Reflections on Management

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

Watts S. Humphrey
with William R. Thomas

Praise for *Reflections on Management*

"Watts Humphrey is best known for software engineering approaches that formalize and systematize software development. But I personally have found Watts to be most inspiring when he is discussing the interpersonal, human side of the software equation. The selections in this book emphasize Watts's deep experience and deep insights into human dynamics and offer a valuable counterpoint to more programmatic writings for which he is better known."

Steve McConnell
Author of *Code Complete* and *Software Estimation*

"Watts Humphrey is doing for the software industry what W. Edwards Deming did with TQM for the automobile industry. For the software executive reading this book, it is my belief that Watts has developed a real weapon for beating your competition: a reliable, repeatable way to create software that has excellent quality *and* reduces the time to deliver it to your customer *and* lowers the cost of the entire software life cycle *and* improves employee morale. All at the same time! For the software engineer or developer reading this book, it is my belief that you are on your way to making your job more productive, satisfying, and fun."

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

"Software development is a daily collision between code, the most black-and-white of technologies, and organizations, the most idiosyncratic of human experiences. Here is the guidebook—the GPS—to success in navigating the fault line between science and art, between code and human experience. Sharing his knowledge with his characteristic style of analytics and anecdotes, Watts Humphrey imparts timeless wisdom on teams, teamwork, and creating complex software successfully and reliably."

Scott D. Cook
Founder & Chairman of the Executive Committee, Intuit Inc.

"Fledgling project managers often ask me if there is *one* book they should read; it was often difficult for me to resist the temptation to recommend Machiavelli's *The Prince*. But now I have a better, and far more positive, recommendation: Watts Humphrey's *Reflections on Management*—a collection of management gems, organized into eight broad themes, that reflect his deep insights and leadership in the field."

Ed Yourdon
Consultant and Author

"I've followed Watts Humphrey's work for as long as I can remember. I recall, in my youth, thinking that he was asking too much. Now that I'm suddenly about his age, I realize how many things he has gotten right. This collection from his most important writings should bring these ideas to the attention of a new audience: I urge them to listen better than I did."

Ron Jeffries
www.XProgramming.com

"You will enjoy this collection for its down-to-earth, accessible prose, its pragmatism, its optimism and, above all, Watts's demonstration that software

The Storming Phase

Teams next go through what is called a **storming** phase. This is often an emotional period where the developers react to the new assignment and to the seemingly impossible challenges. Teams often have no idea how to attack the job and feel threatened by the many unknowns. The requirements are usually ill-defined or unavailable, and the demanded schedule is often impossibly short. No one knows what they are supposed to do, the members often do not know each other, and they may even have a new and unfamiliar team leader. Finally, the developers may have to use a new process or work with unfamiliar tools and technologies.

While even an experienced team would be threatened by any one of these conditions, most new teams face all of these challenges at once. Often, their first reaction is to get emotional. They may challenge the assignment, object to the launch process, or vent their anger and frustration at any convenient and nonthreatening target. The coach is often just such a target.

Although the team's reactions in the storming phase are generally emotional, their concerns are real. The project description is often totally inadequate and the schedule is usually impossibly tight. In the storming phase, teams struggle through the painful process of discovering that they must attack their problems logically. Once they start to focus on how to solve their problems, they will quickly establish an effective way to work together and to get the job done.

The Norming Phase

In the **norming** phase, teams start to address their problems. They establish goals, define roles, determine strategies, and produce plans. In the norming phase, Jeff's team produced a plan. The developers first created a product conceptual design and agreed on the development strategy. They defined the develop-

ment process and estimated the sizes of the major product components. Finally, they made an overall estimate for the total job.

With the help of the coach and the TSP process, the team accomplished this work in only three days. While the developers had originally thought that they could not find any data, they quickly found useful information about several prior projects. Based on these and their own PSP data, they concluded that the job would take them 18 months instead of the 9 months that management had requested. They were confident that their plan was realistic and were ready to defend it.

The Performing Phase

Finally, in the **performing** phase, the team members presented their plan to senior management and the marketing executives. After a rather heated session, they convinced the group that their plan was realistic, and that it was the best that management could expect. By following the TSP process, teams can rapidly work through the forming, storming, and norming phases. Then they are prepared for their first real challenge: to sell their plan to management.

After some debate, management accepted the team's plan and the team started on the job. The project was a resounding success: the team finished a few weeks ahead of the 18-month schedule, and the overall hours were within 10 percent of their original estimate. The finished product had even higher quality than the team had planned.

3.10 THE BEST KIND OF GROUP

By definition, a team is a group of people. Berne has classed groups into three kinds: the **work group** concentrates on the job, the **process group** focuses on internal dynamics, and the **combat**

group fights an external threat.[17] An understanding of these three group types can help you to assist your team in overcoming the group-related problems that could limit its performance.

