Food for Thought™

An e-newsletter published by
Software Quality Consulting, Inc.

Dec 2010
Vol. 7 No. 5

Welcome to **Food for Thought**™, an e-newsletter from Software Quality Consulting. I've created free subscriptions for my valued business contacts. If you find this newsletter informative, I encourage you to continue reading. Feel free to pass this newsletter along to colleagues by using this Forward Email link. If you've received this newsletter from a colleague and would like to subscribe, please use this Enter New Subscription link. If you don't wish to receive this newsletter, use the SafeUnSubscribe link at the bottom of this newsletter, and you won't be bothered again.

Your continued feedback on this newsletter is most welcome. Please send your comments to **steve@swqual.com**

In This Issue

In This Month's Topic I pay tribute to the enormous contributions made by Watts Humphrey.

Regular features to look for each month are:

- Monthly Morsels
  Hints, tips, techniques, and references related to this month's topic

This Month's Topic

Watts Humphrey

The Father of Software Quality

Watts Humphrey
1927-2010

The software engineering community recently lost one of its most influential and dynamic leaders. On October 28, 2010, Watts A. Humphrey passed away – he was 83 years old. Watts was an inspiration to a generation of software engineers and software quality professionals. I have long considered Watts a mentor, a role model, and a person with a unique gift. He had the ability to make software engineering understandable to managers and executives. This gift has led to many significant changes in how software development work is managed. I feel privileged to have benefited from his wisdom and experience.

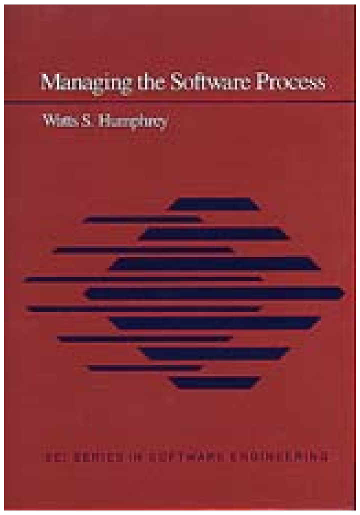
This edition of my e-newsletter is dedicated to Watts and acknowledges the huge impact he has had on software engineering and the software quality profession.

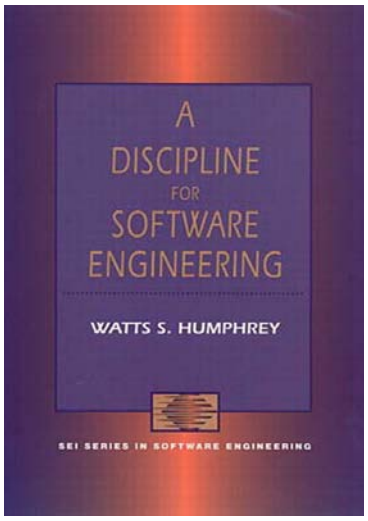
A brief biography…

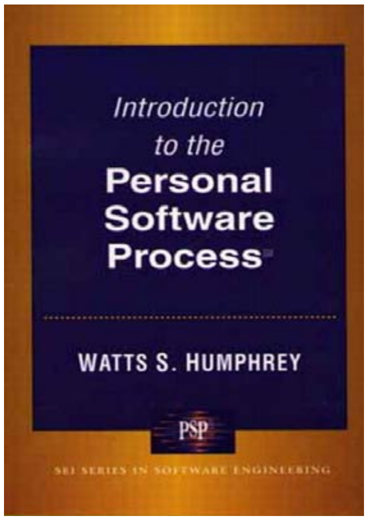
Born in Battle Creek, MI, Watts suffered with dyslexia and had trouble in public school. His family relocated to Connecticut, where schools were better able to help him overcome his dyslexia. He persevered and eventually graduated valedictorian of his high school class.

