软件**制胜**之道

Winning with Software
An Executive Strategy

瓦茨·S·汉弗莱 [Watts S. Humphrey] 著
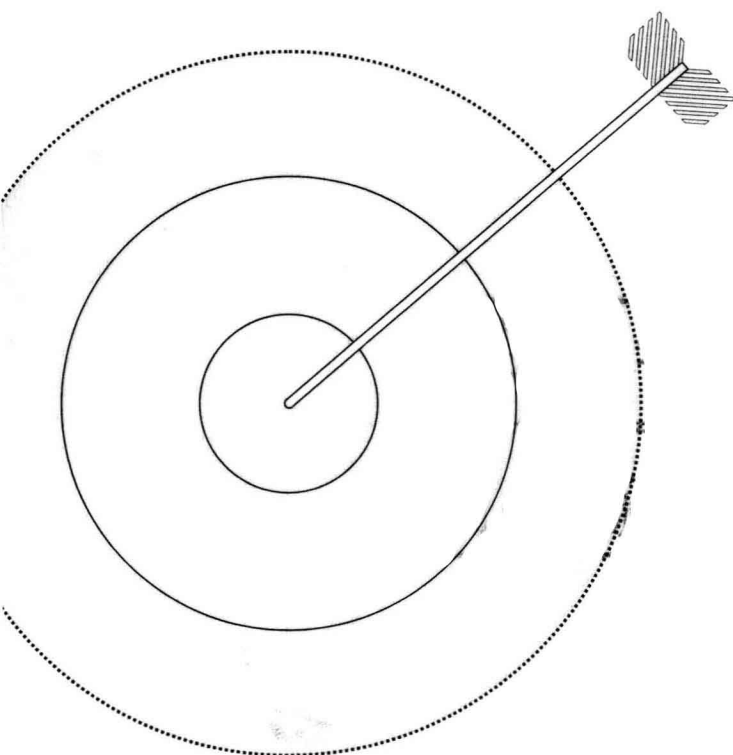
软件制胜之道

Winning with Software
An Executive Strategy

瓦茨·S·汉弗莱 [Watts S. Humphrey] 著

（京）新登字 158 号

内 容 简 介

本书是为高级经理们写的，旨在提高他们所领导的软件小组的经营业绩。

本书描述了软件对业务的影响，综述了一些最常见的软件问题及其原因。还说明了必须领导所在企业进行转变，以及在企业中进行潜在的软件投资所需的步骤。在书末还包含了 5 个建立 TSP 和 PSP 的方法。

出 版 说 明

1984 年，美国国防部出资在卡内基·梅隆大学设立软件工程研究所(Software Engineering Institute，简称 SEI)。SEI 于 1986 年开始研究软件过程能力成熟度模型（Capacity Maturity Model，CMM），1991 年正式推出了 CMM 1.0 版，1993 年推出 CMM 1.1 版。此后，SEI 还完成了能力成熟度模型集成（Capability Maturity Model Integration，简称 CMMI）。目前，CMM 2.0 版已经推出。

CMM 自问世以来备受关注，在一些发达国家和地区得到了广泛应用，成为衡量软件公司软件开发管理水平的重要参考因素，并成为软件过程改进的事实标准。CMM 目前代表着软件发展的一种思路，一种提高软件过程能力的途径。它为软件行业的发展提供了一个良好的框架，是软件过程能力提高的有用工具。

SEI 十几年的研究过程和成果，都浓缩在由 SEI 参与研究工作的资深专家亲自撰写的 SEI 软件工程丛书（SEI Series In Software Engineering）中。

为增强我国软件企业的竞争力，提高国产软件的水平，经三联四方工作室和清华大学出版社共同策划，全面引进了这套丛书，分批影印和翻译出版，这套丛书采取开放式出版，不断改进，不断出版，旨在满足国内软件界人士学习原版软件工程高级教程的愿望。

前　　言

  每当我向高级经理们提及软件时，都会引起很大的反响。大多数是关于他们所受的挫折。他们抱怨责任不明确、质量问题以及令人感到不快的意外事件，而这些问题几乎与软件无关。软件确实也有问题，但是这些问题业已得到解决。没有人认识到软件的商业机遇，他们把软件看成是无法避免的灾祸——某种应尽量避免的东西。虽然大多数经理承认他们业务中的软件部分得到了快速增长，但是他们从来没有把它当作一种财富，更不用说商业机遇了。

