

能力 成熟度

模型集成及其应用

Application of Capability Maturity
Model Integration (CMMI™)

龚波等编著



中国水利水电出版社
www.waterpub.com.cn

能力成熟度模型集成及其应用

龚波 等编著

中国水利水电出版社

内 容 提 要

CMMI (Capability Maturity Model Integration) 是卡内基·梅隆大学 2001 年 9 月推出的系统工程和软件工程的集成成熟度模型, 也是业界流行的工程实施和管理方法。2001 年 12 月, SEI (美国卡耐基·梅隆的软件工程研究所) 正式发布 CMMI 1.1 版本。与原有的能力成熟度模型相比, CMMI 涉及面更广, 覆盖软件工程、系统工程、集成产品开发和系统采购。该模型提供连续模型和阶段模型这两种表示方法, 组织根据自己的过程改进需求可以自由选择合适的表示方法。

全书共分为 7 章, 首先讨论众多 CMM 模型给组织过程改进和交流改进思想所带来的困惑和困难, 然后按照从浅到深的顺序逐步介绍了 CMMI 模型引入、模型基础知识、CMMI 过程域、阶段式表示概要、连续式表示概要、CMMI 评估以及 CMMI 迁移等待, 基本覆盖了 CMMI 产品集的所有产品, 并提供实施 CMMI 过程改进的建议和实践。

本书可以供企业主管、项目经理、系统和软件工程师以及过程改进实践人员参考, 也可供高等院校软件工程专业的高年级学生、研究生和老师使用。

图书在版编目 (CIP) 数据

能力成熟度模型集成及其应用 / 龚波等编著. —北京: 中国水利水电出版社, 2003

ISBN 7-5084-1477-2

I. 能… II. 龚… III. 软件工程 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2003) 第 026058 号

书 名	能力成熟度模型集成及其应用
作 者	龚波 等编著
出版、发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@public3.bta.net.cn (万水) sale@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (发行部)、82562819 (万水)
经 售	全国各地新华书店
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	787×1092 毫米 16 开本 16.75 印张 378 千字
版 次	2003 年 4 月第一版 2003 年 4 月北京第一次印刷
印 数	0001—5000 册
定 价	26.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社发行部负责调换

版权所有·侵权必究

前 言

过程改进的概念和理论是从 Deming、Crosby 和 Juran 等人的质量管理工作中派生而来的，其目的是增强组织工作过程的能力。从本质上讲，过程能力是指一个过程产生预计结果的固有能力和能力。通过加强过程能力，组织就可以预测和度量过程，并且控制和消除引发质量低下和生产率不高的因素。

30 多年来，Deming、Crosby 和 Juran 等人提出的过程管理理论已经被用于解决许多组织共同面临的问题。但是这些理论和解决方案虽然被提出了，但是实践和理论艺术之间仍旧存在很大的差距。现在借助于模型的概念，可以把他们提出的理论比较好地与实践联系起来，真正指导过程管理概念和理论的实施。

现在业界最成熟的过程化评估和改进方法是美国卡耐基·梅隆大学的软件工程研究所 (SEI) 提出的过程能力成熟度模型集。这些模型描述了有效的过程单元的框架，为组织机构描述了从混乱的、不成熟的过程向成熟的、有秩序的过程改进的一条途径。过程成熟度指针对具体的过程明确定义、管理、度量、控制的程度。成熟度表明工程能力增长的潜力，表明组织过程的丰富性及其应用于项目的一致性。改进组织过程要求组织机构加强管理机制，并能长期、始终地关注过程改进。

自从 1991 年 SEI 发布 SW-CMM (V1.0) 以来，人们逐渐开发了多种单一学科的 CMM 模型，其中最具有影响的包括：系统工程 (SE-CMM)、软件工程 (SW-CMM)、软件采办 (SA-CMM)、人力资源管理 (P-CMM)，以及集成化产品和过程开发 (IPPD-CMM) 等。这些模型的实施有助于改善组织过程，以构造更好的产品、提高质量、降低成本，同时模型的实施对于推广过程改进理论和实践起到很大的促进作用。现在许多大公司和意图打入国际市场的公司都积极谋求 CMM 认证，并把 CMM 认证看作国际市场的“准入证”。

