Reflections on Management

How to Manage Your Software Projects, Your Teams, Your Boss, and Yourself

Watts S. Humphrey
with William R. Thomas

Carnegie Mellon
Software Engineering Institute

The SEI Series in Software Engineering

This publication incorporates portions of "Watts New" by Watts S. Humphrey, © 1997-2007, The Personal Software Process (PSP) (CMU/SEI-2000-TR-022) by Watts S. Humphrey, © 2000, and The Team Software Process (TSP) (CMU/SEI-2000-TR-023) by Watts S. Humphrey, © 2000, Carnegie Mellon University, with special permission from its Software Engineering Institute.

Excerpts from the following books are reproduced by permission of Pearson Education, Inc.: *A Discipline for Software Engineering; Introduction to the Personal Software Process℠; Managing Technical People: Innovation, Teamwork, and the Software Process; Introduction to the Team Software Process℠; Winning with Software: An Executive Strategy; PSP℠: A Self-Improvement Process for Software Engineers; TSP℠: Coaching Development Teams;* and *TSP℠: Leading a Development Team.*

Library of Congress Cataloging-in-Publication Data

Humphrey, Watts S., 1927-
Reflections on management : how to manage your software projects, your teams, your boss, and yourself / Watts S. Humphrey with William R. Thomas.
p. cm. — (The SEI series in software engineering)
Includes bibliographical references and index.
ISBN 978-0-321-71153-3 (pbk. : alk. paper)
1. Computer software—Development—Management. 2. Process control. I. Thomas, William R., 1962- II. Title.
QA76.76.D47H863 2010
005.1068—dc22

2010005103

ISBN-13: 978-0-321-71153-3
ISBN-10: 0-321-71153-X
Text printed in the United States on recycled paper at Courier in Stoughton, Massachusetts.
First printing, March 2010

Preface

When projects go badly, our reaction is often to work harder—by which we mean work longer hours. But it's rarely that simple. Projects often go wrong at the very start, and their problems are generally symptoms of a deeply dysfunctional organization.

In a career spanning more than 60 years as a senior manager and researcher, Watts Humphrey has personally helped dozens of organizations go "from the brink of chaos to a sound, businesslike operation," as he wrote in his 2002 book *Winning with Software*. That description applied to Watts's experience with IBM, where he worked for 27 years, supervising 4,000 software professionals in 15 laboratories and 7 countries.

Later, as a senior fellow overseeing the process program at Carnegie Mellon University's Software Engineering Institute (SEI), Watts made an "outrageous commitment"—his words—to transform the world of software. Beginning in 1986, he pioneered the Capability Maturity Model (CMM), the Personal Software Process (PSP), and the Team Software Process (TSP). Those methodologies have helped thousands more organizations and engineers establish and, most importantly, commit to following effective engineering and management practices for their software projects.

Watts did not stop at describing methods for improving software engineering processes. Rather, he made it his personal

responsibility to instruct "all software professionals and their managers to plan and track their work, use the best technical methods, and measure and manage the quality of this work." In addition to teaching courses and presenting at conferences, Watts invoked the power of the pen, authoring 11 books and hundreds of technical reports, journal articles, and columns.

In 2005, at a White House ceremony, Watts was awarded the United States National Medal of Technology by the President of the United States "for his vision of a discipline for software engineering, for his work toward meeting that vision, and for the resultant impact on the U.S. government, industry, and academic communities."

Much of Watts's writing focuses on detailed descriptions of the tools of process management. But an equal amount is a remarkably clear presentation of his vision for properly planned and committed work. He writes in a straightforward and personal style. He draws on anecdotes from his years at IBM and the SEI but also from his earlier experience on the Auburn University wrestling team, for example, and from his service in the U.S. military. While he often describes success, he also recounts times when he felt that he failed and how he learned to approach a problem differently the next time.

This book, drawn from Watts's books, articles, and columns, comprises a collection of advice, stories, and hard-earned wisdom, rather than specific instruction on how to implement the PSP or TSP (which are thoroughly covered in Watts's books on those specific subjects). What emerges for the reader is an understanding that successful software project management is a journey with many obstacles. To succeed, engineers must manage more than their projects. They must use their own experience and that of their teams to first understand and then plan the project ahead. They must influence their teams' attitudes

and methods for doing disciplined work. And they must persuade their bosses to set aside ill-informed notions of schedules and resource commitments and look instead at hard, historical data.

The essays in Part I provide insights on types of plans and the planning process. Part II covers team building and motivation. Part III describes how to work with your managers and persuade them to use best practices. And Part IV examines your personal responsibilities, commitments, and processes.

These essays shine a light on the challenges inherent in software development and can set engineers on the road to understanding how to succeed. And while Watts's particular expertise is software, practitioners in every field of business will benefit from the wisdom and advice contained here.

—Bill Thomas

Prologue

First and foremost, my thanks to Bill Thomas for all the work he did in putting this book together. He did a superb job of selecting topics and ordering the material so that it makes a cohesive whole. Even though I wrote all of the papers, reading them again brings back lots of memories of the wonderful experiences I have had in more than 60 years of professional work. In this time, I have been blessed with many opportunities and many wonderful associations. It has never ceased to amaze me how helpful people can be. Whether they are managers, peers, or subordinates, much of what I have learned has been due to the mentoring, advice, critiques, and even disagreements I have had over these years.