After serving in the Navy, Watts earned a bachelor's degree in physics from the University of Chicago, studying under Enrico Fermi. He then completed a master's degree in physics from the Illinois Institute of Technology and an MBA degree, with an emphasis on manufacturing, from the University of Chicago. There, he later recalled, professor Judson Neff taught him the three most important things in manufacturing: planning, planning, and planning.

> "He said if you don't plan, you can't run a manufacturing operation," Humphrey explained. "That had an enormous impact on me."


Managing the Software Process
Watts S. Humphrey
SEI SERIES IN SOFTWARE ENGINEERING


A DISCIPLINE FOR SOFTWARE ENGINEERING
WATTS S. HUMPHREY
SEI SERIES IN SOFTWARE ENGINEERING


Introduction to the Personal Software Process
WATTS S. HUMPHREY
PSP
SEI SERIES IN SOFTWARE ENGINEERING

Cost accounting also made an enormous impact on his later work.

> "It's a tremendously powerful field, the whole idea of measurement and precision." [1]

In 1953, he came to Boston and worked at Sylvania Labs designing circuits. He attended a course on the Whirlwind computer at MIT. Wanting to learn more, he inquired about computer courses at Northeastern University. At the time, there were none, so he was asked to create and teach one – which he did.

To prepare this course, he spent weeks at the Harvard and MIT libraries. His class consisted of employees from Honeywell Corporation who spent time actually building computers. Watts said this experience reinforced the philosophy that to manage or teach effectively, you need to respect the knowledge and experience of those who you are managing or teaching.

In 1959, he began a long career at IBM where he worked initially as a hardware architect. He eventually became Vice President of Technical Development, where he oversaw over 4,000 engineers in 15 development labs spread over 7 countries. During his time at IBM, he was influenced by several people, including:

- Fred Brooks (author of The Mythical Man-Month)
- Barry Boehm (author of Software Engineering Economics)
- Michael Fagan (creator of the Formal Inspection process)
- Harlan Mills (creator of Chief Programmer Teams and Cleanroom)
- Gerry Weinberg (author of Psychology of Computer Programming)

When asked about his time at IBM, Watts said:

> "I discovered through this period that hardware management principles, while sound, weren't effective in a software setting. Software is large-scale knowledge work. It's hard to manage people when you don't understand what those people are doing." [1]

In 1986, after retiring from IBM, he joined the newly formed Software Engineering Institute (SEI) at Carnegie Mellon University in Pittsburgh. The SEI was established by the US Dept of Defense to provide guidance to the military services in selecting capable software contractors. [2]

It was at the SEI where he made an "outrageous commitment" to change world of software development by developing sound management principles.

> "Changing the world of anything is an outrageous personal commitment. That's what makes it outrageous. I felt it needed to be done. I knew I couldn't do it alone, and I wanted an environment where I could work with folks and do that," Watts explained in a 2010 interview [1].

Significant contributions:

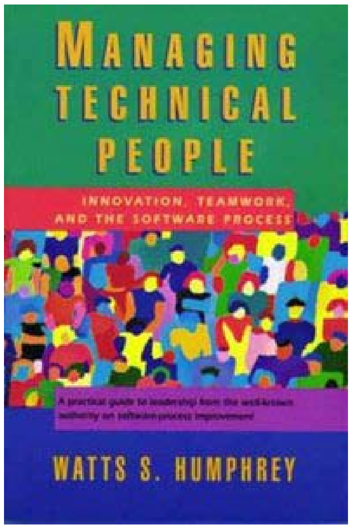
Let's look at just a few of the major contributions Watts has made…

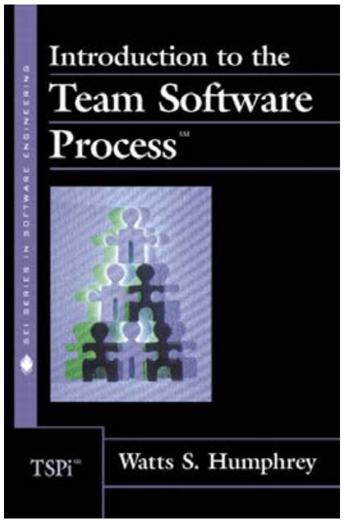
One of his first books – Managing the Software Process, published in 1995, was a direct result of work on the DoD project to provide guidance to DoD in selecting capable software contractors. This book describes "technical and managerial topics that assessments have found most critical for improvement" [3] and has roots in the work of Demming and Juran (statistical process control).

