 124
summary review, 122–123
using TSP data. See TSP data,
using
Fagan, Michael, 289–290
failure costs, 126–127
feasibility, assessing plan, 224–225
federally funded research and
development center
(FFRDC), 165
FFRDC (federally funded research
and development center), 165
financial management
auditable data, 107–109
Quarksoft executive team, 24–25
financial performance analysis
sustaining continuous
improvement with, 295–298
using TSP data, 117–118
financial reviews, executive,
295–298
fixed-price development contracts
commitment ownership and, 22
product offering under, 10
troublesome nature of
knowledge work and, 32
Ford, Henry, 45
Friedman, Thomas, 146, 288
functional teams, using TSP,
248–249
future
of creative destruction, 11–12
of knowledge work, 42–43
leadership failure to realize
problems of, 5–6
taking advantage of changes in,
3–4
G
Galbraith, John Kenneth,
190–191
game industry, software
multidisciplined teams using
TSP, 247
software products using TSP,
167–168
using TSP, 166
Garrett, Scott, 89–90, 100, 147
gathering data. See data, gathering
General Motors (GM), creative
destruction and, 33
getting started with TSP
benefits of, 177–179
best place to begin, 176–177
change management principles,
190–194
coaching team, 180–181,
199–201
critical success factors, 184–185
implementation team, 194–199
introduction strategy, 188–190
launch process, 214–218
management style, 179–180
management’s role in. See TSP
launch, management’s role in
organizational considerations,
179
overview of, 187–188
pilot programs, 201–205
process improvement, 182–183
project management impact, 181–182
project team, 205–211
QA, 182–183
references, 230
summary review, 229–230
test group impact, 182–183
training, 211–214
getting started with TSP,
management’s role
assessing plan feasibility, 224–225
evaluation of team plan, 221–223
launch meeting 1, 219–220
launch meeting 9, plan presentation, 220–221
launch meetings 2-8, 220–222
overview of, 218
reporting, 226
status review issues, 227–229
GM (General Motors), creative destruction and, 33
goals
accomplishment of, 41
commitment ownership and, 22
executive team and, 24–26
house building example, breaking world records, 149–152
leadership, 147
maintaining team cohesion, 102–104
NASA faster, better, cheaper, 147–149
nature of team motivation, 66–67
project launch, 36–37
Quarksoft story, 19
relaunch review of, 38–39

Softtek experience, 10
Softtek story, 9
TSP distributed teams, 248
TSP implementation team, 196
TSP multidisciplined teams, 247
greed, distrust and, 54
Guitar Hero, using TSP, 166

H
hardware failures, testing problem, 133
Harvard Business Review, 4
health care systems, software products built with TSP, 167–168
Hetzler, John, 90, 100
hierarchical communications example, communication hierarchy, 120–121
hierarchical structure, of organizational bureaucracy, 15
Honeywell-Allied Signal merger, 281–282
house building example, breaking world records, 149–152
Humphrey, Watts, 85–86, 307–308

I
IBM, 307
beta testing development at, 135–136
customer satisfaction analysis example, 116
customer satisfaction program of, 301–302
data on defects escaping testing, 131
knowledge work as highest-value part of, 43
IBM (continued)

- length of time to implement TSP, 175–176
- and PC, urgency of change, 5
- successful improvement at, 290–291

Illinois Institute of Technology award, 308

implementation manager
- launch management review meeting, 222
- role of, 253
- TSP weekly meeting topics, 217

implementation strategy
- developing local implementation plan, 239–240
- overall, 232
- steps in TSP introduction strategy, 188–190

implementation team, establishing TSP
- communicating idea, 198–199
- introduction strategy and, 188–190
- launching, 195–197
- motivating participation, 198
- overview of, 194–195
- reviewing plan for, 197
- incentive, vs. performance, 54–56

industry leaders
- eventual complacency of, 33
- reasons for corporate churn, 1–3
- urgency of change and, 4–8

instructors
- certification for, 236, 303
- TSP launch requiring PSP/TSP, 244

Instrument Systems Development Center, 90–91

interim results, used by comparative measures, 196–197
- internal capability, overall rollout plan, 235–236
- internal resistance, overcoming, 234
- Internet problems of size and scale, 6
- introduction strategy, TSP, 188–190

Intuit
- results using TSP, 164–165
- TSP return on investment analysis, 172–173

issues, identifying and resolving
- estimating accuracy, 271–272
- managing task time, 274–275
- overview of, 270–271
- program management problems, 259
- task-time experience, 273–274
- task-time measure, 272–273