Second, I would like to comment briefly on where we are going. While what I have done has been exciting and rewarding, it is only a small step in the direction of the truly astounding changes coming in the not-too-distant future. Software has been hard to manage, because it is a new kind of work: large-scale knowledge work. Starting before the design of the ancient pyramids in Egypt, humans have been doing knowledge work, but on a small scale. While lots of people worked on these massive constructions, only a few of them were creative designers.

The first clues that large-scale creative work could be different were with the ancient cathedrals. While many people worked on

them, the overall architecture was designed by a very few people. However, there were hundreds of skilled artisans who also did creative work. They saw themselves as creating a cathedral for God, and they worked, not for some chief engineer or boss, but for the Almighty. These workers were volunteers, and they had an overall vision and motivation that was more than just doing a job. Of course they didn't manage to tight schedules or control costs, but they did manage themselves.

What makes software more like building cathedrals than traditional work is that it is large-scale creative work. Never before have dozens, hundreds, and even thousands of people tried to work together to produce a single massive creation. Now, with the advances being made in team and multi-team management, we are learning how to do large-scale knowledge work.

Once these methods are widely practiced, we will see an enormous flowering of creative engineering. Large and complex systems will be produced on predictable schedules and for planned costs. As soon as we can do this, the possibilities of what we can design and build will be greatly expanded. We will be able to do many of the things we have thus far only dreamed about.

When we have truly mastered large-scale knowledge work, we will be ready for some unprecedented international crisis like deflecting a rogue meteoroid or reengineering the earth's atmosphere. Assuming that we have the vision and technology, we will then have the management skills to actually bring off such a massive project and to do it on a predictable schedule. Hopefully, such international crises will not arise and, hopefully, there will be no need to escape to another world or to rebuild this one, but with these new knowledge-working methods, we should be able to do it.

Finally, I have dedicated this book to three marvelously skilled doctors. About a year ago I was told I had an inoperable cancer

of the liver and given three to six months to live. By a series of almost miraculous events, we found Dr. David Ryan at Mass General Hospital who introduced us to Dr. Theodore Hong, a radiologist who had invented a treatment specifically designed for my kind of cancer, and to Dr. David Forcioni, a gastroenterologist. Because of the care and skill of these three gentlemen, I completed the treatment and the latest reports show no sign of cancer. Dedicating this book to them is my way of saying thank you.

—Watts Humphrey
January 12, 2010

commitment next time. The estimates should be reviewed to see what was overlooked, and the contingencies should be revised to include the new experiences. By comparing actual performance with the estimates, engineers soon learn to make better estimates. This is why the people who will do the work should make their own plans: to learn how to consistently make commitments they can meet.

4.3 A GOAL IS SOMETHING YOU WANT TO ACHIEVE

The dictionary defines a goal as "the result or achievement toward which effort is directed."[2] Goals concern results and efforts, but most importantly they concern direction. Goals provide direction and focus for our efforts. They clearly define the end that we desire and establish a priority for the required work.

Goals also imply several other things. For example, you need to know whether you have achieved the desired result and where you are along the way. Are you winning or losing and are your efforts likely to be successful? All of these—the result, direction, measurement, and effort—are involved in setting and achieving goals.

Goals are useful for individuals. Few would argue that, without a goal, it is impossible to strive. Without some objective, all the effort seems pointless and a waste of time. After all, if the effort doesn't get you anywhere, why bother? Thus, a goal concerns a destination, and this destination must be some place or some state that you really would like to achieve. This could be losing weight, getting a higher score, or delivering a product, but the goal provides a concrete objective toward which to strive.

2. *Random House Dictionary of the English Language*. 1983. New York: Random House.

Another way to think about goals is in the negative. A key reason given when the presumed better competitor loses in boxing, track, or any other sports competition is that he or she did not want to win badly enough. Similarly, in building products, it is widely accepted that when people don't strive to build quality products, they generally won't. In fact, they really cannot. Challenging goals are not achieved by mistake. If you don't consciously strive for them, you almost certainly will not achieve them.

So, goals are not just an invention of management, they actually satisfy a fundamental human need. The goal defines our purpose: why we are here, why we are working, or what we intend to achieve. Simply put, without a goal, you cannot succeed and, if you cannot succeed, why try? Goals are the motivators for human endeavor. They energize our lives and our work. They give us purpose. Achieving a goal provides a sense of achievement and satisfaction. Goals are important to people and they are even more important for teams.

Teams need goals for all of the same reasons that individuals do. In addition, goals provide a common working framework for the team. The goal is something that everyone agrees on and can cooperatively work to achieve. The goal helps to resolve issues. Does this activity move the team toward the goal or would something else be more effective? If some action does not help to achieve the goal, why bother doing it? After achieving a goal, the team members have something to celebrate. It was hard work, but they brought it off. It was a team achievement and everyone shares in the celebration and in the credit.

Without a common goal on which all members agree, you have a loose collection of individuals who share only a common trait or facility; you cannot have a team. It would be hard to imagine an athletic team where the members did not all share a common goal, agree on precisely what that goal was, and know exactly what the score was at every point in the play. In addition,

most needed. When time is short, engineers should take special care to avoid mistakes. Unfortunately, experience shows that this is the very circumstance when engineers and their managers are least likely to allow the time to do reviews, inspections, or thorough testing.

Loss of trust. If you frequently miss commitments, people will notice. They will learn that when you commit to something, you often don't keep to your word. Such a reputation is hard to repair and will affect your grades, your job ratings, your pay, and even your job security.

Loss of respect for your judgment. When people do not trust what you say, they are unlikely to ask for your opinion and they are more likely to insist that you work to unreasonable schedules.