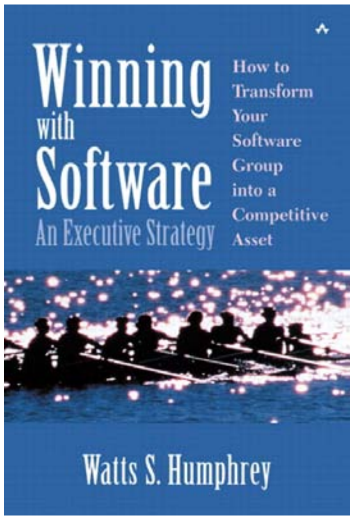
In this book, Watts suggests we assess the software development process by asking three questions:

- How good is my current software process?
- What must I do to improve it?
- Where do I start?

This book also provides basic principles for process assessment. These include:


MANAGING TECHNICAL PEOPLE
INNOVATION, TEAMWORK, AND THE SOFTWARE PROCESS
WATTS S. HUMPHREY




Winning with Software
An Executive Strategy
How to Transform Your Software Group into a Competitive Asset
Watts S. Humphrey

- Major changes to software process must start at the top
- Ultimately, everyone must be involved
- Effective change requires a goal and knowledge of current process
- Change is continuous
- Software process changes will not be retained without conscious effort and periodic reinforcement
- Software process improvement requires investment

In this book, Watts talks about process assessment and process improvement and reinforces the notion of measurement. With regard to process improvement, he states:

"If you don't know where you are a map won't help." [3]

Watts also identifies the key goals of Software Quality Assurance:

- To improve software quality by appropriately monitoring both the software and the development process that produced it

- To ensure full compliance with the established standards and procedures for the software and the software process

- To ensure that inadequacies in product, process, or standards are brought to management's attention so they can be fixed [3]

In his book, Winning with Software: An Executive Strategy, published in 2001, Watts used his unique gift to explain why management techniques used to manage manufacturing are not effective in managing software development work. In this book, he identified several reasons why executives should insist that software quality be measured and managed:

- Poor quality software can cause major property damage and even kill people

- Quality work saves time and money

- If you don't manage software quality, nobody else will

He also identified the five most common causes of project failure:

- Unrealistic schedules

- Inappropriate staffing

- Changing requirements

- Poor quality work

- Believing in magic

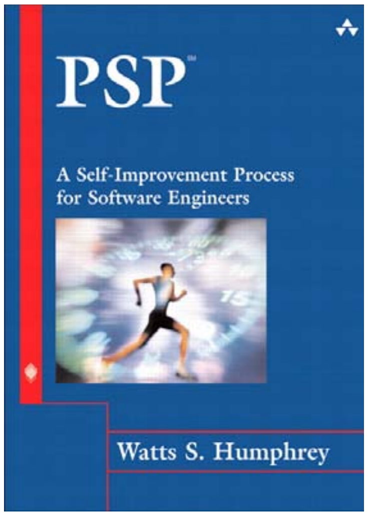
CMM^{sm} and CMMi^{sm}

The development of the CMM^{sm} and CMMi^{sm} represents the most significant advancement in software engineering and software quality in the past 50 years.

While often viewed with skepticism, the CMM^{sm} established a consistent way of assessing (measuring) software development processes. CMM^{sm} assessments provided management with:

- a way to identify what to do and how to do it
- guidance on where to start (Key Process Areas - KPAs)
- focus on data and measurements

The CMM^{sm} and CMMi^{sm} provided ways to make use of data and measurements along with well-known process improvement tools such as Plan-Do-Check-Act to enable



organizations to achieve remarkable improvements in repeatability, consistency, and quality.

