系统与决策丛书/杨建梅 主编

软件过程改进的复杂性工作程序研究

—— SW-CMM实施的方法论

万江平 杨建梅 著

科学出版社
www.sciencep.com

系统与决策丛书/杨建梅主编

软件过程改进的复杂性工作程序研究

——SW-CMM 实施的方法论

万江平　杨建梅　著

本书受国家教育部博士点专项科研基金（20030561024）
及广东省人文社科基地华南理工大学新型工业化
发展研究所资助出版
本书是国家自然科学基金项目（项目批准号 70471091）的
前期研究成果

科学出版社

北京

内 容 简 介

本书在研究软件的本质与沃菲尔德提出的复杂性内在联系的基础上，对软件质量模型和软件能力成熟度模型进行了深入分析，说明了软件过程改进是软件质量管理的必由之路。接着建立了基于沃菲尔德复杂性理论的软件过程改进理论，包括软件过程的价值链分析、软件过程改进的复杂性命题及其合理性的检验等。在此基础上，设计了软件过程改进的沃菲尔德复杂性工作程序（方法论），包括软件过程改进的认知障碍界定和分类、绩效模型、过程模型设计以及实施所需要的知识等。最后，以问卷调查和案例研究来验证理论、方法论的合理性以及有效性。

本书可供软件企业领导、软件企业经营管理人员、政府部门中软件产业管理干部、科研机构中软件管理和相关技术人员阅读使用，也可供高等院校，尤其是软件学院、计算机软件专业和工程管理类专业的研究生和本科高年级学生使用。

图书在版编目（CIP）数据

软件过程改进的复杂性工作程序研究：SW-CMM 实施的方法论/万江平，杨建梅著 .—北京：科学出版社，2004

系统与决策丛书/杨建梅主编

ISBN 7-03-013646-2

Ⅰ.软… Ⅱ.①万… ②杨… Ⅲ.软件工程 Ⅳ.TP311.5

中国版本图书馆 CIP 数据核字（2004）第 065728 号

责任编辑：陈 亮/责任校对：包志虹

责任印制：安春生/封面设计：耕者设计工作室

科学出版社 出版

北京东黄城根北街16号

邮政编码：100717

http://www.[illegible]ep.com.cn

印刷

科学出版社发行 各地新华书店经销

2004 年 9 月第 一 版 开本：B5（720×1000）

2004 年 9 月第一次印刷 印张：15 1/2

印数：1—2 000 字数：283 000

定价：28.00 元

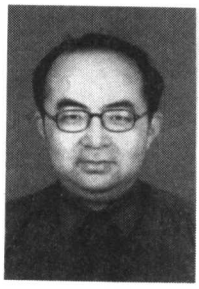
（如有印装质量问题，我社负责调换〈新欣〉）

作者简介

杨建梅　女，1946年生，陕西人。博士，教授，博士生导师，管理学院系统与管理决策研究中心主任。获广东省第四届丁颖科技荣誉奖，享受国务院政府特殊津贴。

主要从事系统科学、产业经济、管理科学等学科交叉领域的教学与研究。理论工作有：提出了产业结构系统工程的研究框架、利益协调软系统方法论、企业战略管理的系统方法论、企业知识管理的系统方法论；另外，对西蒙复杂性构造理论在产业结构中的应用、沃菲尔德管理复杂性理论在软件过程改进及产业政策中的应用、切克兰德人类活动系统在企业中的应用途径也做出了创新性工作。

先后主持过国家自然科学基金、国家社会科学基金、国家教育部博士点基金、人文社会科学基金及广东省自然科学基金等项目的研究，主持的国家自然科学基金被评为优秀研究成果。已在 *IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS* 等国内外著名学术刊物及国际会议会议录上发表论文近百篇。专著《区域产业结构系统工程——理论和实践》1995年获国家教育委员会首届人文社科优秀研究成果二等奖等奖项，还获得过第六届世界经济计量大会中国投稿者前五名资助等科研奖励，入选 Marquis Who's Who in the World。



万江平　男，1964年7月生于江西南昌。博士，国际知识与系统科学协会会员，华南理工大学计算机学院硕士生导师（2001～2003年），华南理工大学工商管理学院副教授、硕士生导师，IBM公司认证专家，中国高级程序员。