The most important single asset a software engineer can have is a reputation for meeting commitments. For people to trust your word, you need to say what you plan to do and then do what you say.

7.12 WHAT DO YOU WANT FROM LIFE?

What do you want from your life? This is a big question that many people have trouble answering. A few points are worth considering as you think about the answer.

One way to get satisfaction from a job is to have status or power. People can get this by being a boss or being put in charge of an important service. Power and status can also be indirect, like making a lot of money, working for an important company, or driving a fancy car. These are all parts of "being" someone.

While there is nothing wrong with status, it is temporary. You may hold an important job for a while but, sooner or later, your next step will be down. Losing status can be a crisis. Some people are devastated when they first lose an important job. It is easy to confuse the importance of a job with personal importance.

I have known managers who were crushed by a demotion. They had built an image of themselves as important people. As long as they held a big job, everybody treated them as important. The minute they lost that job, however, they were just like everyone else. Nobody cared what they said and they stopped getting special treatment. They had lost the corner office and no longer had a secretary. This can be such a severe shock that some people have nervous breakdowns, heart attacks, or family crises. Their reward was status and it is gone.

The need is to decide what it is that you want. Think ahead. When you ultimately retire, what would a satisfying life look like? I suggest that what you have done will be far more rewarding than what you have been. If, for example, you plan to do engineering work, you probably have the instincts of a builder. Maybe you will build systems or components. You could end up building methods or processes. Or you might have a scientific bent and build theories or do research to build fundamental knowledge.

Whatever you build, however, quality will be key. You will get little satisfaction from sloppy work. Somehow, even if no one else finds out, you will know you did a sloppy job. This will destroy your pride in the work and it will limit your satisfaction with life. You cannot honestly say to yourself that you really believe in quality, but you will just get by this one time. There are always lots of excuses. You might even satisfy others with an expedient answer, but you will never satisfy yourself.

When you do quality work, you will be proud. Even if no one else knows, you know you did a first-class job and you are satisfied that you did your best. The surprising thing is that quality work gets known. It may take a long time, but sooner or later quality work is recognized. Whether you know it, you will get credit for the quality of your work.

So ask yourself this question: "Do I want to feel proud of what I do?" Most people would answer yes. But if you really

mean it, you need to set personal standards and strive to meet them. When you meet these standards, raise them and strive again. Challenge yourself to do superior work and you will be surprised at what you can accomplish.

7.13 DEVOTE YOURSELF TO EXCELLENCE

As you look to the future, you will face many questions. How will your field evolve, and what can you do to meet the mounting challenges? While no one can know, your progress probably will be limited by your ability to build your personal skills. Make practice a part of every project and measure and observe your own work. You cannot stand still, so you should treat every project as a way to build talent rather than merely treating your talent as a way to build projects.

Deciding what you want from your chosen field is like asking what you want from life. Surprisingly often, people achieve their objectives, but in ways they did not expect. Life rarely turns out the way we plan. While our carefully developed strategies may go down in flames, a new and more rewarding opportunity shows up in the ashes. The key is to keep an open mind and to keep looking. In life, we all reach the same end, so we need to concentrate on the trip. Just as with a process, once you decide how you want to live, the rest will follow. Devote yourself to excellence, and you just might achieve it. That would be worth the trip.

SOURCES

7.1: From *PSP[SM]: A Self-Improvement Process for Software Engineers*, Chapter 1

7.2: From *The Watts New Collection: Columns by the SEI's Watts Humphrey*, Number 8 2007, "Being Your Own Boss—Part IV: Being a Victim"

7.3: From *Introduction to the Personal Software Process[SM]*, Chapter 1

As a team leader, you will not generally face the problems of organization-wide change. However, it is important to consider the common symptoms of poor leadership and to ensure that your leadership style does not create similar problems. Poor leadership has many symptoms, but it generally stems from a failure to see what is needed and to set a direction that takes advantage of the available resources and opportunities.

It is often difficult to be objective and to establish goals for what to do and how to do it, but the key is to realize that you do not need to do it all by yourself. The modern world is simply too complex and no one person is smart enough or has enough knowledge to figure out everything without assistance. While you likely must make many leadership decisions yourself, you should take advantage of the intelligence, ideas, and creative suggestions of your team.

There is ample evidence that the combined intelligence of a group produces better results than even the most skilled and talented individual.[2] So use your team. It needs leadership; it wants leadership; and it will gladly help you to provide that leadership.

8.4 LEADERSHIP MUST BE EARNED

Management uses resources to accomplish results; leadership motivates people to achieve objectives. Managing is impersonal and can be demeaning. It presumes that those being managed don't have ideas and feelings and must be told what to do and how to do it. Management is appropriate for handling inanimate objects or routine jobs. However, people like to be motivated to accomplish more challenging tasks, and they do not like being herded and directed as if they were so many cattle.

---

2. Watts S. Humphrey. 1997. *Managing Technical People: Innovation, Teamwork, and the Software Process.* Reading, MA: Addison-Wesley.

Most of us enjoy technical work, and we sought development careers because we like to do creative and challenging things. We also like to see the results of our labors, particularly when our products work the way we intended. But when someone treats us as if we were stupid or unthinking, we lose our energy and creative spark. As team leader, you will probably have to manage at least some routine work, but development engineering calls for leadership and for energetic and motivated teams. That is the only way to consistently produce truly superior results.