IT functional teams, 248

iterative development strategy, TSP, 224–225

IV checklist, operational procedure example, 78

J

job efficiency, executive financial reviews of, 297–298

job steps
- in modern technical workplace, 48–49
- nonlinear nature of modern technical work, 50

Taylor’s management principles, 47

John C. Stennis Space Center of DoD, 166
Johns Hopkins Medical Center
  ICU, 78

K
knowledge work
  definition of, 3–4
  management challenges, 3–4, 13
  methods in this book for, 7–8
  nature of, 30–31, 68
  Quarksoft management of, 23
  urgency of change and, 6–7
knowledge work, importance
  as executive-level issue, 29–30
  future of, 42–43
  nature of, 30–31
  references, 44
  summary review, 43–44
  team accomplishments, 40–42
  team example, 35–40
  as troublesome, 31–32
  why customers tolerate shoddy
    software work, 32–34
  why software problems persist, 34–35
knowledge work, large scale. See also extending TSP use; large-scale programs using TSP
  nature of modern technical work, 50
  troublesome nature of, 31–32
  urgency of change and, 6–7
knowledge work, managing
  blame culture, 56–57
  modern technical work, 49–50
  modern technical workers, 50–51
  modern technical workplace, 48–49
  need for trust, 57–58
  overview of, 45
  principles of, 51–53
  quality as top priority, 60–61
  references, 62–63
  summary review, 61–62
  Taylor’s management principles, 46–48
  team leadership and support, 61
  trusting workers, 53–56
  trustworthy teams, 58–59
  using facts and data, 59–60
knowledge workers, motivating
  cost, schedule and quality plans, 72–75
  establishing trust, 54–56, 69–70
  knowledge-working culture, 68–69
  management vs. worker
    objectives, 65–66
  nature of team motivation, 66–67
  operational processes, 76–79
  overcoming skepticism, 85–86
  recording data, 75–76
  references, 87
  required skills, 71–72
  self-management tasks, 71–72
  self-management training, 84–85
  start-up problems, 70–71
  summary review, 86–87
  tracking and reporting progress, 79–84
knowledge-working teams. See also teams, TSP
  creativity of modern, 51
  current unsatisfactory
    performance of, 31–34
  example, 35–40
  example, accomplishments of, 40–42
knowledge-working teams (continued)

handling poor performers on, 109–111

as most important single asset, 44

principles of managing, 51–53

trustworthy, 58–59

knowledge-working teams, motivating. See also teams, TSP

Beckman Coulter, 89–90

Beckman Coulter’s first TSP team, 90–92

building self-directed teams, 97–98

management behavior, 95–97

management issues, 98–100

management style, 100–103

organizational goal of, 152

overview of, 89

summary review, 104–105

team commitment, 92–94

Kuehler, Jack, 290–291

L

Lancaster, Tim

gaining trust of management, 95–97

leading Beckman Coulter’s first TSP team, 90–92

motivating knowledge-working teams, 92–94

quality management and, 141–142

Landmarks of Tomorrow (Drucker), 3

languages, TSP independence from, 209

large-scale programs using TSP development organizations, 250–251

overview of, 249

program management, 251–252

role-manager teams, 253–255

system architecture, 250

team coordination, 252–253

launch. See TSP launch

leadership

building internal capability in rollout plan, 236

coaching. See coaching as critical success factor, 184–185

developing and identifying, 103, 105

evaluating team plan, 222–223

execution, 157–158

extending TSP use, 244

goals, 147–152

identifying, 243

importance of, 145–146

managing knowledge work, 53, 61

motivation and, 155–156

pilot project selection criteria, 206

project review issues, 229

quality transformation and, 140

recognition, 154–155

references, 160

standards of excellence, 156–157

summary review, 158–159

team example, 39–40

training, 152–153, 212–214

TSP implementation costs, 169–170
Leading Development Teams

course, 213

lean and mean organizations,
118–119

lighthouse example,
communication hierarchy,
120–121

LOC (lines of code)
applications developed with
TSP, 167
evaluating pilot projects, 204

local sponsorship, extending TSP
use through, 237–238

long-pole status report
conducting program reviews, 281
monitoring program status
example, 267–269

loyalty, as team accomplishment, 41

M

maintaining sense of urgency,
defining your standards of
excellence, 157

maintenance, as quality-cost
element, 299

Malcolm Baldrige National
Quality Award Board of
Examiners, 308

managed-services delivery model,
Softtek, 10

management. See also knowledge
work, managing
establishing objectives for TSP,
177–179

motivating, 155–156, 198
role in getting started. See
going to get started with TSP,
management’s role