在复杂性科学与软件质量管理、知识系统与电子商务、美国软件工程研究所（SEI）的能力成熟度模型（SW-CMM）、复杂性科学中的沃菲尔德学派以及IBM公司的电子商务解决方案等方面有较深的理解和研究。

已在国内外发表论文50余篇，其中6篇被ISTP检索所引用。1998年提议的《新世纪对我国计算机教育的挑战》获九三学社广东省委参政议政二等奖。2001年7月在香港中文大学获首届“两岸三地博士工作坊博士论文中期交流学术会议”决策与经济组最佳论文计划书大奖。作为主要成员参加过教育部博士点基金、国家自然科学基金、广东省重点项目各1项。

《系统与决策丛书》序言

本丛书之所以取名“系统与决策”，意喻本丛书由方法的“横维”与应用的“纵维”所形成的T形结构的具体含义。其中，“系统”一词，主要指系统方法论，包括系统科学的前沿复杂性科学中的方法论；“决策”一词既指这些方法论的应用范围为管理领域，也指具体的决策方法。我们打算近两年先出版软件过程改进的复杂性工作程序、企业知识管理的系统方法论、企业战略管理的系统方法论以及利益协调软系统方法论等方法论方面的著作，以后希望能出版有关决策方法，主要是博弈论在经济管理中的应用以及行为决策方面的专著。

沃菲尔德提出了“基础——理论——方法论——应用”的科学论域体系。这里的“基础”不是指哲学，而是指科学的普遍前提在某个门类科学具体化后所形成的基本概念等要素，其作用是指导“理论”，而“理论”又提供概念之间关系的定律等以指导“方法论”，“方法论”提供程序以指导“应用”，“应用”反过来又影响“基础”。我们认为，哲学信仰决定了知识工作者构建何种类型的理论，应该作为一层引入；而“方法论”一词最初指一个人追踪另一个人的路径，后来演变为做事情的原则或程序，与具体的方法不同，所以应将方法从方法论中剥离，凸显为新的一层；另外，沃菲尔德的具体科学门类的“基础”可以合并进这个门类相应的“理论”层次，这样就可以得到我们的“哲学信仰——理论——方法论——方法——应用”的知识论域体系，其所以称为知识论域体系，是因为哲学不属于科学的范畴。这个体系中各层之间的支配关系仍然与沃菲尔德科学论域体系中的相同，这样，由于本丛书的系统方法论与决策方法，分别属于方法论层次与方法层次，从而系统方法论必然地决定着何种决策方法的采用。

沃菲尔德的复杂性工作程序属方法论层次，其上一层的“理论”层面，包括复杂性概念、复杂性的定律、分类及指数等内容；其下一层则包括诸如著名的解释结构模型、名义小组方法等众多的管理复杂性方法。勿庸置疑，软件产品比较复杂，应该在复杂性科学的指导下，通过过程改进来保证其质量。开始是直觉告诉我们，可以用沃菲尔德的“管理”复杂性科学去摸索软件过程改进的方向，后来在探讨中我们发现，软件工程界提出的软件产品的本质特点，恰好与沃菲尔德的认知复杂性与情景复杂性的概念有着直接的对应关系，从而用沃菲尔德的复杂性工作程序来研究软件过程的改进就是“正当”的了。

直接起源于工程学科的最著名的系统方法论是系统工程，Hull给出的逻辑步骤是：问题定义——确定目标——系统综合——系统分析——系统评价——系统

创建——系统实施，其本质是一个达到已给目标的最优方案的选择过程，所以，系统工程方法论决定了它与最优化决策方法具有必然的联系。1969 年系统工程在阿波罗登月中的巨大成功，导致了一场将其应用于社会经济问题中去的系统运动，但结果却是失败的。通过深刻的反思，Checkland 指出：系统工程是处理被设计系统问题的方法论，而社会经济问题是人类活动系统中的问题，应该用所谓的软系统方法论去处理。于是可以推出，系统工程方法论的“理论”是被设计的工程系统的各种理论，尤其是控制理论、信息论等，而其哲学信仰可追溯到 Plato 的“设计图”的理性主义。

Checkland 软系统方法论的逻辑步骤是：感知问题情景——表达问题情景——相关系统的根定义——相关系统的概念模型——概念模型与对问题感知的比较——寻找期望与可行的变革——行动以改善问题情景，其本质是一个学习过程，所以软系统方法论使用比较、讨论的学习方法。Checkland 强调人类活动系统是通过感知得到的，对同一个问题情景，每个人都会有自己的不同的感知。由此可知，软系统方法论的“理论”是 Weiber 的社会学理论，其哲学信仰是 Husserl 的现象学。