但是随着系统环境和开发环境的不断复杂化，比如分布式和并行式，单一学科的过程改进模型已经不适合于现代系统开发需求，不能支持并行工程环境。许多组织希望多个单一 CMM 模型能够提供共同的组织框架和共同的工作接口，以消除不同功能部门和过程组之间的屏障。同时，许多组织希望能够在组织范围内总体改进组织的各个学科和过程。然而，原来特定于学科模型由于存在结构、内容和方法等方面的差异，严重限制了这些组织的改进组织过程能力。进一步而言，使用多个独立模型会导致培训、评估和改进活动的成本增加。部分实践人员在应用各种不同的模型来实现特定需要时容易产生混淆。

实现模型集成的目的是：消除不一致性；减少重复；增加透明度和可理解性；提供公共术语；提供一致的风格；建立统一的构造规则；维护公共构件；确保与 ISO 15504 保存一致；实现尽量好的可继承性。

能力成熟度模型集成 (CMMISM) 改进了成熟度模型 (CMM[®]) 的概念。把 CMM 概念扩展到多个学科。像 SW-CMM、EIA/IS731、IPD-CMM、SA-CMM 以及其他过程改进

模型一样，CMMI 模型是有助于组织改进其过程的工具。CMMI 模型有助于组织提高其产品和服务开发采办，以及维护过程。本模型所覆盖的概念包括系统工程、软件工程、集成化产品和过程开发以及供应商源，并且包括传统的 CMM 概念，比如过程管理和项目管理。每个 CMMI 模型与其他模型一起使用，使得组织可以更加容易地实现组织范围的过程改进。

CMMI 模型有两种表示法：阶段式表示和连续式表示。阶段式表示集中于使用成熟度等级来度量组织过程改进，这一点与 SW-CMM 完全相同；连续式表示集中于使用能力等级来度量单个过程的改进程度，这与 EIA/IS 731 相同。

本书从 CMM 模型入手，首先讨论众多 CMM 模型给组织过程改进和交流改进思想所带来的困惑和困难。然后详细介绍 CMMI 思想的提出、发展和现状，并分章节介绍了 CMMI 框架、模型部件、两种表示方法、评估、CMMI 迁移以及不同表示方法的模型规格说明等。可以说本书比较完善和详细地介绍了 CMMI 模型的衍生之路、现状和理论框架。本书共分为 7 章，按照由浅入深的顺序逐步介绍了 CMMI 模型引入、模型基础知识、CMMI 过程域、阶段式表示概要、连续式表示概要、CMMI 评估、CMMI 迁移，基本上覆盖了 CMMI 产品集的所有产品。本书是按照最新的 CMMI-SE/SW/IPPD/SS v1.1 介绍的，保证了内容和资料的实时性。

本书的最终成稿和出版离不开家人对我的支持和鼓励，你们的关心是我继续前进的动力。在此期间，我的儿子龚克也出世了，现在小家伙是我们家的中心，他非常可爱，我衷心希望我的宝贝永远健康快乐。感谢我的爱人为我做出的牺牲。感谢父母多年的养育之恩。

感谢中国水利水电出版社的孙春亮先生和石永锋先生提出的意见和建议，你们对图书出版的深刻洞察力和严谨工作作风，是本书质量的强有力保证。

感谢我的导师何新贵院士对我的指导和教诲，您的严格治学作风永远值得我学习。感谢我的朋友李志给予我的帮助，虽然我们居住在不同的城市，但是我们的心是相通的，真心祝福你万事如意。感谢我的朋友赵军锁提供的书房，非常安静和舒适。

本书是集体合作的结果，龚波负责全书的策划和大部分内容的编写，田丽韞同志负责本书的资料整理和收集。其他对本书做出贡献的人员还有：刘卫宏、张文敏、刘向军、田军、赵军锁、姜南、李志、张巧莉、田蕴哲、牛志奇、丁天、龚志翔、李红玲、白红利、金荣学、薛彪、叶哲、于自跃等。