The Work Group

The work group is usually easy to recognize. The members concentrate on doing the job and they focus on task-related activities. Take, for example, Jeff's team. During its storming phase, the developers were concerned about the lack of data. However, once they started to make the plan, they began to think about where to find relevant information. One developer realized that a major component was similar to part of a previously developed system. On the evening of the first day, she got a copy of the source code and measured that component's size. Another developer knew someone in the Swedish laboratory who had worked on a similar system. Early the next morning, he called his colleague and found out more about that kind of work.

In working groups, team members volunteer for jobs; they don't jockey for position or worry about who has the most to do. They don't complain about unreasonable management, and they don't simply listen to the team leader and the coach. As they think seriously about how to do the job, they can be very creative. This is how teams get committed, and it is a necessary step in producing a sound plan.

The Process Group[18]

The process group is concerned about the team's internal structure and behavior. The developers often have no defined roles

17. Eric Berne. 1966. *The Structure and Dynamics of Organizations and Groups.* New York: Grove Press.

18. While the name *process group* may be a little confusing to those who have worked with software engineering process groups (SEPGs), this terminology has been used for many years by those who study the behavior of groups.

and have not established a pecking order or agreed on how to work together as a cooperative team. The group's energies are concentrated on internal issues. They are not yet ready to get to work.

When teams do not get right to work, it usually means that one or more members are uncomfortable or unclear about their roles or assignments. Get these issues on the table and deal with them directly. With small teams, such problems can usually be handled quite quickly. On larger teams, however, it usually takes a great deal longer. This is both because there are more people involved and because there are often several subteams with separate team leaders. If any subteam leader has a role or responsibility concern, those issues must be resolved before their teams can become work groups. Once the team leaders are in agreement, their subteams can start to resolve their internal issues.

Process issues concern responsibilities and roles. The best way to deal with these issues is to have a team assign specific role responsibilities to its members and to develop detailed plans. When all team members have assigned role responsibilities and when the team has made a detailed plan for doing the work, the process issues will generally go away. This approach will handle most, but not all, process issues.

The Combat Group

The combat group is fighting an external threat. This threat is typically perceived as an attack on the team's mission or responsibilities. The team may think that its very existence is at stake. Mack's team was working as a subcontractor on part of a large project. His group was in a separate department from the larger group, and all the developers in this department worked for him. Mack's people were just finishing one job and there was no other immediate next job for them to do.

The project manager wanted Mack's entire staff transferred to his department. However, Mack's management wanted to keep them as a separate development group. Mack was in a difficult political position. While he was committed to doing the subcontract, he knew that his management wanted his group to remain separate from the larger central development group.

Mack was a developer and didn't like politics. He started sending an alternate to the weekly project management meetings and began to deal very formally with the rest of the project. When Mack's team made a major change in its development strategy, Mack decided not to make a new plan. This made status hard to measure and progress almost impossible to track. The program manager became concerned about Mack's dependability and raised the issue with senior management. When rumor of this escalation leaked out, Mack's team felt even more threatened.

Combat groups are often difficult to deal with. They feel threatened and often behave illogically. This generally makes the situation worse, and may actually increase the threat. As the threat increases, the team members submerge their internal process issues and concentrate on defending the team. It is then almost impossible to find out what bothers the developers. They are concentrated on repelling the perceived attack and will view any outsider as part of the threat. Even the coach will have trouble helping teams that are in combat mode.

The best way to deal with a combat group is to identify the perceived threat and to deal with it directly. Either get management to make the feared change or demonstrate that the team's fears are unfounded. The unknown is generally more frightening than the known, so by confirming a team's fears you will actually reduce the threat. This converts an unknown and potentially frightening risk into a known and specific issue to be addressed. By confirming the team's fears and working with the

team on a plan to handle those fears, you can show that the situation is not as bad as the team had feared.

When a team is behaving like a process or a combat group, it will not work effectively. Its energies will be largely devoted to addressing the perceived internal or external threat. While the team may seem busy and focused, the members will be uneasy. They will know that the team is not performing effectively. Examine the group's behavior and talk to the members, both individually and as a group. Try to understand what troubles them and then, together with the team leader, directly address their concerns.

3.11 TEAMS ADOPT VARIOUS WORKING STYLES

Even when they are working effectively together, teams can adopt various working styles. There is no right or wrong style, and each team style is appropriate under different conditions. By understanding these styles, you can guide your teams to adopt the working styles that best handle the situations they currently face. Larry Constantine defines the four styles of team behavior as the *open group*, the *random group*, the *closed group*, and the *synchronous group*.[19]

These four styles are the extremes. Real team styles generally occur in combination, as shown in Figure 3.1, and it is rare to have a team behave purely in a closed style with no synchronous, open, or random characteristics. Typically, teams will operate in some position nearer to the middle of this figure.

19. Larry L. Constantine. 1993. "Work Organization: Paradigms for Project Management and Organization." *Communications of the ACM* 36, no. 10, 35–43.