One principal distinction between leaders and managers is that managers direct people to obey their orders while leaders lead them. This crucial distinction is best illustrated by an example. One software manager, Ben, told me how he learned what leadership was all about. He was a marine lieutenant in Vietnam and, for the first time, he was leading his platoon into combat. As they approached the front lines, the captain told him, "Take that hill." "That hill" was where the enemy was dug in with a machine gun. There was no time for a discussion, so Ben told his troops, "Follow me," and he started running up the hill. He told me that all he could think of as he ran was not whether he would get shot or what would happen if he got to the top. The question that kept running through his head was, "Are they following?" It turned out that they were and they took the hill, but Ben told me that he learned right then that the two key ingredients of leadership are getting out front and trusting your troops to follow.

So leadership is intensely personal. It is not something that you can order and it is not something that you can measure, evaluate, and test. It is a property like loyalty or trust. It cannot be bought or inherited. It must be earned, and earned through long and often painful experience. It can, however, be lost in an instant. All you need to do is to stop behaving like a leader. Then your followers will stop following. They may continue to

obey you, but you will soon sense that you no longer have their loyalty and trust. You can only tell if you are a leader by what happens: you are leading and they are following their leader.

What sets leaders apart from everyone else is that they have followers, and what attracts followers is a challenging and rewarding goal. It is impossible to be an effective leader without being committed to a cause that animates you and motivates your followers. Your energy and drive then come from your personal commitment to accomplish this objective.

This can't be just any goal—it must be something that you feel strongly about and will strive to accomplish. You must be sufficiently committed to this goal so that you can exhort your troops to achieve it, in spite of all obstacles. While development projects can have this character, that is not always the case. But, as we shall see, it is usually possible to excite creative people about the challenges and rewards of producing something entirely new and original.

8.5 STRIVE TO BE A TRANSFORMATIONAL LEADER

How do you feel about the job you have to do? Are you excited about it and dying to be part of creating this marvelous new product? If you view the job as just another chore, you have little chance of building the team's excitement to the feverish pitch required for great work. Excitement is contagious, but so are boredom and laziness. As a leader you not only set the team's pace, but you also establish the attitude. If you want this team to win, they must act like winners. And for them to act like winners, you must act like a winner and also treat them as winners. It all starts with you.

Think about your job and what you can do to make it an exciting project where people will want to work. If you wake up in the middle of the night with ideas on how to attack a major

Index

A

Accessibility, requirements plans meeting, 30 –31
Accuracy, requirements plans meeting, 32
Activities, categorizing in time management, 198
Administrative support, 195
Agreement
  checking for agreement as involvement technique, 113–114
  as element of commitment, 102–103, 205–206
Airline flight crews, 84–86
Andrews, Frank, 192
Attentive listening (Covey), 57
Attitude, responsibility and, 199–202
Autocratic bosses
  identifying autocratic environments, 153–155
  negative impact on motivation and performance, 150–152
  reasons for autocratic behavior, 152–153

B

Being responsible, 199–202
Benchmarks for performance, 128–129
Benefits of process improvement
  measuring, 168–169
  overview of, 166
Blame, failure and, 185
Booch, Grady, 229
Brooks, Fred, 235
Bugs and defects, 10–11
Building teams, 88–89
Bureaucratic momentum, 218
Bureaucrats, 151–152
Business environment, tailoring project proposal to, 162

C

Capability Maturity Model (CMM), 156, 170, 230, 240–241
Capability Maturity Model Integration (CMMI), 240–241
Categorizing activities in time management, 198

Change
  assessing impact of changes on existing plans, 144–145
  improvement based on trying something new, 186
  perpetual turmoil as quality of poor leadership, 219
  reasons for, 156–157
Clarity, requirements plans meeting, 31
Closed group, 68–69
CMM (Capability Maturity Model), 156, 170, 230, 240–241
CMMI (Capability Maturity Model Integration), 240–241
Coaching. *See also* Leading and coaching teams
  playing dumb as means of encouraging involvement, 112–113
  power of, 109–110
  team leaders, 119–120
Cockpit flight crews, 84–86
Code inspection, 173
Cohesion
  qualities needed by effective teams, 52
  between team members in self-directed teams, 71
Combat groups, 63 –65
Commitment
  analyzing before agreeing, 206
  changing commitment system in an organization, 175–176
  communicating when unable to meet, 207
  documenting, 207
  in jelled teams, 51 –52
  making and sustaining, 104–105
  making changes based on agreement, 141–142
  management of, 207–209
  as a motivator, 102–104
  nature of, 204–206
  planning and, 24, 29, 124, 139–143, 206–207
  properties of self-directed teams, 71, 72
  by team members, 79–81
  by teams, 40–41, 103–104
  trust and, 79
Communication
  experts inhibiting team communication, 117
  skills needed by effective teams, 56–58
  when unable to meet commitment, 207
Complaint, victimization and, 185
Completion dates, committing to, 24
Compromise, teams and, 44
Computer History Museum, 229
Concerns, sensitivity to, 114–115
Confidence, lack of, 45
Consequences of defects, 10–11
Constantine, Larry, 65
Continuing costs in process improvement projects, 166
Contributions by team members, 84–86
Cooperation
  dealing with uncooperative team member, 93
  failure to cooperate in teams, 44
  in self-directed teams, 71
  standards for, 217

Costs
  continuing costs for process improvement project, 166
  of cutting support staff, 146
  of defects, 9
  estimating in scheduling, 27
  introduction costs for process improvement project, 164–166
  PSP and, 238
  training, 165
Covey, Stephen R., 57, 203
Credibility
  agreement based on, 102–103
  commitments must be credible, 104
  managing commitments and, 209
  meeting commitments and, 80
Crises, autocratic decision making in, 152
Customers, effects of incompetent planning on, 28

