



H3RALD

Book Review: Reflections on Management

The key to becoming a true leader

by Fabio Cevalco

Saturday, 05 February 2011

Originally published on H3RALD.com

Table of Contents

About Watts S. Humphrey	2
Structure and Organization	2
Writing Style.....	3
Contents.....	3
Part I: Managing Your Projects	3
Part II: Managing Your Teams.....	4
Part III: Managing Your Boss	4
Part IV: Managing Yourself	5
Final Thoughts	5

When I was offered to review this book, I was a bit skeptical: a book on *management*? I normally read and review books on programming and software development methodologies. However, I work as a Documentation Technical Leader, and while I don't technically *manage* a whole team yet (damn economic crisis!), someday I may end up doing just that, so I gave [Reflections on Management](#) a try.

It's short, after all, I'll probably read it in a couple of weeks and move on — I thought. Well, beware of short books: I thought exactly the same thing when I picked up [The Elements of Style](#), and it still follows me around everywhere, so that I can re-read bits of it whenever I need to.

This short but dense masterpiece by Watts S. Humphrey and William R. Thomas is one those books you always end up carrying around in your pocket (or stored in your favorite ebook reader). It's a short but very dense collection of tips and tricks to succeed as a leader and a manager — of *anything*: "*Your Software Projects, Your Teams, Your Boss, and Yourself*", as the book subtitle says. It doesn't "just" help forging great managers and leaders, it also explains, with practical examples and no-nonsense

explanations, how to deal with those annoying people in suits who constantly keep asking you for impossible things...

About Watts S. Humphrey

Generally, I don't bother writing anything about the authors in my reviews: you can easily find this kind of information online if you want to. I'll make an exception in this case, you'll understand why as you read along.

[Watts S. Humphrey](#) was a true legend in Software Engineering, he's often referred to as *The Father of Software Quality*. He worked at IBM for 27 years and eventually became Vice President of Technical Development. In the 80s, he arrived at the [Software Engineering Institute \(SEI\)](#) where he developed some key development processes of our time: the Software Capability Maturity Model (CMM), the Personal Software Process (PSP), and the Team Software Process (TSP). He received many awards, culminating with the *National Medal of Technology* in 2005.

He wrote several books focusing mainly on software development and managing software projects through his PSP and TSP methodologies. *Reflections on Management – How to Manage Your Software Projects, Your Teams, Your Boss, and Yourself* was the last book published while he was alive. [Leadership, Teamwork, and Trust: Building a Competitive Software Capability](#), co-authored with James W. Over, was published posthumously.

Watts S. Humphrey [died](#) on October 28, 2010.

Structure and Organization

In many ways, *Reflections on Management* can be seen as the *summa* of Humphrey's work on PSP, TSP and management of software projects, condensed in a very readable 288-page-book, co-written with [William R. Thomas](#), Senior Technical Writer and manager of SEI's Technical Publications Team.

I noticed the tech writer's touch simply by flicking through the pages of the book when I first got it: its structure is impeccable.

Organized into four parts, totalling 8 chapters, an Epilogue and an Appendix, this book is a prime example of order and readability: pick any section title (just the title) of any section in any chapter, and you get a clear idea of their content and purpose, and a key principle of management. Examples? Sure:

- Chapter 8: Learning to Lead
 - 8.1 How You Behave Affects Your Team
 - 8.2 Leaders Set an Example for Their Teams
 - 8.3 Learn to Avoid the Symptoms of Poor Leadership
 - [...]



If you print the Table of Contents alone you get a priceless cheat sheet on management and leadership. If you want slightly more detail, each chapter contains a summary table of all its sections, with a two-line summary of its contents. There are no subsections, only first-level sections, which make the book much easier to understand and “digest”.

You can read it all at once, then you should keep it readily available for consultation. It will take you only a few seconds to look through the contents and pick the most relevant section in a time of need.

Writing Style

The book is very clear and simple to read, always. Each section is self-contained, and always aims to make a point, usually expressed right in its title. If I were to find a common pattern in most of the chapters of this book, it would be the following:

1. Identification of the problem – a particular situation or aspect is described in a way that problems are self-explanatory.
2. Labeling and classification – the situation is analyzed and often a set of causes are presented to the reader, often labeled or classified.
3. List of possible solutions – a list possible solutions is presented to the reader, often as a definition list.
4. Solution details – more details are provided to prove the effectiveness of the solution, often including personal anecdotes.

