PART I

Team Formation

Most of us understand coaching and its purpose because of our early experiences on athletic teams. Although few of my early coaches were especially memorable, Coach Umbach was truly amazing. He wasn't a particularly impressive guy: only about 5 feet, 6 inches tall and 158 pounds. But no matter how big you were, if you ever got on a wrestling mat with him, you would never forget it.

Throughout my high school education, I had enjoyed sports but was never much good. This was frustrating because I was willing to practice and work out, but somehow my natural talents did not include ball games. My dad had been a wrestler in college and had enjoyed it, so when I got out of the Navy and went to college, he urged me to try out for the wrestling team. Since I was the only light heavyweight who tried out, I made the cut and was told to show up for practice. As we waited for the coach the next day, I got to know my teammates and it turned out that none of us had ever wrestled before. No one knew what wrestling practice involved or that coach Arnold W. "Swede" Umbach was a truly extraordinary coach.

When the coach arrived, we started with a brief warm up and then a two-mile run, with the coach in front. Then he paired us up and had us work on a few basic holds. The coach took turns wrestling with each of us. After an hour and a half, when we were all pretty beat, we ended with another two-mile run, again with the coach in front. The workouts were so tough that the matches seemed

easy. By the end of the year, several of us were undefeated, the team took the 13-state championship, and we were campus heroes. All of this from a ragtag bunch of inexperienced recruits. It was Coach Umbach who made the team.

Our coach's dedication, commitment, and energy were amazing, but what I found most inspiring was that he really cared about how each of us did. I have always remembered how he made a small band of raw recruits into a championship team and how he fostered the kind of cohesive team spirit that made losing simply unthinkable. I remember on my second match, after completing the regular three rounds, I was laying flat on my back, so exhausted that I knew I couldn't even get up for the final two tie-breaker rounds. All I could hear was Coach Umbach whispering in my ear that the other guy was every bit as beat as I was. All I had to do was get in there and "explode." To this day, I don't know how I did it, but I did, and I won.

When writing this book, I learned that I was on the first Auburn team Coach Umbach coached and that, in his 27 years at Auburn, he had 25 Southeastern Intercollegiate Wrestling Association Championship teams, 127 conference champions, and 4 national champions. In 282 competitive meets, his teams only lost 28. "Swede" Umbach was an inspiring coach then and he has been an inspiration to me ever since.

This book is about coaching. While it provides lots of tips, hints, and guidelines, the key thing to remember is that your attitude and commitment to the team and its members will largely determine how quickly your team turns into an effective working unit and how rapidly the members grow and develop their skills and capabilities. While you share the responsibility for the team with the team leader, your dedication and commitment to the team and every one of its members is critically important to their performance. Part I of this book provides the context for all that follows. Its four chapters cover various aspects of teams, teamwork, and team coaching.

Chapter 1 discusses what teams are, the nature of self-directed teams, and why such teams are particularly important for development work. It also describes the importance of teamwork in developing modern large and complex computer-based systems as well as the teamwork issues involved in doing creative and innovative work.

Chapter 2 discusses team behavior, how teams work, and the various kinds of teams. It also discusses the conditions required to make teams effective along with the various styles and personalities exhibited by teams. This chapter describes a few of the most common team types and the methods that are most effective for coaching them. Several examples illustrate key aspects of team behavior and help explain the team characteristics to consider as you coach a development team.

Chapter 3 covers your responsibilities as a TSP coach, what you do, and how your role changes as your team and team leader gain experience. The chapter also discusses how you can support the team and its leader in becoming self-

sufficient and in learning how to be fully effective. The chapter closes with a discussion of your relationship with the team leader.

Chapter 4 describes the teambuilding process, various teambuilding approaches, and how the TSP launch builds teams. The chapter also reviews the need for team involvement in the launch process and provides guidelines to help you obtain team members' participation in that process. It also discusses why observers are excluded from most of the TSP launch meetings and provides examples of what typically happens when visitors are permitted.

1

Development Teams

This chapter discusses why teams are needed, the innovative nature of development work, and the conditions required for teams to do such work. It then describes what a team is, the distinction between teams and work groups, and the various kinds of teams. Finally, the chapter discusses how to build and run high-performing teams, together with an overview of how you, the team coach, contribute to these activities.