D

Data/facts, focusing on as involvement technique, 120
Decision making
  autocratic style, 150–151
  by groups, 153
  leadership and, 219
  team involvement in, 126
Dedication to excellence, properties of self-directed teams, 71
Defects
  bugs contrasted with, 10–11
  in code, 4–5
  dangerous in critical systems, 3–4
  defined, 8–9
  managing, 7
  preventing, 13
  programmers and, 7–8
  PSP and, 238
  removal vs. prevention, 9
Delay is usually worst choice, 202–204
DeMarco, Tom, 41, 192
Deming, W. Edward, 239
Design, steps in quality process, 13
Developers. *See also* Software engineers
  effects of incompetent planning, 28
  scheduling and, 26–27
  wanting to work in team environment, 178
Disagreements, sensitivity to, 114–115
Discipline in self-directed teams, 73–74
Discovery process, team approach to, 83
Discussion
  preventing monopolization of, 115–117
  questions as means of getting involvement in, 111–112
Disruptive behavior, dealing with in team environment, 92
Documenting commitments, 207
"Don't-rock-the-boat," 218
Drucker, Peter, 229
Dyer, Jean L., 40
Dynamic planning, 33

E

Emotions
  emotional reinforcement as basis of autocratic style, 153
  reacting to problems and, 200

Empathic listening (Covey), 57–58
Employees, zero turnover in self-directed teams, 70
Errors
  impact on large-scale systems, 4
  people making, 9
Estimates. *See also* Planning
  adjusting and exploring alternatives, 136–137
  comparing actual performance with, 81
  costs, 27
  guessing, 125
Ethics of commitment, 79–81
Evaluation measures in reward-based motivation, 101
Excellence
  devoting yourself to, 211
  properties of self-directed team, 71
Executive priorities, 161, 163
Exhaustion strategy in negotiation, 92
Experts
  managing, 117–119
  playing dumb as means of encouraging involvement of others, 112–113
Extrinsic motivation leaderships, 223

F

Facts
  focusing on as involvement technique, 120
  supporting process improvement project, 166–167
Fagan, Michael, 173
Failure, blame and, 185
Fear
  combat groups and, 64–65
  as a motivator, 100–101
Feedback, qualities needed by effective teams, 53–54
Flight crews, 84–86
Forming phase, teams, 58–59
Function creep, 45

G

Gilb, Tom, 173
Goals
  benefits of, 215
  challenge of setting intermediate, 106
  challenging goals needed by effective teams, 52–53
  creating a sense of urgency with, 105–107
  defining, 14
  defining quality goals, 187
  followers attracted to leaders by goals, 222
  impossible goals causing team failure, 46, 49
  plans for meeting short-term, 128
  in self-directed teams, 72–73
  setting priorities, 14–16
  source materials for, 16
  team development over time and, 54–55
  team members setting, 81–83, 87–88
  teams committing to, 40–41
  tracking, 53–54
  translating long-term objectives into short-term, 105–107
Greed as a motivator, 101–102
Greene, Maurice, 184

Groups
closed group style, 68–69
combat groups, 63–65
decision making by, 153
open group style, 66–67
overview of, 61–62
process groups, 62–63
random group style, 67–68
synchronous group style, 69
work groups, 62
working styles, 65–66
Guessing, 125

H

Habits, autocratic decision making due to, 152
Hard negotiation strategy, 90
Help, team members asking for and giving, 94–95
Hot buttons, including management issues in project proposal, 163

I

Iacocca, Lee, 44, 215
Ideas, new, 83
Ignoring (Covey), 57
Improvement
designing process for self-improvement, 184–185
efforts. *See* Process improvement
improving quality of your work, 184–185
steps in, 187
Inspections
code, 173
steps in quality process, 12
Interruptions, managing time and, 189–190
Intrinsic motivation leadership, 223–224
Introduction costs in process improvement projects, 164–166
Involvement techniques
asking question, 111–112
checking for agreement, 113–114
coaching team leaders, 119–120
focusing on facts and data, 120
managing experts, 117–119
not allowing observers or outsiders, 120–123
overview of, 110–111
playing dumb, 112–113
preventing monopolization of discussion, 115–117
sensitivity to concerns or disagreements, 114–115

J

Jelled team
common understanding as first step in, 55
communication critical in, 56
definition of, 41
qualities of, 51–52
Jobs
getting satisfaction from, 209–210
job hopping in response to doomed project, 147
need for job security, 177
zero turnover in self-directed teams, 70
Journey, quality, 11