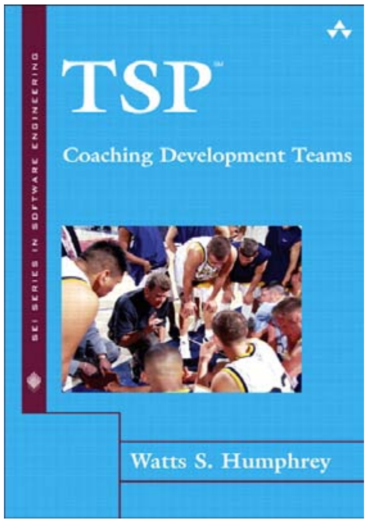
Shortly after the CMM^{sm} and CMMi^{sm} were established, Watts was honored as the first Fellow of the Software Engineering Institute. With this honor, he was now free to investigate any topics that were of interest to him.

What followed is truly remarkable…

The Personal Software Process (PSP)

In his book the Personal Software Process (PSP), Watts undertook to better understand the personal characteristics and behaviors of software engineers that lead to higher quality software. He did this in a unique way – he gave himself a challenge – to develop software and document the practices he determined would lead to better quality. This book and the accompanying training courses Watts developed and taught have helped improve software engineering skills and software quality throughout the world.

The Team Software Process (TSP)


TSP
Coaching Development Teams
SEI SERIES IN SOFTWARE ENGINEERING
Watts S. Humphrey

After creating the PSP, Watts recognized that the PSP was not sufficient since software was developed by teams not by individuals. Watts then developed the Team Software Process (TSP) as an outgrowth of the PSP.

The Quality Attitude

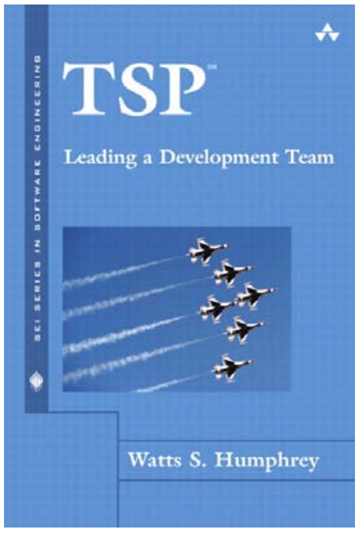
After creating the PSP and TSP, Watts continued his work at the SEI. He began writing a series of newsletters called *Watts New.*

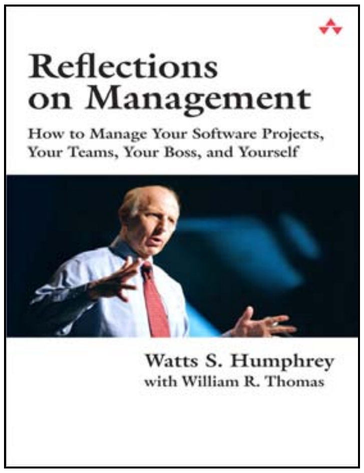
> In an attempt to make Watt's wisdom accessible to as wide an audience as possible, the SEI has published a compilation of many of his articles originally published in the SEI Newsletter *Watts New.*

One of my favorite issues of this newsletter was one titled The Quality Attitude [6]. In this article, Watts states the importance of having an attitude that reinforces quality. He said:

- Every developer must view quality as a personal responsibility and strive to make every product element defect free. [6]