  只要采纳本书讲述的方法，每个组织都可以改变它们的软件团队。波音公司的第一个团队使测试时间减少了 94%；某空军小组使他们的生产率翻了一番；Teradyne 项目组开发出了一种大型而无缺陷的产品。这些组织以及其他一些组织，都通过软件取得了显著的成效，这是因为在项目一开始时，组织的管理层就将重点放在软件所带来的机遇上。

威胁还是机遇

  软件真是一种不可思议的技术。软件的生产成本为零，可以在顷刻间分布于世界各地；软件既不会耗尽，也不会变坏；软件几乎是实现任何复杂功能的最经济灵活的方法。现在，在工程和科学的几乎任何领域，专家们都要将大半的时间花在软件的使用、开发、增强和维护上。无论如何，软件是一件大事。

  为了明确表示软件带来的机遇，考虑一下与之相对的另一方面：潜在的威胁。拿制造业来讲。假设你的主要竞争对手掌握了这样一种技术，它使制造成本下降了一半，在货物提交上也很及时，并且它们提供的产品经久耐用、不易损坏。如果你不能及时投资该项技术，无疑你将陷入困境。相反地，考虑一下若你的企业掌握了该项技术而你的竞争对手没有，会有什么机遇呢？虽然确切地讲软件是这样一种技术，但是几乎没有组织把它看成一种机遇或者一种威胁。主要原因在于很多经理认

为他们的组织不会涉及太多的软件工作，而在那些认识到这一点的经理中，几乎无人有足够的软件知识或者经验去欣赏软件为他们的商业所作出的贡献。他们一旦认为软件问题得到了控制，他们就会尽力避开这个主题。

越来越多的经理认识到软件是一笔强大的企业财富。然而，同样是这些经理们，他们还发现要步入现代工程软件的殿堂，就必须进行一次组织转变。更有甚者，他们已经发现，他们必须身体力行地领导这种转变。这不是一种简单的变化。与所有转变一样，这种转变不仅仅是告知人们应该做些什么。为了说明这种转变所涉及的工作，最好的办法是说明本书所描述的方法是如何创建的。

转变之旅

我在IBM公司工作的27年间，其中一项工作是担当程序设计主管。我要监督15个实验室和7个国家中的4000位软件专家。在4年时间内，我们把这个组织从濒临混乱的边缘带回到有条不紊的经营状态。第一步是制定有效的工程管理准则，并且要求员工必须遵守这些准则。为了确保这些原则为大家所理解，我们派送1000个管理员参加为期1周的培训课程。效果非常显著！该组织之前从来没有按时提交过一项产品，在管理员们都经过了培训并且遵循所要求的规划和责任过程之后，该组织在之后的2年半时间内没有错过一次机会。

1986年，我从IBM公司退休，那时我已经从总体上考察了软件业。显然，软件是一种关键性技术，但是我们也可以清楚地看到，拙劣的软件工程习惯既严重地制约了美国经济的发展，也制约了一般社会的经济发展。我当时提出了所谓的"难以容忍的义务（outrageous commitment）"。我的义务是改变软件业，目标是把我在IBM中经历的那些非常成功的做法和原则带给整个软件业。

当我从IBM退休时，我加入了卡内基·梅隆大学的软件工程研究所（Software Engineering Institute，SEI），并担任软件过程项目的主管。当时美国国防部刚刚成立SEI，旨在改进软件习

惯。该任务与我提出的"难以容忍的义务"完全一致，它旨在让所有软件专家以及他们的管理员们规划和跟踪他们的工作，使用最好的技术方法，以及度量和管理这种工作的质量。我深信，如果他们这样做了，效果必定是很显著的。

不久以后，与一批志同道合的 SEI 专家们一起，我们开发了能力成熟度模型（Capability Maturity Model，CMM），以指导组织采纳健全的管理准则。CMM 相当有效，现在已经为世界各地的上千家组织采纳。现在，CMM 已经成为国际标准，美国政府所属的很多部门都使用它，用来衡量内部软件工作，以及评估和监督合约商的工作。

虽然 CMM 工作取得了非常成功，并且将持续下去，但是我很快就看出了存在的问题。CMM 提供了优秀的管理指南，但是它主要影响管理员以及他们的技术人员。CMM 不直接影响工程师的工作，因此工程师及其团队仍然处于艰难的奋斗之中。毫无疑问，更好的管理会起到一定的辅助作用，但是我不久就认识到：除非我们改变了软件专家们本身的做法，否则我们永远不能实现一种真正的专家级软件工程能力。因此，下一个急待解决的问题是激发工程小组实现这个目标。我要他们知道最佳的方法，但是我还要他们每天真正地实践这些方法。我为此研究的方法称为个人软件过程（Personal Software Process，PSP）和团队软件过程（Team Software Process，TSP）。这些方法的开发在第 6 章和第 7 章进行了叙述。