K

Katzenbach, Jon R., 87
Knowledge work/knowledge workers, 229, 234–235

L
Lack of confidence in team, 45
Launch process, TSP, 48, 240
Leadership. *See also* Leading and coaching teams
  behavior affecting team, 213–215
  from below, 225–227
  circumstances creating leaders, 224–225
  coaching team leaders, 119–120
  earning, 220–222
  failure resulting from leadership problems, 46, 48–49
  goal setting and, 106
  ineffective, 43–44
  making a difference, 98–99
  managers compared with leaders, 220–221
  overview of, 213
  problems, 48
  in self-directed teams, 74–75
  setting example for team, 215–217
  source materials for, 227
  symptoms of poor, 217–220
  team support for, 179
  transformational leaders, 222–224
Leading and coaching teams
  agreement as means of creating involvement, 113–114
  asking questions to stimulate involvement, 111–112
  building management team, 125–127
  coaching team leaders, 119–120
  commitment as a motivator, 102–104
  fear as a motivator, 100–101
  focusing on facts and data, 120
  greed as a motivator, 101–102
  leadership making a difference, 98–99
  making and sustaining commitments, 104–105
  managing experts to stimulate participation, 117–119
  motivation and, 99–100
  not allowing observers or outsiders in team discussions, 120–123
  overview of, 97–98
  playing dumb as means of encouraging involvement, 112–113
  power of coaching, 109–110
  preventing anyone from monopolizing discussion, 115–117
  rational management style, 127–129
  sensitivity to concerns or disagreements, 114–115
  short-term goals for creating a sense of urgency, 105–107
  source materials for, 129
  team involvement in selecting new members, 107–108
  team processes during storming phase, 123–125
  techniques for involving team members, 110–111
Lean and mean organizations, 145–146
Lighthouse example, 203–204
Life, getting satisfaction from, 209
Linberg, Kurt, 178
Listening, 56–57
Lister, Timothy, 41

Lone Ranger approach. *See* Self-sufficiency
Loser, behaving like, 185

M

Maintaining
  plans, 34–36
  teams, 88
Management. *See also* Self-management
  autocratic. *See* Autocratic bosses
  avoiding competition with, 149–150
  building management team, 125–127
  changes and, 144–145
  communicating with about needed changes, 156–157
  control issues and, 143–144
  dealing with unreasonable bosses, 145
  effects of incompetent planning, 28
  expectations for team leaders, 176–179
  function of, 220
  getting support for improvement programs from, 155–156
  identifying managers whose support is needed, 157–159
  identifying reasons why managers might support your project, 159–161
  informing of project progress, 141–142
  knowledge/awareness of problems in projects, 148–149
  leaders compared with managers, 220–221
  lower-level, 160
  negotiating project due dates with, 134–137
  not delaying communication of problems to, 202
  planning before making commitments to, 139–143
  required schedules, 26–27
  reviewing detailed plans with, 25–27
  role in making priority decisions, 48
  solution orientation vs. problem orientation, 149
  teaching managers to negotiate with you, 143–145
  working with teams, 231–233
Maslow's hierarchy, 100
MacArthur, General Douglas, 224
Measurement in diagnosis and improvement, 186–187
Measuring process improvement benefits, 168
Measuring quality
  partial measurement, 12
  personal measurement, 12
  user-based measurement, 13
Membership. *See* Team members
Methods, care in introducing in mid-project, 138–139
Microsoft TSP team, 234
Milestones, 154
Monitoring performance, 129
Morale problems causing team failure, 46, 50–51
Motivation
  autocratic bosses having negative impact on, 150–153
  commitment as a motivator, 102–104

Motivation (*continued*)
fear as a motivator, 100–101
greed as a motivator, 101–102
leadership's role in, 220
overview of, 99–100
performance and, 103
Multidiscipline skills, in team process, 43

N

Negotiating projects and defending plans
autocratic bosses and, 150–153
doomed projects and, 146–150
identifying an autocratic environment, 153–155
lean and mean organizations, 145–146
maintaining team focus on top priorities, 137–139
management expectations for team leaders, 176–179
planning before making commitments, 139–143
process improvement. *See* Process improvement
projects getting into trouble at beginning, 134–137
source materials for, 179–180
teaching managers to negotiate with you, 143–145
Negotiation
as communication skill, 58
as element of commitment, 102
with management. *See* Negotiating projects and defending plans
power of, 233–234
strategies of team members, 89–92
teaching managers to negotiate with you, 143–145
Norming phase, teams, 60–61

O

O'Brian, Bridget, 84
Observers/outsiders, not allowing in team discussions, 120–123
Open group, 66
Operational processes, TSP and, 239
Overhead, effects of cutting, 146
Ownership
of commitments, 105
properties of self-directed teams, 71, 73
responsibility based on, 199–202

P

Parochialism, 218
Partial measurement, steps in quality process, 12
Participation
creating synergy, 83–84
failure to participate as common problem in teams, 44–45
importance of, 84
nonparticipation hurting overall performance, 92–94
Paulk, Mark, 175
Peer evaluation, in team, 46
Peer pressure, team performance and, 92–93
Pelz, Donald, 192
Performance
autocratic bosses having negative impact on, 150–153
benchmarks, 128–129

comparing actual performance with estimates, 81
credibility and, 102–103
as element of commitment, 103
goals, focusing team on, 87–88
morale problems effecting, 50–51
nonparticipation by team member hurting team performance, 92–94
standards for, 216–217
team performance vs. individual performance, 42–43, 53
Performing phase, teams, 61
Period plans
comparing with product plans, 20–23
overview of, 20
Personal measurement, steps in quality process, 12
Personal Software Process. *See* PSP (Personal Software Process)
Phantom issues, fighting in high-pressure projects, 192–194
Planning
adjusting estimates and exploring alternatives, 136–137
commitment supported by, 124, 206–207
data for, 35–36
dynamic, 32–34
frequent plans to compensate for inaccuracy, 32–34
hardest time to plan is when it is most needed, 18–20
improving accuracy by reviewing previous errors, 196–197
incompetent, 27–30
maintaining plans, 34–36
before making commitments, 139–143
negotiating due dates based on, 136
overview of, 17
period plans and product plans, 20–23
product planning for each major task, 23–25
in PSP, 238
requirements to be met by, 30–32
reviewing plans with management, 25–27
in self-directed teams, 73
for short-term goals, 128
time management and, 197–198
tracking time as basis of, 196
updating plans, 35
uses of plans, 29
Playing dumb as involvement technique, 112–113
Polarization, avoiding in negotiation, 91
Poor leadership symptoms, 217–220
Power
autocratic decision making in power vacuums, 152
corrupting nature of, 151–152
of negotiation, 234
what do you want from life, 209–210
Precision, requirements plans must meet, 31–32
Pressure
managing, 193–194
software developers under, 230
Pretending (Covey), 57
Principled negotiation, 90–92