software quality problem,
128–129

TSP introduction strategy, 190
TSP self-management style,
179–180

management, teamworking
development
building self-directed teams,
97–98
developing confidence of,
96–97
developing trust of, 95

issues with TSP adoption,
98–100

manager problems vs., 59
proof of motivation, 95–96
Quarksoft system of, 20–23

Managing at the Speed of Change
(Conner), 191–192

mandating action, executive role
in, 302–303

Marshall, Rick, 90, 100

medical devices
safety quality of, 127–128
software products built with
TSP, 167–168

medical procedures in hospitals,
example of operational
processes, 78

mental tasks (MT), performance
vs. incentive, 55–56

mentor coaches, 243–244

Microsoft
results using TSP, 164–165
testing Windows OS, 134

training of temporary personnel
in India, 234

middle-management, motivating,
198
INDEX

MIT, 54–55
Model T Ford, 45
monitoring program status, 258, 266–270
motivation
   accelerated projects requiring, 152
   building, 155–156
   of knowledge workers. See
      knowledge workers, motivating
   of knowledge-working team. See
      knowledge-working teams, motivating
management, 155–156
management behavior changes
   after seeing team, 95–96
of modern technical workers,
   50–51
for participation in TSP
   implementation team, 198
performance vs. incentive,
   54–56
self-management training
   objective, 85
MT (mental tasks), performance
   vs. incentive, 55–56
multidisciplined teams
   development organizations and,
      250–251
   in large-scale programs, 249
   pilotig TSP on large, 208
program management,
   251–252
role-manager teams, 253–255
system architecture, 250
team coordination, 252–253
using TSP, 247
multi-site vs. single-site projects,
   208

N
NAPCS (North American Product
   Classification System),
      167–168
NASA goals example of faster,
   better, cheaper, 147–149
National Medal of Technology, 307
nature of knowledge work
   knowledge-working culture
     and, 68
   overview of, 30–31
Naval Air Systems Command
   facilities of DoD, using TSP,
      166
NAVO (Naval Oceanographic
   Office) of DoD, examples
   building functional team
      motivation, 248–249
   building team motivation, 260
   reasons for using TSP, 166
Networked Systems Survivability
   Program (CERT), 165
North American Product
   Classification System
   (NAPCS), 167–168
Northern Illinois University, 309
nuclear power plants, safety quality
   of, 127–128

O
objectives
   building coaching capability,
      241
   management vs. knowledge worker, 65–66
   overall rollout plan, 232–235
PSP course, self-management training, 84–85
Softtek’s TSP implementation,
      166
team-based discipline in
team building, 21
that TSP will not help you
achieve, 178
open communications
credibility and, 94
requirements for tracking and
reporting progress, 80
open-source programming,
motivation for, 54
operating systems, TSP
independence from, 209
operational processes
benefits of, 78–79
bureaucracy handling details of,
15–16
discipline needed to follow, 84
guiding people, 77–78
operator error, testing problem,
133
Oracle, results using TSP, 164–165
organization structural analysis, 118
organizational considerations, TSP
implementation, 179
organizational plan, 232
organizational processes, 76–77
organizations, TSP certification
for, 304
Over, James W., 308–309
overall implementation strategy, 232
overall rollout plan. See rollout plan

P
pace of TSP implementation, 174
participation, team-member,
102–103
Partner Network, SEI, 164
partners
overcoming resistance to TSP, 235

people
communicating with data,
120–122
employee satisfaction analysis,
116–117
not using personal data in
evaluating, 108–109
operational processes guiding,
77–78
Quarksoft development of, 23
Quarksoft evaluation and
motivation of, 26
Softtek, 11
staff support analysis, 118–119
workforce one hundred years
ago, 47–48
performance
benefits of TSP implementation,
177–178
coaching evaluation, 243–244
focus on process compliance vs.,
77
handling poor performers,
109–111
length of time to implement
TSP and, 175
of modern technical workers, 51
motivation and. See motivation
pilot project evaluation,
203–204
program reviews of quality,
280–281
Quarksoft management style,
19–20
Softtek experience, 9–11
vs. incentive, 54–56
period status report, 226
personal data
management of, 108–109
not using to micromanage teams, 178–179