利益协调软系统方法论是对 Checkland 软系统方法论的发展，用于处理利益冲突明显的所谓软人类活动系统中的问题。它以利益协调过程代替单纯的学习过程，其使用的方法是博弈论等群决策的方法。

Flood 与 Jackson 将众多的、分别基于自然-实证主义的系统方法论与基于人-文化主义的系统方法论，根据系统隐喻加以分类式集成，形成了系统方法论的系统体系，以及使用这个体系的全面系统干预元方法论。这些系统隐喻有包括团队、联盟、监狱隐喻的政治隐喻，以及机器、有机体、大脑、文化隐喻等。全面系统干预元方法论的“理论”是评论式系统思考的多元主义、社会学意识以及人类解放的理论，其哲学信仰是 Habermas 的批判诠释学。

通过政治隐喻，我们将利益协调软系统方法论纳入系统方法论的系统之中。在扩充了的系统方法论体系的基础上，结合企业战略研究的九大学派，提出了企业战略管理的系统方法论；结合企业知识管理本身的方法论，提出企业知识管理的系统方法论。这些企业管理系统方法论的提出，不仅对企业管理有用，而且也为系统方法论摆脱空洞名声找到了一个“与具体学科本身的方法论相结合”的出路。

本丛书即将出版的几本书是我们研究团队共同完成的成果。作为探索性的努力，错漏之处在所难免，诚挚地希望读者不吝赐教。

杨建梅

2004 年 8 月于广州

Preface

Technology almost always leads science. So said Sir Geoffrey Vickers, who used as an example the famous Japanese swords dating back to the 12th century, being of an outstanding metallurgical nature, even though metallurgy was not known at the time.

The Roman system for water conduction, at it's peak at around 300 A. D., consisted of 11 aqueducts supplied the city with approximately 1, 170. 000 cubic meters of water per day. About 1400 years elapse before Daniel Bernoulli discovered the laws of fluid flow in pipes.

The computer software field could have exhibited a departure from this technology-science relationship, since Augustus De Morgan discovered, in 1847, the theory of relations, and Boole published his algebra of propositions in the same year. And while the computer software field clearly rests, in its scientific base, upon the mathematics of logic, the software market developed so rapidly that it is no stretch to say that the scientific base remains to be fully acknowledged even today; having constantly been overlooked in the technology. A technology that keeps moving without regard to its scientific base is constantly a candidate for study.

Fortunately Wan Jiangping and his doctoral adviser Yang Jianmei have, almost alone in the modern software literature, managed to sort out the essential scientific base and attach it to the business environment; something that is long overdue. The full implications of this work remain to be seen over the next decade, but an improvement in the quality of software design on the part of those organizations who take advantage of it seems to be inevitable.

It is my pleasure to have been invited to write a preface for this work.

John N. Warfield, Ph. D.
Palm Harbor, Florida, USA
March 19, 2004

序

计算机软件对于信息技术的发展越来越显出其重要性，而软件质量又越来越影响其广泛应用和迅速发展。如何改进软件过程已成为一个亟待解决的问题，而软件过程无疑又是一个极为复杂的过程。

尽管国内外有不少人也在从事相关的研究，但是本书作者从系统科学的角度，特别是从复杂性方面来描述和解决与软件工程相关的问题。作者选用了美国沃菲尔德提出的复杂性理论来描述软件工程的复杂性。在软件开发过程中提出了五个命题，在软件支持过程中提出了七个命题，从而进一步提出了软件过程改进的复杂性工作程序。作者在撰写本书时既能站在当代新兴学科（如系统科学和复杂性研究、认知科学等）的角度，同时又注意了实证研究，利用了他们的理论研究成果对美国微软公司团队成功法则和印度 Infosys 公司软件项目管理进行了科学分析。作者还广泛调查了广东地区的一些信息企业，并以美资旭电（深圳）科技有限公司资讯科技和系统服务部软件项目管理和广州市灵狐系统工程公司软件项目管理为对象进行了深入的案例研究。