龚波

2003 年 1 月于北京

目 录

前言

第 1 章 CMMI 模型引入	1
1.1 软件过程改进	1
1.1.1 规范的软件过程是软件活动成功的关键	1
1.1.2 软件工业浪潮	2
1.1.3 软件过程运动	3
1.1.4 软件过程改进的效果	5
1.2 CMM 引发的问题	8
1.3 CMMI 的出现	11
1.3.1 CMMI 组织结构	13
1.3.2 CMMI 开发的指导原则	14
1.3.3 CMMI 源模型	16
1.3.4 CMMI 与相关模型的兼容性	20
1.3.5 CMMI 范围和概念	21
1.3.6 CMMI 产品集	21
1.4 总结	22
第 2 章 CMMI 模型基础	23
本章目标	23
2.1 CMMI 框架	23
2.2 CMMI 模型部件	24
2.3 制度化	25
2.4 CMMI 表示方法	27
2.4.1 CMMI 阶段式表示	29
2.4.2 CMMI 连续式表示	37
2.4.3 两种表示法的区别和阶段等价关系	45
2.5 使用 CMMI 模型	49
2.5.1 认识 CMMI 模型	49
2.5.2 过程评估	50
2.5.3 选择模型	52
2.5.4 选择表示方法	52
2.5.5 模型剪裁	53
2.5.6 迁移到 CMMI	55

2.6 总结	57
第3章 CMMI 过程域	58
3.1 CMMI 过程域	58
3.2 过程管理类.....	61
3.2.1 基本过程管理类过程域.....	61
3.2.2 高级过程管理过程域.....	65
3.3 项目管理类.....	68
3.3.1 基本项目管理类过程域.....	69
3.3.2 高级项目管理类过程域.....	73
3.4 工程类.....	80
3.5 支持类.....	92
3.5.1 基本的支持类过程域.....	93
3.5.2 高级的支持类过程域.....	98
3.6 总结	102
第4章 CMMI 阶段式表示	103
4.1 结构综述.....	103
4.2 成熟度等级概述.....	104
4.3 模型概述.....	105
4.3.1 成熟度等级 1: 初始级.....	105
4.3.2 成熟度等级 2: 已管理级.....	106
4.3.3 成熟度等级 3: 已定义级.....	128
4.3.4 成熟度等级 4: 定量管理级.....	173
4.3.5 成熟度等级 5: 优化级.....	179
4.4 总结	185
第5章 CMMI 连续式表示	186
5.1 结构综述.....	186
5.2 能力等级概述.....	186
5.2.1 能力等级 0: 未执行级.....	186
5.2.2 能力等级 1: 已执行级.....	187
5.2.3 能力等级 2: 已管理级.....	187
5.2.4 能力等级 3: 已定义级.....	188
5.2.5 能力等级 4: 定量管理级.....	189
5.2.6 能力等级 5: 优化级.....	190
5.3 模型概述.....	190
5.3.1 过程管理.....	190
5.3.2 项目管理.....	198
5.3.3 工程.....	211

5.3.4 支持	221
5.4 总结	230
第 6 章 CMMI 评估	231
6.1 评估基础	231
6.2 SCAMPI 概述	232
6.2.1 源方法	233
6.2.2 方法目标和特征	233
6.2.3 目标证据源	233
6.2.4 SCAMPI 设计原则和假设	234
6.3 SCAMPI 模型使用	235
6.3.1 SCAMPI 剪裁	236
6.3.2 时间线和角色	236
6.4 方法描述	237
6.4.1 阶段 1: 计划和准备评估	238
6.4.2 阶段 2: 执行评估	240
6.4.3 阶段 3: 报告结果	241
6.5 CMMI 评估需求	242
6.5.1 介绍	243
6.5.2 CMMI 评估方法的好处和特点	243
6.5.3 CMM 评估方法类结构	243
6.6 IDEAL 模型	244
6.7 总结	246
第 7 章 CMMI 迁移	247
7.1 技术迁移	247
7.2 过程或技术接纳曲线	248
7.2.1 Moore 技术生命周期曲线	248
7.2.2 Patterson-Connor 接纳曲线	250
7.3 集成化过程改进的原则	251
7.4 实施集成化过程改进的范例	252
7.4.1 从 SW-CMM 迁移到 CMMI	252
7.4.2 从 SE-CMM/SECM 迁移到 CMMI-SE/SW	254
7.5 总结	256
附录 A 向 CMMI 模型添加新学科	257
参考文献	258

第 1 章 CMMI 模型引入

“产品的质量在很大程度上依赖于开发和维护该产品的过程的质量”

——Shewhart, Juran, Deming 和 Humphrey

本章目标

在本章中，我们将首先引入基于模型的过程改进的概念，然后介绍 SEI 推出的 CMM 模型概念，简单浏览 CMM 产品集为组织带来的过程改进收益。接下来讨论由于现代工程环境和并行工程等新技术的引入，众多单一学科 CMM 模型所引发的问题，使得集成众多单一学科模型势在必行。本章从 CMMI 组织、开发原则、源模型、范围和概念等方面入手，对 CMMI 进行初步介绍，为进一步的学习做铺垫。