Personal Software Process. See PSP (Personal Software Process) course

physical tasks (PT), performance vs. incentive, 55–56

pilot projects
communicating purpose of, 199
coloring, 202
early success as critical to, 184, 188–190
evaluating, 202–205
launch meeting 1 considering purpose of, 219–220
length of time to implement, 174–175
overview of, 201–202
selecting participants, 210–211
selection criteria for, 206–210
planned value (PV), calculating EV of project status, 266–267

planning manager
evaluating team plan, 222–223
launch management review meeting, 222–223
maintaining program-wide communication, 284
managing quality, 141–142
role of, 253
role-manager team for, 254
TSP weekly meeting topics, 217

planning process
coaching team, 199–201
communicating idea, 198–199
launching TSP implementation team, 195–197
local implementation plan, 239–240
management’s role during launch. See TSP launch, management’s role in motivating TSP implementation, 198
program plans. See program plans, aggressive but realistic quality plans. See quality plans reviewing implementation plan, 197
selecting participants, 210–211
selecting project teams, 205–210
TSP pilot programs, 201–205
platforms, TSP independence from, 209
postmortems
assessing plan feasibility for cycle, 224–225
calculating cost of implementation, 170, 172
coloring, 218
management reports on, 226
TSP launch process, 215
TSP weekly meeting topics, 217
preparation phase, Conner’s change model, 192–193
prevention, crises
ensuring effective program-wide communication, 284
identifying and resolving issues, 270
prevention costs, COQ, 126–127
principles
of creative destruction, 12
Taylor’s management, 46–48
principles, managing knowledge work and workers
management style adhering to, 100–103
overview of, 51–53
quality as top priority, 60–61
relying on facts and data for decisions, 59–60
summary review, 62
team leadership and support, 61
trusting knowledge workers, 53–58
trustworthy knowledge-working teams, 58–59
priority
coaching capability as, 241
continuous improvement as, 295, 301–302
quality as top, 125
rollout plan for maintaining, 236–237
privacy, management respect for team member, 227
PROBE (PROxy Based Estimating) method, for product size, 74
problems
building trust with knowledge workers, 58–59
identifying and resolving issues with early warning, 270
problems, future
as different and more challenging, 12
failure of leadership to recognize, 5–6
urgency of change and, 6–7
procedures, bureaucratic
delay and inefficiency from, 15
Quarksoft management practices eliminating, 21
software crisis and, 16–18
summary review of, 27–28
process fidelity, capability requirements, 245
process improvement, 182–183
process managers
launch management review meeting, 222
role of, 253
TSP weekly meeting topics, 217
product managers, 215, 302
product quality. See quality
productivity
gains of TSP implementation, 170, 177–178
managing with facts and data, 124
Softtek focus on, 9
Taylor’s management principles for, 46–48
TSP vs. non-TSP teams, 172–174
productivity analysis, using TSP data, 112–113
program management using TSP
continuing responsibilities of, 283–285
dealing with customers, 281–283
establishing aggressive but realistic plans, 259–265
identifying and resolving issues, 270–275
in large-scale programs, 251–252
managing quality, 275–281
monitoring program status, 266–270
overview of, 257–258
program management problem, 258–259
program management using TSP

(continued)
references, 286
summary review, 285–286

program managers
building local sponsorship from, 238
focus on quality, 283–284
identifying, 243
leading program management team, 252
maintaining program-wide communication, 284–285
project team from two companies agreeing to, 40

program plans, aggressive but realistic
assessing plan, 261–263
balancing resources and schedules, 264–265
building team motivation, 260–261
defined, 258
overview of, 259–260
product size estimate, 263
program management problems, 281–283
quality plan, 264
resource estimate, 263–264
reviewing risks, 265
program reviews, of quality performance, 280–281
program status, monitoring defined, 258
earned value measure and, 266–267
earned value measures, limitations of, 269–270
example status report, 267–269
overview of, 266
programming
future of knowledge work and, 43
lack of vision at IBM, 5
motivation for open-source, 54
programs, SEI, 165
progress reviews. See also project status review, 237
project launch. See TSP launch
project managers
chain of sponsorship for, 210
cost of TSP implementation, 169–170
pilot project selection criteria, 206
project review issues, 227–229
role of, 181–182
training, 212–214
TSP launch process, 215–218
for TSPm multi-teams, 251–252
with very-dissatisfied customer ratings, 116
project ownership, motivating teams, 93–94
project postmortem report, 226
project status review
describing, 82–83
establishing credible reports on, 94
measuring, 81–82
overview of, 79–80
report formats for, 226
reports, 83–84
review issues, 227–229
rollout plan for priority, 237
tracking, 80
TSP launch process, 216–217
project-by-project implementation, 174
project-level EBITA tracking, Quarksoft, 24–25
projects. See also pilot projects
Pronovost, Dr. Peter, 78
PROxy Based Estimating (PROBE) method, for product size, 74
PSP (Personal Software Process) course
certification, 303
courses, 213–214
defect rates of experienced developers, 129–130
describing project status, 82–83
Humphrey’s leadership in developing, 307
identifying initial TSP coaches, 200
making cost, schedule and quality plans, 73–75
measuring project status, 82
objectives, self-management training, 84–85
overcoming external resistance to TSP, 234
overcoming skepticism, 85–86
self-management tasks, 71–72
TSP launch requiring instructors in, 244
PSP: A Self-Improvement Process for Software Engineers (Humphrey), 245
PSP Advanced course, 213
PSP Fundamentals course, 213
PT (physical tasks), performance vs. incentive, 55–56
PV (planned value), calculating EV of project status, 266–267