By doing so, the author makes sure that everything he writes about can be easily understood and accepted, because proven by personal experience.

Contents

Part I: Managing Your Projects

The book starts with a general introduction on Software Quality. If you are new to the subject (and you *shouldn't* be), this is probably one of the best and to-the-point overviews you'll ever find, written by the man who almost came up with the concept.

The second chapter is about planning. Actually, the whole book is about planning at different levels, so no, you should not dismiss this part. *Good* plans are important, and they are your best weapon against management, if you excuse the expression.

Someone may object that if you're working in an *agile* team, you shouldn't spend a lot of energy in long-term planning, but rather focus on dealing with frequent requirement changes and deliver often and regularly. While this can be true, planning is still important: you won't produce any rigorous schedule or design documents, but you still have to be able to provide accurate estimates and very often!

Part II: Managing Your Teams

The second part of the book focuses the *Team*, the people in it, their roles, their responsibility and its leadership. Chapter 3 introduces Tom DeMarco's concept of *Jelled Team*, i.e. a team that is more than the sum of its parts, and is characterized by cohesion, challenging goals, frequent feedback, a common working framework and good communication.

The Holy Grail. The dream of every team leader and its members. The good news is, it can be done. Any team can jell, and teams *like to jell* furthermore, if the proper conditions exist, and the three chapter in this third part will teach you everything from being a good team member to becoming a great team leader.

In many ways, this was my favorite part of the book. It's amazing how a lifetime of experience is distilled in just a few pages. Chapter 5 (Leading and Coaching your Teams) is very, very inspiring and very helpful in understanding how to become a good team leader, how to motivate and involve people, and how to manage them rationally.

The story of Humphrey's high school wrestling coach Umbach is a classic example of a truly dedicated, inspiring, and successful leader:

"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."

Part III: Managing Your Boss

The third part consists of a single chapter: *Negotiation your projects and defending your plans*. It doesn't matter if you want to pretend otherwise: as soon as you become a team leader and you have to deal with management, you'll have to deal with complex internal politics.

This chapter is about learning to be pragmatically diplomatic and deal with management. It's about creating good plans that can survive confrontation with your managers, no matter what their demands are.

There's no silver bullet: I appreciated the honesty of the author when providing solutions. Section 6.6 (What to do when a project is doomed) is an example of this:

You're on a project and it's heading south. While everybody is trying their hardest, and you're doing your level best to help, you can feel it in your bones: the project is doomed to fail. What can you do? You have three choices.

Keep plugging away and hope things will improve.

Look for another job.

Try to fix the problems.

That's right. Look for another job. That almost made me laugh, but it made me understand that in some extreme situation that may just be the best solution.

Part IV: Managing Yourself

The last part of the book is about you. I would probably have moved it earlier on in the book, maybe right after the first part, but it serves as a good ending for the book. Chapter 7 (Taking Control of Your Work) is a must-read for anyone. It teaches you how to manage your working life, from time management (The 18 Hour Work Week) to psychological aspects (What Do You Want From Life?).

Chapter 8 (Learning to Lead), is a nice writeup on the essence of Leadership, and what it means to be a good leader rather than a manager. A great read.

Final Thoughts

Reading certain sections of this book felt a little bit weird at first. TSP, PSP, heavy planning and documents... are they still relevant in a “real world” now dominated by *agile* practices, Scrum, Kanban and similar? Do you really have to provide detailed plans and documentation to convince management?

- You may not have detailed design documents, but you still have user stories.
- You may not be required to plan ahead of 6 months, but you still have to plan frequently and provide accurate estimates.
- You may not be required to trace and track everything you do, but at the very least you have to monitor your *velocity* and produce *burndown charts*.

Yes, you read “PSP” and “TSP” everywhere in the book, but they are just labels. The methodologies and processes may change, but *the principles* will always remain true. This book is about understanding the very essence of management and leadership, and it will remain an invaluable resource for anyone who wants to build a career in the Software Industry.