总之，该书理论研究和实证研究相结合，对于从事软件工程开发的工作者无疑是一本值得阅读的好书。

顾基发

2004 年 5 月于北京

前　言

随着信息技术的发展和网络新经济时代的到来，计算机软件对经济发展起着越来越明显的驱动作用。质量是软件的核心，而软件是一种逻辑产品，软件质量改进既困难又复杂。为降低软件质量改进的困难程度及其复杂性，计算机软件界进行了大量的研究，问题解决的切入点正逐渐集中到软件生产的软件过程中。IBM OS/360 总设计师布鲁克斯（Brooks）博士根据实践经验指出了软件的四个本质：复杂性、一致性、可变性和可视性。通过比较，本书发现布鲁克斯提出的复杂性与沃菲尔德（J. N. Warfield）提出的认知复杂性极为相似，而一致性、可变性和可视性类似于沃菲尔德提出的情景复杂性。因此，本书就将沃菲尔德复杂性理论应用于软件过程改进。

本书主要内容如下：

第 1 章是绪论。说明软件过程改进复杂性的根源在于软件的复杂性，并对研究内容、研究方法、创新之处和论文结构进行了说明。

第 2 章是软件过程改进的复杂性工作程序理论基础（合理性）。本书将软件过程按照价值链模型（M. Porter）分成软件开发过程和软件支持过程。并根据沃菲尔德复杂性理论的科学模型的论域和通用设计科学对软件工程七原理（B. W. Boehm）应用通用设计，得出了软件开发过程复杂性五命题。对软件过程改进六原理（Watts S. Humphrey）应用结构复杂性理论，得出了软件支持过程复杂性七命题。这 12 个复杂性命题是软件过程改进的理论基础。

第 3 章是软件过程改进的复杂性工作程序（结构化表示）。首先对 25 种认知障碍、复杂性工作程序和 16 条通用设计法则进行了述评。在此基础上设计了软件过程改进的复杂性工作程序，包括两个阶段：发现和解决。

第 4 章是软件生产的支持结构（应用）。首先讨论了软件过程中的知识管理，并参考传统的企业模型、微软企业模型和印度 Infosys 公司的知识管理，提出了软件企业模型，并为软件生产建立了一个支持结构来帮助软件企业实现其商业目标。

第 5 章是实证研究。包括问卷调查和案例研究。问卷调查包括软件过程改进要素和软件过程改进理论两部分。以美资旭电（深圳）科技有限公司资讯科技和系统服务部软件项目管理方法以及广州市灵狐系统工程公司软件项目管理为对象进行了案例研究。

总之，本书将沃菲尔德复杂性理论应用于软件过程改进，提出了软件开发过

程复杂性五命题和软件支持过程复杂性七命题，设计了软件过程改进的复杂性工作程序及软件生产支持结构，并对上述成果进行了实证研究，目的是降低软件过程改进的认知复杂性。

感谢国际系统研究联合会主席顾基发教授和沃菲尔德教授为本书写序。由于作者水平有限，不当之处，敬请指正。

作　者

2004 年 8 月 6 日

于华南理工大学系统与管理决策研究中心

目　录

第1章 绪 论

计算机软件是一种逻辑产品，其质量的改进既困难又复杂。为降低软件质量改进的困难程度及其复杂性，软件界对这个问题进行了大量的研究，解决的切入点正逐渐地集中到软件生产的软件过程中。本章对软件质量、软件过程改进（软件质量属性和软件过程分类）以及复杂性工作程序和知识技术（认知障碍分类，复杂性工作程序以及交互式管理支持结构、知识技术的概念、戴明（W. Edwards Deming）的知识原理、软件过程改进方法论、软件工程知识体系、项目管理知识体系和通过项目进行商业运作等）进行了述评，同时对论文研究内容、研究方法、创新之处及论文结构进行了说明。

1.1 问题的提出

1.1.1 软件质量的重要性

随着信息技术的发展和网络新经济时代的到来，计算机软件对经济发展的驱动作用显得越来越重要。目前计算机软件已经成为商业决策的引擎和现代科学研究及工程问题寻求解答的基础。计算机软件已经且正在被嵌入各种类型的系统中：交通、医疗、电信、军事、工业生产过程、娱乐、办公……不胜枚举。“计算机软件正在成为21世纪乃至下一个世纪国民经济发展的灵魂和核心”[1]。