Before starting the team and team-working discussions, it is important to first briefly describe the Team Software Process (TSP). This is because much of this book describes how to coach teams that are using the TSP to guide their work. While the coaching concepts and methods described will apply equally well to other types of teams, the TSP is a powerful process and, by using it to guide your teams, you will find it easier to be an effective coach and to produce "winning" teams.

1.1 TSP Overview

The TSP process is designed to guide development teams while they design and develop software-intensive systems. While it will work for many other kinds of

teams, its initial focus was on software development work and that is where it has been most widely used.

The TSP process guides development teams and their management in planning and developing quality products on predictable schedules. It provides detailed guidance and its process scripts lead the developers through launching and operating their teams. During the initial launch process, the team first learns the product requirements and project goals from management. It then produces a plan to meet these objectives. At the end of the generally four-day launch, the team meets with management to agree on the team's plan and delivery commitments.

Following the launch, the TSP process guides the team and team leader in working as a self-directed team. As described later in this chapter, a **self-directed team** defines its own strategy and process, produces its own plan, and negotiates it own commitments with management. Experience shows that when the members of such teams are properly trained, led, and coached, they typically deliver quality products on schedule. They also find their work more productive and much more rewarding. For more information on the TSP process, the required training, and the benefits of using the TSP, see the books *Winning with Software: An Executive Strategy; PSP*^{SM}*: A Self-Improvement Process for Software Engineers;* and *TSP*^{SM}*—Leading a Development Team* (Humphrey 2002; Humphrey 2005; Humphrey 2006).

1.2 Why Teams Are Needed

Modern systems are becoming increasingly complex. Aircraft, automobiles, computer printers, television sets, and even electric razors contain software—often lots of software—and the amount of software in these products has been rapidly increasing over time. The design of such systems is vastly more complex than it was only a few years ago. Although there are still many relatively modest-sized systems, the trend is for the software content of just about every product to increase by about ten times every five years. As shown in Figure 1.1, this trend has more or less occurred for decades, and it appears likely to continue for the foreseeable future.

Today, many development projects simply cannot be handled by an individual. Most jobs are just too large for one person to complete in a reasonable time. Time-to-market is the critical performance measure in many industries, and the speed with which organizations form and deploy teams has been the single most important factor in determining competitive success.

Teams also have a range of talents and capabilities. Some jobs are so complex and involve so many different specialties that one person could not possibly

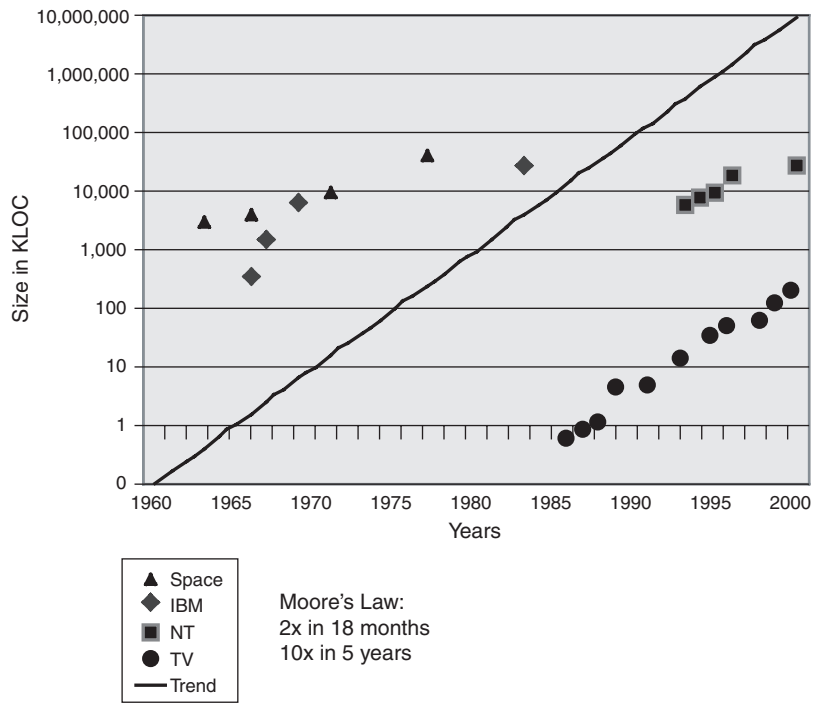

Figure 1.1 Program Size Growth

handle them: No single person could know enough or have a broad enough range of experiences. With a team, the members can specialize on parts of the job and concentrate on those tasks that best fit their abilities. Reasonably sized teams contain a mix of talents and, when these talents are fully utilized, such teams can achieve better performance than any of their individual members could achieve alone.