1.1 软件过程改进

1.1.1 规范的软件过程是软件活动成功的关键

实践表明，能够在规定时间和预算内开发可靠的系统对于很多组织来说是非常困难的。在日常的商业活动和生活中，软件扮演着越来越重要的角色，因此因软件延期、质量低下等引发的问题对这个世界的影响越来越大，也就出现了所谓的“软件危机”问题。类似的故事和故障我们已经见得很多了，最典型的是欧洲空间中心卫星发射失败的例子。

之所以出现“软件危机”，并出现对过程改进的强烈需求，主要原因是工程环境日趋复杂。我们正处于系统复杂性增长曲线的上升阶段，这种复杂性具体指：软件复杂性、系统管理复杂性、开发过程的复杂性等，而与此对应的是管理手段和管理理论的相对简单性。正如 Robert Britcher 描述的那样“实践的复杂性总是令理论的简单性相形见绌”。

多年以来，工程开发和研究机构一直在寻找这个问题的答案。20 世纪 80 年代出现的 CASE 工具曾许诺可以依靠技术和工具来很好地解决这个问题，事实证明这是完全错误的。现在软件业认识到“有工具的笨蛋依然是笨蛋”。通常，软件解决方案要涉及到三方面要素：人、过程和技术。首先，经验表明当实施解决方案时，通常最不可能出现问题的是技术要素，而过程和人员是导致项目成功或者失败的关键。同样，在许多机构中，软件项目常常严重超期或超出预算，项目处于混乱状态，根本不能体现新方法、新技术和新工具所带来的优势。

其次，软件项目是集体活动，需要组织、团队的共同努力才能完成。Humphrey 在 1995

年曾经说过“在一个球队中，一些人踢足球，一些人打棒球，而其他有些人打橄榄球。在这种环境下，即使拥有最好的球员，也难以组织一支优秀的球队”。可以看出，怎样保证团队人员按照固定的成功过程来使用技术和工具才是最重要的，也就是怎样保证过程规范化才是项目成功的关键。

我们可以把软件活动与团体舞蹈进行类比。在软件企业里面有许多天才，他们很擅长单兵作战。尽管他们很聪明，但是如果他们只专注于个人的职责，而忽视了共同的职责，所开发的软件肯定是不符合要求的；就像在团体舞蹈中，虽然每个舞蹈演员都很优秀，可以开独舞晚会，但是当他们参加团体舞蹈时，必须有所抑制，否则会踩到别人的脚。他们必须学会一起跳舞，他们知道自己的那部分很重要，但是当实现自己那部分时，他们需要意识到其他人处于哪个位子，他们下一步将进行到哪一步，这样所有人就能步调一致了。

在团体舞蹈中，就需要舞蹈编排来安排每个人的动作，以及怎样与别人配合，怎样根据音乐和其他人的动作来做出自己的动作。在软件活动中，同样需要相应的软件过程设计，保证每个人的交互接口，每个人的职责。如果每个人都能够很好地完成自己的动作，那么观众才能欣赏到赏心悦目的舞蹈，客户才能得到想要的软件。

1.1.2 软件工业浪潮

软件工业发展呈现三次浪潮（Robert Lai, 1993）。第一次浪潮以 20 世纪 70 年代逐渐发展起来的瀑布式生命周期和结构化方法为特征；第二次浪潮是追溯于 20 世纪质量控制运动的过程成熟运动；第三次浪潮是预期的软件工业化。每次浪潮都是对上次发展浪潮的改进。

第一次浪潮的重要特征是用图表来表示软件活动的能力（Royce, 1970），从而使软件开发向工程规范化靠近。与“go-to”语句和“意大利面条式编码”相比，结构化方法向前迈进一大步。结构化方法是为了满足那些利用共享系统来建立复杂的交互式商业应用和进行系统维护而发展起来的。结构化方法关注方法本身，即需求的正规化和设计，以及创建最终需求遵循情况的记录方法。尽管这是将软件开发从“手工作坊”向大规模生产转换的开端，但还是远远不够。那些真正导致项目成功或者失败的真实原因（比如，项目管理或者需求管理）还未成为关注焦点。