软件质量，例如软件的可靠性、安全性和可维护性等，对社会经济甚至国家安全的影响日益明显。例如，阿里亚娜火箭发射失败就是软件中一个小小的Bug造成的，而由于软件错误造成核导弹误警报的恐慌也不乏其例[2]。又例如，在1991年海湾战争期间，一枚飞毛腿导弹刺入爱国者反导弹的外壳中，打中了位于沙特阿拉伯的一座军营（造成28名美国军人死亡，98人受伤），原因是爱国者反导弹的软件中包含有一个累计时故障[3]。这些例子显示了软件质量的重要性。那么，如何进行软件质量改进呢？本书作者认为，首先要分析软件质量改进的复杂性。

1.1.2 软件质量改进的复杂性

软件质量改进的复杂性最突出的实例是美国IBM公司在1963～1966年开发的IBM 360机操作系统。这一项目花了5 000人/年的工作量，最多时有1 000人投入开发工作，写出了近100万行源程序。尽管投入了这样多的人力和物力，

得到的结果却非常糟糕：据统计，这个操作系统每次发行的新版本都是从前一版本中找出 1 000 个程序错误而改正的结果。IBM OS/360 总设计师布鲁克斯（Brooks）博士事后总结了他在组织开发过程中的沉痛教训："……正像一只逃亡的野兽落到泥潭中做垂死挣扎，越是挣扎，陷得越深。最后无法逃脱灭顶的灾难，……一批批程序员被迫在泥潭中拼命挣扎，……谁也没有料到问题会陷入这样的困境……"

IBM OS/360 的案例说明软件质量改进的问题突出表现在组织软件开发过程中。由于软件开发过程中遇到的问题找不到解决的办法，致使问题积累起来，形成了日益尖锐的矛盾。这些问题包括[4]：

（1）软件开发无计划：由于缺乏软件开发的经验和有关软件开发数据的积累，使得开发工作的计划很难制定。主观盲目制定出的计划，执行起来和实际情况有很大差距，常常导致超过经费预算；另外，工作量的估计不准确，进度计划无法遵循，开发工作完成的期限经常一拖再拖。而对于已经拖延了的项目，为了加快进度，企业经常采用增加人力的办法，结果往往适得其反，不仅未能加快，反而更加延误了。在这种情况下，软件开发的投资商和软件的用户对软件开发工作既不满意，也不信任。

（2）软件需求描述不充分：作为软件设计依据的需求，如果在开发的初期阶段提得不够明确，或是未能得到确切的表达，开发工作开始后软件人员和用户又未能及时交换意见，一些问题将因不能及时解决而隐蔽下来，造成开发后期矛盾的集中暴露，而此时问题已经既难于分析也难于挽回了。

（3）软件开发过程无规范：开发过程没有统一的、公认的方法论和规范指导，参加人员往往各行其是，加之不重视文档工作，使设计和实现过程的资料很不完整；或是忽视了每个人工作与其他人的接口部分，发现了问题修修补补，这样的软件很难维护。

（4）软件产品无评测手段：未能在测试阶段充分做好检测工作，提交用户的软件质量比较差。在应用领域工作的不可靠软件，轻者影响系统的正常工作，重者发生事故，甚至造成生命财产的重大损失。

上面四种情况都说明软件开发过程的复杂性与开发人员的认知活动有关，人类认知活动的本质就是复杂的。本书认为软件质量改进复杂性的根源在于软件是一种智力密集型的逻辑产品。

1.1.3 软件产品的复杂性

布鲁克斯博士在其著名的文章"没有银弹"（*No Silver Bullet*）一文中指出[3]：软件的特殊性使寻找能解决所有软件产品的问题的方法将成为完全不可能的事。他将软件中的难题分为两类：本质问题（essence，即由软件本质决定的

固有的难题）和非本质问题（accidents，即目前遇到的，但并非软件产品固有的困难）。他指出了四个本质问题，即复杂性（complexity）、一致性（conformity）、可变性（changeability）和不可见性（invisibility）。他在文章中使用的“复杂性”的含义是“令人费解的、复杂难懂的”（complicated or intricate）。实际上，所有这些名称都是使用它们的非技术含义。

1．复杂性

复杂性是软件固有的属性。一个软件无论如何设计，其各部分一定会有交互。例如，模块的状态取决于参数的状态，全局变量的状态（能被几个模块访问的变量）将对整个产品产生影响。当然，复杂性可以通过例如面向对象技术这类的方式降低，但永远不可能完全消除。