Participation on a team also improves individual performance. Members often have complementary skills and can learn from each other while providing mutual help and support. Examples of such support are product inspections, group design sessions, and one-on-one assistance. By working with more experienced developers, novices can learn more quickly than they could alone. When properly formed, led, and coached, teams can provide a rich and stimulating environment for building development talent.

1.3 What Are Teams?

A team is a group of people who share a common goal. They must all be committed to this goal and have a common set of practices to guide them as they work to achieve that goal. The definition that best describes a TSP team is one that I have adapted from Dyer (Dyer 1984, 286).

- A team consists of at least two people.
- The members are working toward a common goal.
- Each person has one or more specific roles.
- Completion of the mission requires interdependency among the members.

Every one of these elements is important. For example, it is obvious that a team must have more than one member. The need for common goals, however, is more subtle. When groups of people work together without a common goal, they have no need to interact or to support each other. Without some mutual goal, people do not strive—they just put in their time. Then they can have no common group focus and there can be no team. Even if the goal seems simple, such as processing all of today's orders before quitting time, common objectives energize teams.

A principal requirement for team membership is that all of the members have a common set of working guidelines and practices. For example, on sports teams, they must all play the same game. While the reason seems obvious in athletics, it is not so clear in development. However, unless everyone follows consistent guidelines and practices, their work will be unrelated and it will be difficult, if not impossible, for them to cooperate. For a team to work cooperatively and effectively, the members must have a common set of working practices and strive to meet a common team goal.

The reason that team members need roles is not as obvious. Roles provide a sense of ownership and belonging, and they guide the members in doing their work. Roles also ensure that someone is assigned to handle each of the team's important tasks, and they help to spread project management responsibilities across the team.

A further important aspect of team performance is cooperation and interdependence. Each team member must depend to some degree on the performance of the other members. With a common goal and strong team-member interdependence, teams develop the trust and cohesion required to jell. DeMarco and Lister define a jelled team as follows (DeMarco 1987, 123).

> A jelled team is a group of people so strongly knit that the whole is greater than the sum of the parts. The production of such a team is greater than that of the same people working in unjelled form. Just as important, the enjoyment that the people derive from their work is greater than what you'd expect given the nature of the work itself.

1.4 Kinds of Teams

In sports, there are different kinds of teams. For example, a basketball team has a high degree of interdependence. Conversely, wrestling and track teams have individual competitors who support each other emotionally but not interactively. Baseball team members have relatively static positions or roles, but they all must work cooperatively for the team to win. While tennis doubles partners may dynamically switch their roles, the quality of their play depends on how well they cooperate and support each other.

Development Teams

In development work, there are also different kinds of teams. **Project** teams often behave much like baseball and basketball teams. They may have multiple specialties, like hardware design, systems integration, software implementation, and test. However, all of these specialties must work closely together to produce a quality product in the shortest possible time.

On systems maintenance and enhancement teams, the developers are often relatively independent, much like wrestling and track teams. Maintenance activities are often handled by what we call **functional** teams, while development groups are called **project** teams. Although the TSP supports both types of teams, the discussions in this book principally concern project teams. Most of this discussion will apply equally to both project and functional teams; some of the special considerations for functional teams are covered in Chapter 22.

Team Size

Teams cannot have fewer than two members, but there is no clear limit on how big they can be. While teams presumably could be of almost any size, high-performance groups tend to be quite small. This is because when teams have more than about 10 to 15 people, they lose the intimacy that makes them effective. This is true in sports, the performing arts, research, and development.

The upper limits on TSP team size are set at around 12 to 15, although we typically say that TSP teams can have up to 20 people. When teams get much larger than about a dozen members, the recommended practice is to break them into multiple, smaller teams and use the multi-team process (TSPm is described in Chapters 21 and 23). The TSPm process preserves the benefits of small team size while retaining many of the advantages of a single overall project team.

Small, Large, and Mega-Teams

Team characteristics and structure change with team size. At the low end with two developers, the members can work together informally and the more structured communication and coordination mechanisms needed for larger teams are unnecessary. Even though two-person teams can use a much less formal process, the TSP principles and the full range of TSP launch activities are still needed to define goals, produce plans, and negotiate commitments with management.