Priorities
goal setting and, 15
maintaining team focus on, 137–139
management role in setting, 48
managing commitments and, 208
Problem solving
getting and offering help, 94
team approach, 83
Problems
leadership, 48
morale, 50
team, 43–46
Process
design, 184–185
groups, 62–63
operational, 239
scripts, 239
Process improvement
benefits of, 166
building a case for, 155–156
business environment and, 162
CMMI and, 240
constant evolution in, 231
continuing costs, 166
defining proposal for, 162
facts and studies supporting, 166–167
identifying hot buttons, 163
identifying managers whose support is needed, 157–159
identifying reasons why managers might support your project, 159–161
introduction costs, 164–166
measuring benefits of, 168–169
prototyping, 164
PSP and, 237
reasons to make changes, 156–157
sanity checks, 163–164
savings from, 167
strategic case for, 161–162
tactical case for, 169–176
Procrastination, common problem in teams, 45
Product plans
comparing with period plans, 20–23
creating for each major task, 23–25
overview of, 20
what is included in, 25
Productivity, workplace stability and, 219
Programmers, defect prevention by, 7–8. *See also* developers
Programming, exacting nature of, 4
Projects
changing jobs in response to doomed project, 147
fighting phantom issues in high-pressure projects, 192–194
fixing problems in doomed projects, 148–150
getting into trouble at beginning, 134–137
maintaining control of, 141
plugging away on doomed projects, 147
what to do when a project is doomed, 146–147
Proposal for process improvement project, 162
Prototyping, process improvement project, 164
PSP (Personal Software Process)
building planning skills with, 27

CMM, CMMI and, 172
knowledge work and, 230
overview of, 237–238
personal planning and, 141
planning and, 20
software engineers and, 7
TSP based on training in, 240

Q

Quality
challenge, 3
design quality, 173
improving quality of your work, 4, 186–188
journey, 11
management, 142
managing commitments and, 208–209
poor quality as common problem in teams, 45
PSP and, 238
self-directed teams and, 70
standards for, 216–217
what do you want from life, 209–211
Quality ownership, 12
Questions, asking as involvement technique, 111–112

R

Random group, 67–68
Rational management style, 127–129
Recruitment, involving team in selection of new members, 107–108
Requirements
need for clear, 6
plans must meet, 30–32
quality program requires clear, 6
Requirements statement in PSP, 238
Resources, team failure caused by inadequate, 46–48
Respect
fear inhibiting, 100
managing commitments and, 209
Responsible, being, 199–202
Responsibility
based on ownerships and attitude, 199–202
delay is generally the worst alternative, 202–204
Reviewing plans, 25
Reward-based motivation, 101
Roles
of leaders, 214
team members accepting team roles, 86–87
team membership and, 41
Roosevelt, Franklin Delano, 224

S

Sanity checks, process improvement project, 163–164
Satisfaction with life, 209–210
Savings from process improvement project, 167
Schedules
commitments and, 139
cost estimating and, 27
defects and, 9
developers and, 26–27
failure to meet caused by impossible goals, 49
failure to meet caused by inadequate staffing, 47
plan updates and, 35
planning and, 20
slipping, 105

Scripts, process, 239
SEI (Software Engineering Institute), 18, 237
Selective listening (Covey), 57
Self-actualization (Maslow)
  fear inhibiting, 100
  greed substituted for, 101–102
Self-centeredness, qualities of poor leadership, 218
Self-directed teams, 69–75
  leadership in, 74–75
  management control issues and, 143–144
  overview of, 69–70
  properties of, 71–74
Self-management
  being your own boss and not being a victim, 185–186
  commitment as a state of mind, 204–207
  commitment management, 207–209
  considering what you want from life, 209–211
  delay is generally the worst alternative, 202–204
  designing a process for improvement, 184–185
  devoting yourself to excellence, 211
  fighting phantom issues in high-pressure projects, 192–194
  getting needed support, 194–195
  improving quality of work, 186–188
  knowledge work, 229–235
  learning to manage yourself, 230
  logical basis of time management, 196–198
  responsibility based on ownerships and attitude, 199–202
  source materials for, 211–212
  time management, 188–192
  work involved with, 231
Self-sufficiency
  balancing with team participation, 85–86
  getting and offering help vs. working alone, 94
Senior management, 158
Sensitivity to concerns or disagreements, 114–115
Shirking team goals, 53–54
Short-term goals
  for creating a sense of urgency, 105–107
  planning, 128
  translating long-term objectives into, 105–107
Skills, properties of self-directed teams, 71
Smith, John, 184
Soft negotiation, 90
Software
  CMM (Capability Maturity Model) for, 240–241
  development plans, 25–26
Software Engineering Institute (SEI), 18, 237
Software engineers. *See also* Developers
  finding/fixing defects, 11
  function of, 7
  importance of defects to, 9
  as pioneers of knowledge work, 229–235
  planning as critical part of job, 23
  PSP and, 237–238