Q
QA (quality assurance), TSP implementation, 182–183
quality
accelerated project example of, 151
Beckman Coulter example of, 141–142
cost-of-quality measure, 126–127
economics of software, 136–139
executive review of, 298–301
gains when implementing TSP, 170, 178
management recognition of, 154–155, 228
managing program, 259, 276–280
priority, 23, 53, 60–61, 125, 283–284
problem of software, 32, 128–132
problem of testing for, 132–136
program reviews of, 280–281
project review issues, 228–229
references, 143
safety-critical systems and, 127–128
standards of excellence and, 156–157
summary review, 142–143
transforming product, 139–141
quality manager
evaluating team plan, 223
launch management review meeting and, 223
role of, 141–142, 253
TSP weekly meeting topics, 217
quality plans
assessing aggressive but realistic program plan, 264
quality plans (continued)
  making, 74–75, 84, 276
  management’s role in TSP
  process, 223
  presenting in TSP launch
  process, 215, 221
  program reviews of, 280
  reviewing, 264
Quarksoft example
  executive team, 23–24
  financial management, 24–25,
  295–296
  leadership goal, 147
  management system, 20–23
  profitability, financial perfor-
  mance analysis, 117–118
  R&D improvement, 25–26
  reasons for using TSP business
  model, 166
  story, 18–20
Qureshi, Humayun
  Beckman Coulter’s first TSP
  team, 90–91
  team commitment and, 92–94
  team gaining trust of
  management, 95–97
R&D, Quarksoft executive team,
  23–26
Reagan, President Ronald, 69
recognizing quality work
  by leadership, 154–155
  of modern technical workers,
  51–52
  sustaining TSP by, 305
recording data
  management rewarding teams
  for, 228
  for self-management, 75–76
self-management training in, 84
  TSP status reporting scripts for,
  226
relaunch
  calculating cost of
  implementation, 172
  quality management, 141–142
  team accomplishments, 41
  team example, 38–39
  removal rates, quality plans and, 75
reorganization problems, and
  adoption of TSP, 99–100
reporting
  establishing credible status, 94
  project status, 83–84
  project status, formats for, 226
  required skills, knowledge workers
  and self-management tasks, 71–72
requirements for open
  communications, tracking and
  reporting progress, 80
Research, Technology, and
  Systems Solutions Program
  (RTSS), 165
resistance to TSP, overcoming,
  233–235
resource contention problem,
  testing, 133
resources
  balancing schedules and,
  264–265
  estimating in aggressive
  program plan, 263–264
  TSP introduction strategy, 190
return on investment (ROI)
  analysis, 162, 171–174
review
  pilot program, 202
  TSP implementation plan, 197
review rate measure, quality, 276–277
review yield measure, quality, 276–277
rewards, worker. See also knowledge workers, motivating modern technical workers, 50–51 performance vs., 54–56 for quality work, 305 trusting knowledge workers, 54–55
Rickover, Admiral, 57–58
risk mitigation in aggressive program plans, 265 in TSP implementation plans, 197 when making improvements, 291–293
River Rouge production facilities, 45
roadmap, to appendices in this book, 162–164
Roger’s change model, 191–194
ROI (return on investment) analysis, 162, 171–174 role-manager teams, 253–255, 284–285
RTSS (Research, Technology, and Systems Solutions Program), 165
rules, bureaucratic, 15
S safety-critical systems, 127–128
sales effort, and corporate churn, 2
scale, 130 costs of implementing TSP, 169–170 development work on a large. See program management using TSP future problems concerning, 6 testing on progressively larger, 130
schedule aggressive program plans and, 263 balancing resources to achieve, 264–265 objectives TSP will not help you achieve, 178 pilot project evaluation, 204 plan for, 72–73 Quarksoft example, 20, 22–25 software crisis and failure of, 7, 16 TSP implementation plan review, 197 as TSP performance benefit, 177 TSP workload balancing and, 11
Schumpeter, Joseph, 1–2, 12
scientific management Taylor’s principles of, 46–48 work of Frederick Taylor, 45
scope, TSP implementation dependent on, 174
SEI (Software Engineering Institute) about James Over, 308 about Watts Humphrey, 307 definition, 165 establishing benchmarks, 114 evaluation and certification services, 165
SEI (Software Engineering Institute) (continued)
knowledge-working team example, 35–36
organizations using TSP belonging to, 164
programs of, 165
project launch, 36–37
Softtek story, 8–9
self-discipline, self-management training, 84
self-management
Beckman Coulter’s first TSP team, 90–92
building self-directed teams, 97–98
impact on TSP project management, 181–182
making cost, schedule and quality plans, 72–75
overview of, 71–72
PSP course objectives, 84