The basic TSP process has been designed for teams with about 3 to 15 members. This is where it works most effectively. These basic or unit teams each have a single team leader who directly supervises all of the team's work. As team size grows, it becomes increasingly difficult for the team leader to closely monitor the work of all the members. Depending on the team leader, the needs of the team members, and the nature of the work, large teams should be handled as multiple teams, with a separate leader for each unit team. These leaders directly supervise the team members, while the overall program manager supervises the team leaders through what is called the **leadership team**.

A single multi-team process can handle teams of up to a few hundred members. As the number of unit team leaders grows, it soon becomes impractical for a single program manager to directly supervise all of the unit team leaders. One way that program managers deal with this problem is to break the overall team into several multi-teams, each headed by an assistant program manager. A second approach is to form a program office, with an associate program manager to share the workload. A third approach is to form a program manager staff to handle many of the program manager functions. When team size grows beyond about 1,000 or so members, the issue of team structure becomes one of organizational design. Using the TSP with such mega-teams is discussed in Chapter 24.

1.5 The Nature of Self-Directed Teams

For development work, it is important to have motivated and energetic teams. On such teams, the members sense what is needed without being told, pitch in to help, and do whatever is needed to get the job done. This is their job, they own it, and they intend to finish it. This is the characteristic behavior of what we call **self-directed teams**. Self-directed teams stick together to the end of the job. Typically, employee turnover on such teams is zero. The members may know that the team will be dispersed, the organization disbanded, or the contract transferred, but this is their project and they will do their utmost to see it through. Their principal commitment is to the team, not to the project or the organization.

While a self-directed team would be useful for any kind of job, such teams are essential for complex and creative development work. This kind of work requires everyone's wholehearted participation. If the team members are not committed to the project and dedicated to its goals, they will not strive to do a quality job. Quality work is not done by mistake; it is done by thinking, caring, and motivated people. Self-directed teams have some special properties that set them apart from all other teams. The following are the six properties of self-directed teams.

- A sense of membership and belonging
- Commitment to a common team goal
- Ownership of the process and plan
- The skill to make a plan, the conviction to defend it, and the discipline to follow it
- A shared commitment to honest, truthful, and respectful behavior
- Dedication to excellence

Membership and Belonging

The members of a self-directed team are part of a cohesive and distinct group, and there is no question who is on the team and who is not. All of the members share a common bond of membership, and they seem to have a special way to communicate. They are so familiar with the job and with each other that they can almost speak in shorthand. The most impressive aspect of a self-directed team is the way that its members work together. Cooperation is the essence of teamwork and, when team members do not cooperate, they cannot build the trust and spirit required for effective teamwork. Self-directed teams are close-knit and cohesive, and every member is a valued contributor.

Cohesion is the bond that knits the members together. Cohesion requires contact and close association. The team members must share a common workspace, see each other regularly, and communicate freely and openly. You can't legislate cohesion; it is a consequence of the team's working context. Cohesion is a fundamental property of a self-directed team.

Team cohesion is strengthened by the support the members provide to each other. Human beings are social animals, and few like to work entirely by themselves, at least for very long. Team membership provides a comfortable human environment and a source of mutual commitment, support, and motivation. All of the members of such teams make a special effort to meet their obligations to their teammates.

When a team does not have clear boundaries and its members seem to randomly drift on and off the team, the members cannot rely on each other. This is

the principal problem with part-time team members. When developers are simultaneously assigned to several projects, they have split loyalties, and their teammates cannot count on them for support and assistance.

While it is normal for developers to have demands from prior projects, these must be the exception, and every developer should have a principal project assignment. Teams with a substantial number of part-time members rarely jell. When management is unwilling to commit a developer to a project, that developer is not likely to feel fully committed to that project or make the kind of effort that produces superior results.

Commitment to a Common Goal

Self-directed teams share a commitment to a common goal. The importance of the goal, however, is not so much because of its value to the organization as it is to the team. The team members' motivation comes from the common commitment that the team has made to meeting the goal. This is what the members have decided to do, and they will do their utmost to bring it off.

To maintain this commitment, the team must receive feedback on its work. Whoever heard of a winning team that didn't know the score? To be motivated, teams must know when they are ahead and when they are behind. They must see progress every day. Only then can teams continue pushing to achieve their goals. For example, studies have shown that to achieve high personal and team performance, feedback is the single most important ingredient.