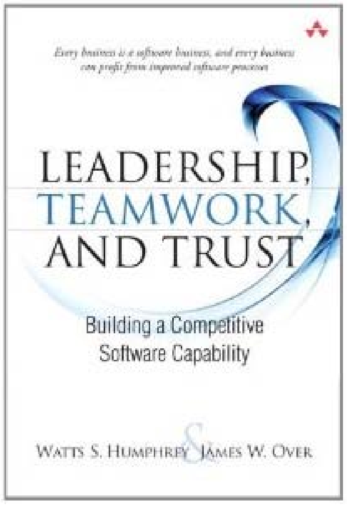
This article also introduces the notion of the **testing footprint:**


TSP
Leading a Development Team
SEI SERIES IN SOFTWARE ENGINEERING
Watts S. Humphrey


Reflections on Management

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

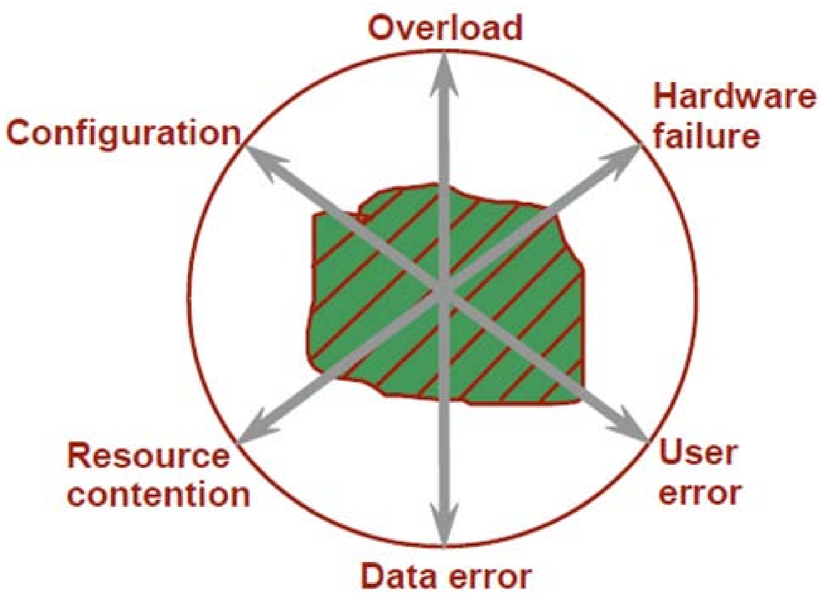
Watts S. Humphrey with William R. Thomas


Every business is a software business, and every business can profit from improved software processes

LEADERSHIP, TEAMWORK, AND TRUST

Building a Competitive Software Capability

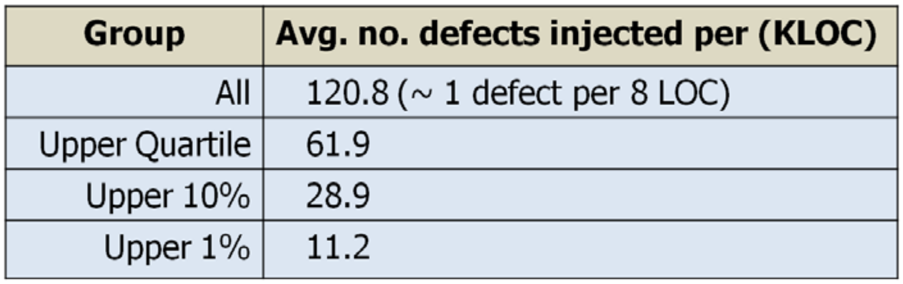
Watts S. Humphrey & James W. Over



What he pointed out was that the shaded area is the area covered by system testing. However, once software is released, customers often use software in ways that were not covered by system testing. When this happens, customers find defects not found in system testing. Knowing your **testing footprint** and how it differs from how your customers use your software is critical to reducing the number of defects customers find.

Watts reported that some development groups are now producing reasonably large-scale software products that have had no defects found by their users. While these products actually have latent defects, for all practical purposes, they are defect free. Today, Watts reported that there are but a few development teams can consistently produce such software.