recording data, 75–76
start-up problems, 70–71
tracking and reporting progress, 79–84
training, 84–85
TSP implementing concept of, 179–180
using operational processes, 76–79
senior managers. See executive team
SEPM (Software Engineering Process Management Program), 165
service orientation, of modern economies, 43
shortcuts, pitfalls of, 77
simplistic measures, as improvement risk, 291–292
single-site vs. multi-site projects, 208
size
coaching team, 181
estimating product, using historical data, 263
estimating product, using PROBE for accuracy, 73–74
executive financial reviews of job, 296–297
managing business, 6, 12
pilot project team, 208
skepticism
overcoming in knowledge workers, 85–86
removing disruptive team members, 211
selecting team for pilot projects, 211
skills
building internal capability, 235–236
coaching team, 181
extending TSP use, 245
knowledge worker, 71–72
modern technical worker, 50–51
slavery, 54
Small Business Administration study, 2–3
small businesses, 2–3, 6
Softtek
addressing process fidelity problem, 245
leadership goal of, 147
people, 11
product offering, 10–11
project performance, 9–10
reasons for using TSP business model, 166
story of, 8–9
service orientation, of modern economies, 43
shortcuts, pitfalls of, 77
simplistic measures, as improvement risk, 291–292
self-management
Beckman Coulter’s first TSP team, 90–92
building self-directed teams, 97–98
impact on TSP project management, 181–182
making cost, schedule and quality plans, 72–75
overview of, 71–72
PSP course objectives, 84
recording data, 75–76
start-up problems, 70–71
tracking and reporting progress, 79–84
training, 84–85
TSP implementing concept of, 179–180
using operational processes, 76–79
senior managers. See executive team
SEPM (Software Engineering Process Management Program), 165
service orientation, of modern economies, 43
shortcuts, pitfalls of, 77
simplistic measures, as improvement risk, 291–292
single-site vs. multi-site projects, 208
size
coaching team, 181
estimating product, using historical data, 263
estimating product, using PROBE for accuracy, 73–74
executive financial reviews of job, 296–297
managing business, 6, 12
pilot project team, 208
skepticism
overcoming in knowledge workers, 85–86
removing disruptive team members, 211
selecting team for pilot projects, 211
skills
building internal capability, 235–236
coaching team, 181
extending TSP use, 245
knowledge worker, 71–72
modern technical worker, 50–51
slavery, 54
Small Business Administration study, 2–3
small businesses, 2–3, 6
Softtek
addressing process fidelity problem, 245
leadership goal of, 147
people, 11
product offering, 10–11
project performance, 9–10
reasons for using TSP business model, 166
story of, 8–9
software
  crisis, 16–18
  as first large-scale knowledge work, 31
functional teams for maintenance of, 248
quality. See quality training developers, 212–214
TSP applications, 167–168
why knowledge work is troublesome, 31–32
Software Capability Evaluation, SEI, 307
Software Engineering Institute. See SEI (Software Engineering Institute)
Software Engineering Process Management Program (SEPM), 165
Software Maintenance Organization of DoD, 166
Software Process Assessment, SEI, 307
Software Process Program, SEI, 307
sponsorship
  building executive, 194
  building local, 237–238
  selecting pilot project participants, 210
staff support analysis, 118–119
standards of excellence
  communicating idea of TSP, 198–199
  defining your, 156–157
TSP introduction strategy, 188
start-up
  motivating knowledge workers for, 70–71
TSP return on investment analysis, 171–173
status reports
  earned value measure and, 266–267
  earned value measure limitations, 269–270
  program-level example of, 267–269
projects. See project status review steps, TSP implementation overall rollout plan, 232–235
TSP introduction strategy, 188–190
stuck-accelerator problems, Toyota, 128
style, management, 100–101
supervision
  inhibiting modern technical workers, 51
  Taylor’s management principles of, 47–48
suppliers, overcoming resistance to TSP, 234
support
  coaching as, 152–153
  continuous improvement requiring, 294
  functional teams providing, 248–249
  house building in record-time example of, 151–152
  importance of, 152
  in overall implementation strategy, 232
  recognition as, 154–155
  training as, 152–153
support managers, 217, 253
surgical procedures in hospitals, example of operational processes, 78–79
sustaining TSP. See continuous improvement, sustaining
system architecture, large-scale programs, 250
system software, built with TSP, 167–168
system testing
as quality-cost element, 299
software quality economics, 137–139
in software quality transformation, 140