Software quality
challenge of, 3–6
defects are not bugs, 10–11
eight steps for consistent quality, 5
goal setting and, 14–16
as never ending journey, 11–14
what it is, 6–9
Specificity, requirements plans must meet, 31
Standards for performance, 216–217
Status, 209–210
Steps, improvement, 187
Storming phase, teams, 60, 123–125
Strategic case for process improvement
benefits of, 166
business environment and, 162
calculating savings, 167
continuing costs, 166
defining proposal, 162
facts and studies supporting, 166–167
identifying hot buttons, 163
introduction costs, 164–166
measuring benefits of, 168–169
overview of, 161–162
prototyping, 164
sanity check, 163–164
Strategic thinking by managers, 158
Strategy, negotiating, 89–92
Studies, supporting process improvement project, 166–167
Support
costs of inadequate, 195
getting adequate, 191
getting and providing help, 94–95
getting needed support staff, 194–195
management support for change, 157–159
standards for, 217
Surowiecki, James, 153
Symptoms, poor leadership, 217–220
Synchronous group, 69
Synergy, participation creating, 83–84

T

Tactical case for process improvement, 169–176
changing commitment system of organizations, 175–176
code inspection project, 172–173
expanding small successes into larger projects, 175
instruction course project, 173–175
justifying small steps as alternative to large scale program, 171–172
options for overcoming management resistance, 170–171
overview of, 169–170
Taking charge, vs. being a victim, 186
Task
orientation of work groups, 62
time, 188–189
Team leaders. *See also* Leadership
coaching, 119–120

Team leaders (*continued*)
management expectations for, 176–179
principal job, 177
Team members
accepting/performing team roles, 86–87
building and maintaining team, 88–89
contributing with personal knowledge, 84–86
doing what is needed, 78
establishing and striving to meet goals, 87–88
getting and offering help, 94–95
goal setting by, 40, 81–83
involvement in selection of new members, 107–108
making and meeting commitments, 79–81
negotiation strategies of, 89–92
nonparticipation hurting overall performance, 92–94
overview of, 77
participation creating synergy, 83–84
properties of self-directed teams, 71
rewarding nature of membership in jelled teams, 51
source materials for, 95–96
success of, 178
team building obligations, 86
Team Software Process. *See* TSP (Team Software Process)
Teams
balancing workloads, 36
building and maintaining, 88–89
building management team, 125–127
challenging goals needed by, 52–53
closed group style, 68–69
cohesion of, 52
combat group style, 63–65
committing to common goals, 40–41
common problems, 43
communication skills, 56–58
development over time, 54–55
development professionals wanting to work in team environment, 178
effectiveness in performing complex creative work, 142–143
facts and data strengthening negotiation, 233–234
failure caused by leadership problems, 48–49
failures caused by impossible goals, 49
failures caused by inadequate staffing, 47–48
failures caused by morale problems, 50–51
feedback and goal tracking in, 53–54
forming phase, 58–59
goals, 40, 87
jelled teams, 51–52
leaders setting example for, 215–217
leadership behavior affecting, 213–215
leading and coaching. *See* Leading and coaching teams
maintaining focus on top priorities, 137–139

maintaining the team, 88
management working with, 231–233
norming phase, 60–61
obligations of team members in building, 86
open group style, 66–67
overview of, 39–40
performing better than individuals alone, 42–43
performing phase, 61
problems in, 43–46
process groups, 62–63
random group style, 67–68
reasons teams fail, 46–47
relaunch, to update plans, 34
roles, 86–87
self-directed, 69–75
source materials for, 75
storming phase, 60, 123–125
styles, 65–66
synchronous group style, 69
TSP team-building task, 240
types of groups and, 61–62
work groups, 62
working framework needed by, 54
working styles, 65–66
Technical support, 195
Test and fix steps in quality process, 11–12
Threats
combat groups and, 63–65
fear as a motivator and, 100
Time management, 188–192
breaks improving effectiveness, 191–192
focusing on critical tasks, 190–191
getting adequate support, 191
interruptions, 189–190
interspersing different kinds of work during day, 185
logical basis of, 196–198
managing commitments and, 207–208
tracking time use, 188–189
Transactional leadership, 223
Training costs, 165
Transformational leaders, 222–224
Truman, Harry, 224–225
Trust, commitment and, 79, 209
Trusting teams, 121
TSP (Team Software Process)
CMM, CMMI and, 172
description of TSP team, 40
developing and defending plan, 141
handling pressure, 194
knowledge work and, 230
launch process for addressing resource problems, 48
management control issues and, 143–144
negotiating plans with management, 20, 194
overview of, 239–240
team role definition, 86–87

U

Unbiased estimates, 32
Updating plans, 34–36
Urgency, short-term goals for, 106–107
User-based measurement, 13
Users, software quality related to needs of, 6

V
Valente, Judith, 84–85
Victimization
  be your own boss, not a victim, 185–186
  vs. being responsible, 200
Visibility
  of commitment, 104
  team styles and, 56
Voluntary nature of commitment, 104, 205

W
Winners
  behaving like, 185
  don't complain, 185
Work groups, 62
Working framework
  common working framework needed by effective teams, 54
  goals providing, 82
Working styles, groups
  closed group, 68–69
  open group, 66–67
  overview of, 65–66
  random group, 67–68
  synchronous group, 69
Workplace stability, productivity and, 219

Z
Zimbardo, Philip, 151