本书的其余部分讲述了如何投资这些方法。这些方法正在为其他组织产生显著的效果，你应当把这看成既是一种威胁又是一种机遇。正如 Teradyne 公司的一位工程管理人员告诉我的那样："在掌握了 TSP 之后，我们已遥遥领先于竞争对手，无人能够赶上我们"。

根据我的经验，采用了 TSP 的项目可以使他们的生产率翻一番，并且可以使产品质量提高一个数量级。所需的投资主要用于培训和指导，而这些投资通常可以在 12 到 18 个月内得到回报。在团队经过培训并且获得某些经验之后，就不用为 TSP过程投入太多的管理费用了。然而，经理的领导是必须的，以

使你的职员得到正确的培训，并在相当长的时间内支持他们，使他们始终如一地获取实践这些方法的经验。

为什么要阅读本书

本书是为高级经理而写的，旨在提高他们领导的软件小组的经营业绩。书中所称的"你"，特指 CEO、副总裁和部门总经理。本书资料适用于那些肩负企业收益的经理，以及那些管理大部分组织资源的经理。总之，很多资料都带有某种商业色彩，同时使其中所含的技术术语最少。然而，我确实较为深入地研究了技术资料，这一点可能会出乎许多经理们的预料。

我之所以这么做的原因有三点。第一，经理们往往对那些绘声绘色的讲述持怀疑态度，希望更深入地了解故事背后的内涵。PSP 和 TSP 有着丰富的内涵，本书中也讲了很多，足以使你找到该主题的感觉。第二，软件是一种非常吸引人的业务，你可能希望更深入地研究该主题中的某些要素。第三，高级和中级经理可以拿这一资料作为指南，提高他们组织的软件能力。

本书的组织和内容

本书描述软件业务。不管你是否已经意识到，你总是置身于软件业务中，并且软件小组的业绩对经营业绩有着重大的影响。我首先描述了软件对业务的影响，然后综述了一些最常见的软件问题及其原因。最后，描述了你必须领导的转变，以及在你的企业中进行潜在的软件投资所需的举动。

在本书的结尾部分，我包含了 5 个关于建立 TSP 和 PSP 方法，并使这些方法成为日常业务中的标准组成部分。附录 F 对你可以从这些转变中取得的投资回报进行了简单的经济分析。

…… ……

——瓦茨·S·汉弗莱

Winning

with

Software

An Executive Strategy

Watts S. Humphrey

✦Addison-Wesley

Boston • San Francisco • New York • Toronto • Montreal
London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

Carnegie Mellon
Software Engineering Institute

The SEI Series in Software Engineering

Winning
with
Software

For my family.
My work has been rewarding,
but nothing beats coming home.

Preface

When I mention software to senior executives, I get lots of reactions. Most are frustrated. They complain about missed commitments, quality problems, and unpleasant surprises. Others have been less closely involved. Software was a problem, but those problems have been handled. No one mentions the business opportunities of software. They think of software as a necessary evil—something to be avoided if possible. While most executives would agree that the software part of their business is growing very quickly, they never think of it as an asset or an opportunity.

By using the methods described in this book, organizations have transformed their software groups. The first Boeing team cut test time by 94%; an air force group doubled productivity; a Teradyne project delivered a large defect-free product. These and other organizations are getting outstanding results. However, they all started with a management focus on the opportunities with software.

THREAT OR OPPORTUNITY

Software is a truly incredible technology. It has a zero production cost, can be distributed worldwide in seconds, does not wear out or deteriorate, and is the most economical and flexible

way to implement almost any complex function. In just about any field of engineering or science, more than half a typical professional's time is now spent in using, developing, enhancing, or maintaining software. By any measure, software is big business.

To visualize the opportunities with software, consider the inverse: potential threats. Take manufacturing, for example. Suppose your leading competitor mastered a technology that cut manufacturing costs in half, eliminated distribution delays, and provided products that never wore out or deteriorated. If you did not quickly capitalize on that technology, you would almost certainly be in trouble. Conversely, think of the opportunities if your organization mastered this technology and your competitors did not. While software is precisely such a technology, few organizations see it as either an opportunity or a threat. The principal reason is that many executives don't think their organizations do much software work and, of those that do, few have enough software knowledge or experience to appreciate how it contributes to their business. Once they think the software problems are under control, they do their best to avoid the subject.