Watts also collected some defect data from a group of 810 experienced software engineers who developed over 8,000 programs. An analysis of the date showed that:



Group	Avg. no. defects injected per (KLOC)
All	120.8 (~ 1 defect per 8 LOC)
Upper Quartile	61.9
Upper 10%	28.9
Upper 1%	11.2

What this analysis shows is that, on average, experienced software engineers inject about 1 defect for every 8 lines of code written. In many cases, about 95% of these defects are found prior to release. As a result, software is released with some known defects and a significant number of unknown defects.

Given this, let's look at a typical example consisting of one million lines of code:

- One million Lines of Code (LOC) = 1,000 KLOC

  If we assume average defect injection rate of 120 defects/KLOC, that would result in about 120,000 defects injected. If we also assume that 95% are found, that results in 114,000 defects found

- Number of Unknown defects = defects injected – defects found

= (120,000 – 114,000)
= 6,000 defects remaining

This means that there could be as many as 6,000 defects were not found in the released software. Watts recognized this and observed:

> "So, the first required attitude change for software professionals and their managers is to accept the fact that testing alone will not produce quality software systems. Also, since defective software cannot be secure, testing alone won't produce secure systems either." [6]

Watts Humphrey's Legacy

There have been only a handful of people who have made significant contributes to the state of the art in software engineering and software quality assurance. In my opinion, the contributions of Watts Humphrey are far and away the most significant. He has managed to achieve his "outrageous commitment" to change the way software development projects are managed by developing and applying sound management principles.

We will forever be indebted to his accomplishments, we will miss his wisdom and wit and his ability to change the course of software engineering.

Monthly Morsels

Every month in this space, you'll find additional information related to this month's topic.

1. An Interview with Watts Humphrey by Grady Booch, published at InformIT Website.

2. "A method for assessing the software engineering capability of contractors," SEI Technical Report SEI-87-TR-23, September 1987

3. Humphrey, W., Managing the Software Process, Addison-Wesley, 1995.

4. Humphrey, W., Winning with Software: An Executive Strategy, Addison-Wesley, 2002

5. Humphrey, W., A Discipline for Software Engineering, Addison-Wesley, 1995.

6. Humphrey, W., "The Quality Attitude", news@sei, Number 3, 2004

The Watts Collection

The SEI has put together a comprehensive collection of the work that Watts has published called The Watts Collection. I urge you to take advantage of this resource as it contains the wisdom of one of the greatest software engineering minds of our time.

Complete list of books by Watts:

Leadership, Teamwork, and Trust: Building a Competitive Software Capability (SEI Series in Software Engineering), Addison-Wesley, to be published 2011.

Reflections on Management: How to Manage Your Software Projects, Your Teams, Your Boss, and Yourself. Addison-Wesley, Reading, MA, 2010

TSP, Coaching Development Teams, Addison-Wesley, Reading, MA, 2006

TSP, Leading a Development Team, Addison-Wesley, Reading, MA, 2006

PSP, A Self-Improvement Process for Software Engineers, Addison-Wesley, Reading, MA, 2001

Winning with Software: An Executive Strategy, Addison-Wesley, Reading, MA, 2001

Introduction to the Team Software Process, Addison-Wesley, Reading, MA, 1999

Managing Technical People - Innovation, Teamwork and Software Process, Addison-Wesley, Reading, MA, 1997

Introduction to the Personal Software Process, Addison-Wesley , Reading, MA, 1996

A Discipline for Software Engineering. Addison-Wesley, Reading, MA, 1995

Managing the Software Process, Addison-Wesley, Reading, MA, 1989

Switching Circuits: With Computer Applications, McGraw-Hill, 1958

About SQC

Software Quality Consulting provides a full-range of software engineering services for safety-critical industries and mission-critical projects. Our goal is to help create safety-critical and mission-critical software that meets our client's needs, complies with all applicable standards and regulations, with the highest level of quality possible, and in the most cost-effective and timely manner possible.

To learn more about how we can help your organization, visit our web site or send us an email.