T
task-hour measure, productivity analysis, 113
tasks. See self-management
task-time, resolving issues experience, 272–273
managing, 274–275
measure, 272–273
Taylor, Frederick Winslow, 45
team-based discipline,
teambuilding objective, 21
teams, TSP
accomplishments, 40–42
certification, 304
coordination in large-scale programs, 252–253
establishing, 194–199
evaluating pilot projects, 202–204
example of, 35–40
forming, 188–189
identifying and resolving issues. See issues, identifying and resolving
in large-scale programs, 249–253
management’s treatment of, 227
motivating to produce aggressive plan, 260–261
Quarksoft commitment ownership, 21–22
Quarksoft management system for, 21
selecting members for pilot project, 189, 207, 210–211
Softtek work life experience of, 11
team plan evaluation, 221–223
training, 212–214
TSP launch process, 215–218
TSP support for, 168–169
test manager
managing quality, 141–142
role of, 253
TSP weekly meeting topics, 217
testing
conclusions on, 135
EV measure limitations, 269–270
failed improvement program example, 289–290
finding only fraction of defects, 130–131
functional teams for, 248
in modern technical workplace, 49
problem of, 131–135
project status, 81–82
software quality economics and, 137–139
team example, 39–40
TSP implementation and, 182–183
Xerox defect-removal times, 139–140
Texas Instruments (TI), 5
The 7 Habits of Highly Effective People (Covey), 120–121
The Age of Discontinuity (Drucker), 3–4
The Principles of Scientific Management (Taylor), 45–48
Three Mile Island crisis, 58
Threehundred360-degree rating, 26, 27
time. See also task-time, resolving issues
evaluating pilot projects, 204
reducing with multidisciplined teams, 247
required for implementation, 174–176
timeline
assessing plan feasibility, 224–225
launching TSP implementation team, 196–197
tool support analysis, 119
tools
executing TSP launch, 216
TSP requirements, 210
top-down management
implementation strategy, 232
not effective in modern technical workplace, 49
Taylor’s management principles, 46
training proceeding from, 212
Toyota
creative destruction and, 33
results of shoddy work, 33–34
stuck-accelerator problems, 128
tracking
project progress, 79–84
project-level EBITA, 24–25
recording data on personal work, 76
training coaches, 199–201, 243
as element of leadership, 152–153
managers, 99–100
participants in pilot projects, 210
software development through, 152–153
Taylor’s management principles, 46
TSP implementation and, 211–214
TSP implementation costs, 169–170
TSP implementation plan review, 197
TSP implementation time, 175
training, self-management. See also self-management
convincing teams through, 59
objectives, 84–85
PSP objectives, 84–85
start-up problem, 70–71
transportation, safety quality of, 127–128
Treviño, Blanca, 8–11
trust
basing decisions on fact and data, 59–60
blame culture and, 56–57
elements of, 69–70
identifying and resolving issues. See issues, identifying and resolving
of knowledge workers. See also self-management, 53–56, 68–69
trust (continued)
management’s behavior after establishing, 94
managing customer relations program, 259, 281–283
need for, 57–58
of teams, 58–59, 261
TSP (Team Software Process)
Beckman Coulter’s first team, 90–92
companies using, 164–167
Humphrey’s leadership in developing, 307
knowledge-working team, 35–40
management issues, 98–100
motivating teams, 92–94
PSP training in, 71–72
Quarksoft executive team, 23–26
Quarksoft management, 20–23
Quarksoft story, 18–19
reporting project status, 83–84
for self-management, 59, 69
Softtek experience, 9–11
start-up problem, 70–71
team accomplishments, 40–41
TSP launch requiring instructors in, 244
TSP and your organization applications developed with,
167–168
benefits of, 177–179
characteristics of project teams, 168–169
coaching team, 180–181
cost of implementation, 169–170
critical success factors and barriers, 184–185
getting started, 176–177
management style, 179–180
organizational considerations, 179
overview of, 162–163
process improvement, QA, and test group impact, 182–183
project management impact, 181–182
references, 185
return on investment, 171–174