A growing number of executives have found that software is a powerful business asset. However, these same executives have also found that moving into the modern world of engineered software requires an organizational transformation. What is more, they have discovered that they must personally lead this transformation. It is not a simple change and, like all changes, this transformation involves more than just telling people what to do. The best way to explain what is involved is to tell the story of how the methods described in this book were created.

THE TRANSFORMATION JOURNEY

During my 27 years with IBM, one of my jobs was director of programming. I supervised 4,000 software professionals in 15

laboratories and 7 countries. In four years, we took this organization from the brink of chaos to a sound, businesslike operation. The first step was to establish effective engineering and management practices and to require that these practices be followed. To ensure that these practices were understood, we sent 1,000 managers to a one-week training course. The results were extraordinary. This organization had never before delivered a product on time. Once the managers were all trained and following a disciplined planning and commitment process, the organization did not miss a single commitment for the next two and a half years.

When I retired from IBM, in 1986, I looked at the software industry in general. It was obvious that software was a crucial technology, but it was also clear that the poor state of software engineering practice seriously constrained both the U.S. economy and society in general. I made what I called an "outrageous commitment." My commitment was to transform the world of software. The objective was to bring to the world in general the practices and principles that I had found so successful at IBM.

On my retirement from IBM, I joined the Software Engineering Institute (SEI) at Carnegie Mellon University and was made director of the software process program. The SEI had just been established by the U.S. Department of Defense to improve the state of software practice. This mission was completely consistent with my "outrageous commitment," which was to get 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. I was convinced that if they did, the results would be extraordinary.

Together with a small team of like-minded SEI professionals, we soon developed the Capability Maturity Model (CMM)® to

---

® Registered in the U.S. Patent and Trademark Office.

guide organizations in adopting sound management practices. The CMM has been highly effective and is used by thousands of organizations throughout the world. The CMM is now an international standard, and it is used by many branches of the U.S. government to evaluate internal software work and to assess and oversee the work of their contractors.

Although the CMM effort was and continues to be highly successful, I soon saw problems. The CMM provides excellent management guidance, but its principal impact is on the managers and their technical staffs. The CMM does not directly affect the work of the engineers, and the engineers and their teams were still struggling. There is no question that better management helps, but I soon realized that until we changed the practices of the software professionals themselves, we could never achieve a truly expert software engineering capability. Therefore, the next challenge was to motivate engineering groups to do just that. I wanted them to know the best methods, but I also wanted them to actually practice these methods every day. The techniques I developed to do this are called the Personal Software Process (PSP)^{SM} and the Team Software Process (TSP)^{SM}. The development of these methods is described in Chapters 6 and 7.

The story of how your organization can capitalize on these methods is told in the rest of this book. These methods are producing extraordinary results for other organizations, and you can view this as either a threat or an opportunity. As an engineering manager at Teradyne told me, "With the TSP, we're so far ahead of the competition that nobody will ever catch us."

It has been my experience that projects that use the TSP can double their productivity and improve product quality by an

---

^{SM} Personal Software Process, PSP, Team Software Process, and TSP are service marks of Carnegie Mellon University.

order of magnitude. The investment required is predominantly training and mentoring costs and these costs typically are recovered within 12 to 18 months. Once teams have been trained and acquire some experience, there is no significant overhead to the TSP process. However, executive leadership is required to get your people trained properly and to support them long enough to gain the experience to practice these methods consistently.

WHY YOU SHOULD READ THIS BOOK

This book is written for senior executives who want to improve the business performance of their software groups. When I use the word *you,* I am talking to CEOs, vice presidents, and division general managers. The message of the book is designed for executives who have profit responsibility and who directly control a substantial portion of their organization's resources. As a result, much of the material has a business slant and contains a minimum of technical jargon. However, I do delve a little more deeply into the technical material than many executives might expect.

I do this for three reasons. First, executives often are suspicious of impressive presentations and like to dig a little deeper to see if there is substance behind the story. There is plenty of substance to the PSP and TSP, and I have included enough to give you a feel for the subject. Second, software is a fascinating business and you may wish to explore some elements of the subject more deeply. Third, senior and mid-level managers can use this material to guide them in improving their organization's software capability.

BOOK ORGANIZATION AND CONTENTS

In this book, I describe the software business. Whether or not you know it, you are in the software business and the performance of