> Goal tracking and feedback are critically important. Effective teams are aware of their performance and can see the progress they are making toward their goals. In a study of air defense crews, those with frequent and precise feedback on goal performance improved on almost every criterion. This compares with the stable, unimproving performance of crews that did not get feedback. (Humphrey 2000, 21)

Owning the Process and Plan

Another property of self-directed teams is a sense of ownership. This is not just *any* job these teams are doing; it is *their* job. They feel responsible for it and have decided just how to do it. Such teams speak of their work with a special pride. To feel this sense of ownership, teams must define their own processes, produce their own plans, and track and report on their own work. The members must be solely responsible for doing this job, and they must know that nobody else will do it. This responsibility provides a sense of personal importance and a feeling of self-respect.

Finally, when a team has a defined process and a detailed plan, the members will know what to do. While this seems obvious, it is also fundamental. When a group is unsure about what to do and doesn't know where to get guidance or help, it cannot jell. It is merely a group of confused people looking for direction. Under these conditions, the members will work to different priorities and not effectively support each other. Following a process and a plan provides stability and builds the team's motivation and energy. To be self-directed, teams must have a common goal, they need a defined and understood process, and they must have a detailed plan.

Skill and Discipline

Self-directed teams are especially well-suited for development work. They define the process and the plan for doing the work, they believe in and will defend that plan, and they have the discipline to follow the plan. Discipline, in fact, is what separates the experts from the amateurs in any field. Their willingness to rehearse, to practice, and to continually improve is what makes them experts and what makes superior work so natural that they can devote their energies to being creative. Studies have shown that the principal distinction between world-class performers and those who finish in the middle of the pack is their disciplined behavior (Gawande 2002).

Honest, Truthful, and Respectful Behavior

The essence of teamwork is trust: Do the members all trust each other to do their jobs? While trust is a nebulous concept, most of us know it when we see it. In fact, learning about trust is a big part of growing up. Most of us start life as trusting dependents: Without even thinking about it, we trust our parents to take care of and support us. However, those of us with siblings soon learn that other people are not always trustworthy—they take things we want and do things that we don't like.

Depending on our personal backgrounds and experiences, we may or may not be naturally trusting people. What is even more important, we may or may not be trustworthy people. While many people feel that the key to a truly superior team is finding the right teammates, that is not the case. The real key is to *be* the right teammate. To have a truly rewarding team experience, every team member must be trustworthy. That means that all of the members must be trustworthy and respect the rights and needs of their teammates, deal honestly and openly with all team members, and respond rationally to the guidance and leadership of the coach and team leader.

A Dedication to Excellence

The final property of self-directed teams is their dedication to excellence. For teams to work cooperatively and to maintain their energy and motivation, all members must strive to do more than just their share. Everyone volunteers for the tough assignments, pitches in, and contributes to the best of his or her ability. The spirit and energy of such teams depends, however, on the quality of everyone's work. If one member does sloppy work, makes frequent mistakes, and causes excessive rework, it wastes everyone's time. If this happens often, everyone will know why and resent it. Poor work by any team member can quickly destroy the team's spirit.

1.6 The Team Leader and Coach Roles

While all of the steps of teambuilding, training, goal setting, and feedback are necessary, they alone will not produce high-performance teams. That takes leadership and coaching. The team leader and coach are both essential in team formation and operation. The coach role is, in fact, the subject of this entire book. Figure 1.2 provides a conceptual view of the various aspects of team leadership and coaching, and how they relate in building and supporting teamwork.

Your role in building and maintaining effective teamwork is the topic of this book. After you have read this book, you will understand the conditions for effective teamwork, and you will know how and why to establish these conditions. This will prepare you to build and coach a self-directed development team.

1.7 Coaching Workload

Coaching is a full-time job. While one coach can generally support several TSP teams, there are periods, particularly with new teams, when the coach must be available full time to support just one team. Even after teams have been coached through several months of a project, they must still be monitored to ensure that they are performing the process properly and that they are effectively gathering and using their data to both guide their work and to improve their performance.