roadmap to appendices, 162–164
summary review, 185
time required for implementation, 174–176
who is using TSP, 164–167
TSP: Coaching Development Teams (Humphrey), 247, 249
TSP data
auditing, 111–112
overview of, 107–108
personal, 108–109
TSP data, using
benchmark analysis, 113–114
coaching support analysis, 114–115
cost-of-quality analysis, 115–116
customer satisfaction analysis, 116
employee satisfaction analysis, 116–117
facilities analysis, 117
financial performance analysis, 117–118
organization structural analysis, 118
overview of, 112
productivity analysis, 112–113
staff support analysis, 118–119
tool support analysis, 119
TSP Executive Strategy seminar
building sponsorship, 237
defined, 213
developing local implementation plan, 239–240
TSP implementation steps, TSP
introduction strategy, 188–190
TSP introduction strategy, 188–190
TSP launch
commitment ownership, 21
implementation team, 195–197
and length of time to implement
TSP, 174–176
local implementation plan,
239–240
process, 214–218
program plans. See program plans,
aggressive but realistic
team accomplishments, 41
team example, 36–39
TSP launch, management’s role in
assessing plan feasibility,
224–225
evaluation of team plan,
221–223
launch meeting 1, 219–220
launch meeting 9, plan
presentation, 220–221
launch meetings 2–8, 220–222
overview of, 218
reporting, 226
status review issues, 227–229
TSP: Leading a Development Team
(Humphrey), 245
TSP Team Member Training
course, 213
TSP: Coaching Development Teams
(Humphrey), 240
TSPF (TSP for Functional Teams),
207
TSPf (TSP functional) process, 249
TSPm (TSP multi-team) process
defined, 247
development organizations,
250–251
in large-scale programs,
249–253
program management,
251–252
role-manager teams, 253–255
system architecture, 250
team coordination, 252–253
TSP-OEC (TSP Organization
Evaluation and Certification)
process, 234
Twain, Mark, 108
U
unit testing, software quality
economics, 137–139
University of Chicago, 54–55, 307
unskilled laborers, Taylor
improving productivity of, 47
urgency
of change, 4–8
defining standards of excellence
with sense of, 157
U.S. Army, program management
text example of strained customer
relations, 281–282
U.S. Navy’s Submarine Service,
57–58
V
Vancouver Winter Olympics,
2010, 154
Vazquez, Carlos Montes de Oca,
23–26, 118
Vazquez, Cesar Montes de Oca
financial performance analysis,
117–118, 296
leadership goal of, 147
Quarksoft executive team, 23–24
Quarksoft story, 18–19
Vicarious Visions, 166, 247
warranties, quality
  quality measure of TSP impacting cost of, 178
  Softtek offering, 10
  software game industry and, 166
  software organizations rarely providing, 32
  Toyota example, replacing GM as worldwide leader, 33
  weapons systems, safety quality of, 127–128
weekly meeting topics, TSP, 217–218
Whitford, Larry, 90–94, 100
work. See also knowledge work
  modern technical, 49–50
  one hundred years ago, 46–48, 49
  recording data to track personal, 75–76
worker-management animosity, factory workers, 48
workers. See also knowledge workers, motivating autonomy of, 52
  blame culture and, 56–57
  continuous improvement benefiting, 293–294
  modern technical, 50–51
  need for trust management, 57–58
  one hundred years ago, 46–48
  principles of managing knowledge, 51–53
  rewards. See rewards, worker
  Softtek experience, 11–12
workforce
  employee satisfaction analysis, 116–117
  knowledge work as essence of modern, 145–146
  not using personal data in evaluating, 108–109
  one hundred years ago, 46–48
  operational processes guiding, 77–78
  Quarksoft example, 23, 26
  Softtek example, 11–12
  workload balancing benefits of TSP implementation, 11
  through teamwork, 151
workplace
  distributed teams in modern, 248
  facilities analysis of, 117
  identifying and resolving issues in, 270–275
  modern technical, 48–49
  during Taylor’s day, 47–48
  world record time, accelerating project in, 149–152
Wyrwa, Carl, 90, 100

X
Xerox
  defect-removal times, 139–140
  quality data, 136–137

Y
yield before test measure, quality, 276–277
Yu-na, Kim, 154

Z
Zen and the Art of Motorcycle Maintenance (Pirsig), 154
Ziemer, Jason, 248–249, 260, xxii