他认为如果将一套软件简化，则整个过程将是毫无用处的。数学和物理的简化技术之所以有用，只是因为那些系统的复杂性是非本质问题，而不是像在软件产品中那样，复杂性是本质问题。

软件的这种本质的复杂性使软件产品很难理解。实际上，通常没有人能真正从整体上理解一个大型软件产品。若产品作为一个整体不能被理解，则只有较小的、独立的组成部分能被完全理解。这会导致开发组成员间信息交流不通畅，进而引起时间和费用上的超支，这是大规模软件产品开发中的通病。此外，对产品的所有方面缺乏理解会使规格说明出现错误。

这种本质上的复杂性不仅影响软件产品本身，而且也会影响软件过程的管理。除非管理人员能得到他所管理的工程的精确信息，否则他将很难确定该软件工程各个阶段的人员需求，也很难准确地做出预算。于是，向管理者提交的关于进度和最后期限的报告就不可能准确。如果管理者和向管理者提交报告的人均不知道哪个松散环节需要抓紧，则拟订测试计划也将是很困难的。另外，如果一位在编人员离开了，则试图训练一位替补人员就像一场恶梦。

软件复杂性的另一个结果是使维护过程十分复杂。众所周知，软件生命周期2/3的时间是用于维护。除非维护人员真正理解产品，否则改正性维护或增强性维护将会给产品带来破坏，使其后的维护必须修正由原来维护所带来的破坏。由粗心而带来的破坏总是可能存在的，即使由原作者做修改也是如此，而如果维护人员在不了解产品的情况下做修改，破坏无疑会进一步加剧。贫乏的文档，或者没有文档，更糟糕的是不正确的文档，经常是不能进行正确维护的主要原因。但无论文档多么好，软件固有的复杂性也使维护人员难以消除它给维护工作带来的不利影响。

2．一致性

布鲁克斯博士提出第一类一致性是软件必须和现有系统接口，从而使软件的复杂性达到了不必要的程度；第二类一致性是人们错误地认为软件是最容易调整

的部分，使得软件的复杂性达到了不必要的程度。本书认为这增加了软件过程改进的复杂性，换言之，使复杂性升级了。

3．可变性

系统的功能体现在它的软件中，通过修改软件，就可以收到改变系统功能的效果。布鲁克斯博士认为可变性是软件的本质属性，是一个不可克服的固有的复杂性问题。原因有下面四点：

（1）软件是现实世界的模型，现实世界改变了，软件也必须适应，反之只有死亡；

（2）如果一套软件有用，则感到满意的用户也会进一步要求扩展一些超过产品原来设计中的功能；

（3）软件的最大优势就是它比改变硬件要容易得多；

（4）成功软件的生存周期比它所服务的硬件更长。

一般来说，要使软件能在这些新硬件上运行，必须对软件做一定的修改。因此，软件的这些属性决定了软件修改的压力将一直存在，而且经常是大幅度的修改。本书认为这也是软件过程改进复杂性的困难所在。

4．不可见性

软件无法形象化表示的结果不仅使软件难以理解，而且严重妨碍了软件专业人员之间的联系。例如，除了将 150 页的程序清单给同事，再给一张需要改动的说明表之外，似乎再没有其他办法了。

使用各种可视化图表（例如数据流程图等）是使产品的某些方面可视化的极有用、极其重要的方法。问题是这些图表不能体现产品的各个方面，也无法判断某一种形象化表示法遗漏了哪些信息。本书认为不可见性也增加了软件过程改进的复杂性。

布鲁克斯博士认为软件技术中已经有了三个主要的突破：高级语言、分时共享和软件开发环境。他建议要改变软件生产方式，并指出在软件生产方面取得重大突破的最大希望在于培训和鼓励伟大的设计师，本书用图 1.1 来表示。换言之，本书认为布鲁克斯博士提出的软件复杂性（认知复杂性，详见 1.2.2）问题根源至少包括一致性、可变性及不可见性这三个软件本质问题（情景复杂性，详见 1.2.2）。可以通过软件过程设计（软件过程改进，software process improvement，SPI，详见 1.2.1）的途径来解决软件的复杂性。那么，软件过程改进的复杂性又是什么呢?

1.1.4 软件过程改进的复杂性

著名的软件工程专家 Tom DeMacro 集 30 年软件项目管理的经验[5]，认为软件项目中对人员的管理问题不能像其他事物那样简单地划分，机械地对待。他特