The amount of time required to coach a development team can vary widely. It depends on the TSP experience of the members, team size, the nature of the work, and the business environment. However, one coach should generally be able to support three to five teams of six to twelve developers each. Since there will be periods when one of these teams will occupy the coach full time, it is

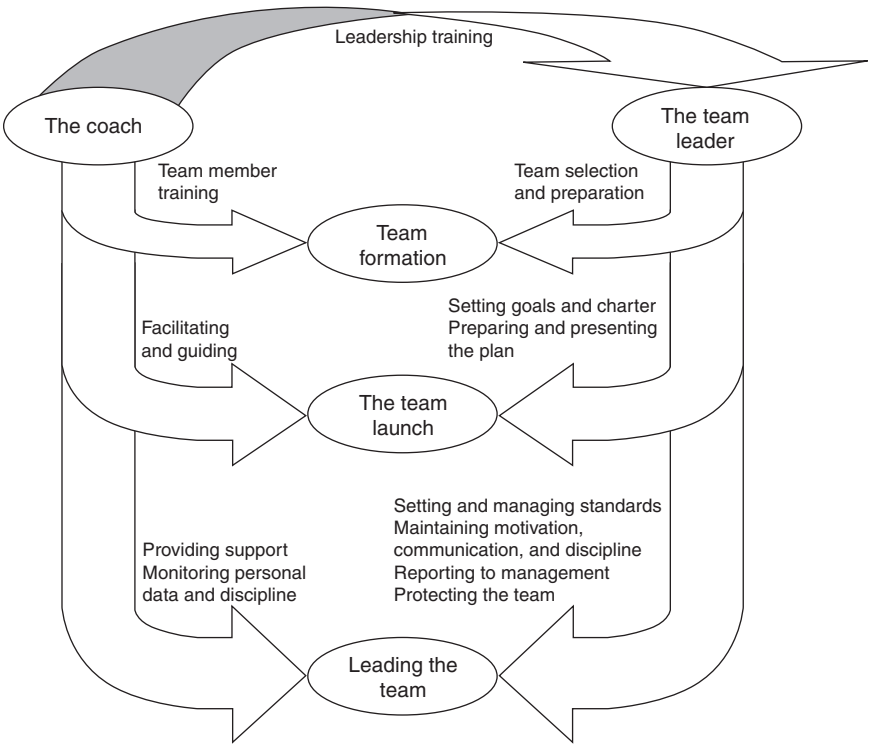

Figure 1.2 The Team Leader and Coach Roles

always desirable to have at least two coaches in any organization, even if there are only two teams.

1.8 Summary

Today, many development jobs are so complex and involve so many different specialties that they simply cannot be done by any one person. Teams are needed for most jobs and, when the members are fully and effectively utilized, these teams can achieve better results than any individual member could produce alone. By participating on a team, developers both improve their personal performance and learn from their peers.

In sports, there are different kinds of teams. Basketball teams have a high degree of interdependence, whereas wrestling and track teams are composed of individual competitors who support each other emotionally but not interactively. In development work, maintenance projects are like wrestling and track teams, and they are often handled by what are called *functional* teams. Development groups generally behave more like basketball teams and they are called *project* teams.

A team consists of at least two people who share a common goal and who have specific roles. They also need an agreed-upon strategy, a common process, and a shared plan. The team members must communicate freely, they must be committed to the job, and they must know their status as they do the work. The TSP process provides a defined framework for building and sustaining self-directed teams that have these properties.

While all of the steps of teambuilding, training, goal setting, and feedback are needed to produce a high-performing team, they alone are not sufficient. The final key ingredient is leadership and coaching. The team leader and coach are essential to team formation and effectiveness. Your role as a TSP coach is the subject of this book.

References

DeMarco, Tom, and Timothy Lister. *Peopleware: Productive Projects and Teams.* New York: Dorset House Publishing Co. 1987.

Dyer, Jean L. Team Research and Team Training: A State-of-the-Art Review. *Human Factors Review.* The Human Factors Society, Inc. 1984, pp. 286 and 309.

Gawande, Atul. The Learning Curve. *The New Yorker.* January 26, 2002, pp. 52–61.

Humphrey, Watts S. *Introduction to the Team Software Process*[SM]. Boston: Addison-Wesley. 2000.

Humphrey, Watts S. *Winning with Software: An Executive Strategy.* Boston: Addison-Wesley. 2002.

Humphrey, Watts S. *PSP*[SM]*: A Self-Improvement Process for Software Engineers.* Boston: Addison-Wesley. 2005.

Humphrey, Watts S. *TSP*[SM]*—Leading a Development Team.* Boston: Addison-Wesley. 2006.