过程成熟度是结构化方法的进一步发展。软件过程管理的范围比分析、设计和编码更广泛，它覆盖软件开发过程及其支持活动的各个方面。目前，软件工业处于软件成熟度浪潮的中心。通过软件过程改进网络（SPIN, Software Process Improvement Networks）、软件工程研究所（SEI, Software Engineering Institute）和国际化标准组织（ISO, International Standards Organization）的努力，软件过程运动在全世界范围内得以传播，并且获得越来越多的人支持。第二次浪潮有可能进一步推进软件工程化，使软件工业成为以过程为中心、注重过程的工业领域，就像制造业一样。

Watts Humphrey 在软件开发中引入了质量统计概念。过程成熟度运动为软件行业的第三次浪潮——“软件工业化”铺平道路。在第三次浪潮中，软件开发将变得如同装配和制造过程一样。引发第三次浪潮的技术因素包括面向对象技术和复用组件库，使得利用标准

复用组件来装配软件变成可能。第三次浪潮启动的关键因素是软件工程过程，实现质量、进度和功能的可视性和可度量性。

1.1.3 软件过程运动

软件成熟度运动的先驱是 Watts Humphrey 和卡耐基·梅隆大学的软件工程研究所（SEI）。Watts Humphrey 奠定了软件过程运动的基础性原则，他把软件任务看作可控的、可测试的和可提高的过程（Humphrey, 1989）。Humphrey 和同事从 20 世纪 80 年代就开始了有关软件过程的研究性工作。

软件过程运动可以追溯到已经存在近 70 年的产品质量管理原则。质量运动的权威是 Shewhart、Deming、Juran 和 Crosby。Walter Shewhart 在 20 世纪 30 年代发表了质量的统计控制原则。Edwards Deming、Joseph Juran 和 Phill Crosby 进一步发展和验证了 Shewhart 的思想。其中 Phil Crosby 发展了质量成熟度的量化。统计控制的原则是质量运动和软件过程运动的主题。Watts Humphrey 说到：“如果整个开发过程处于统计控制之下，那么仅仅通过改进过程就可以持续获得好结果；如果离开对过程的统计控制，就无法实现持续改进。”

二战之后，Edwards Deming 把过程统计控制的概念运用到日本工业界。Watts Humphrey 认为，尽管两者存在差异，但是适用于汽车、相机、手表和钢铁等制造业的生产原则也同样适合于软件行业。Humphrey 将 Deming 的原则、Juran 改进的内容，以及 Crosby 的成熟度量结合到一起，并运用到软件开发过程中。Watts Humphrey 的早期工作对软件过程产生很大影响，引起全世界的关注。后来出版的《管理软件过程》更是推进了软件过程化运动。

美国国防部（DoD）赞助 SEI 的研究工作，Watts Humphrey 被任命为软件过程计划的主任，这个计划后来产生了 CMM。除了 CMM 之外，软件过程运动在全世界产生了其他有影响的模型和标准，比如：

- 英国国防部（MoD）资助的软件过程标准和软件过程改进项目发展成为 SPICE，强调采用国际标准对软件供应商的软件过程能力进行评估的必要性，最终导致 ISO/IEC 15504 过程评估标准的开发。
- 原联邦德国军方资助的 V 模型，它是软件开发标准，包括软件工程生命周期和支持活动。
- 以加拿大贝尔公司为主开发的 Trillium，是个主动的、持续改进的软件过程改进标准。
- 国际标准化组织（ISO）开发的 ISO9000 系列标准。其中部分文档适用于软件开发，包括 ISO 9000-3、ISO/IEC 12207 和 ISO/IEC 15504 等。

现在，与过程管理相关的研究项目在逐年增加，人们希望能够有充足的手段以改进开发过程，实现开发过程的可控性，将失效限制在萌芽中。软件过程是指软件开发人员开发和维护软件及相关产品（如项目计划、设计文档、代码、测试用例和用户手册）的一套行为、方法、实践及变换过程。

概念 过程定义：过程是要达到特定目标而必须实施的一组实践；过程定义也许包括工具、方法、资料和/或人员。过程可以提供富有建设性的、具有高度次序化的焦点行为。

对于焦点行为是人员的行为而言,员工工作再努力,如果不通过过程化来管理,也是枉然;对于焦点行为是技术的行为而言,如果没有正确的“路标”,技术再好效果也不佳;在具有合适的过程“路标”环境下,新技术才能发挥其最大效能。

过程模型定义:过程模型是描述有效过程特征的结构化元素集合。这些过程经过经验证明是有效的。模型有助于设定过程改进目标和优先级、增强过程,并且为确保稳定的、有能力的和成熟的过程提供指南,还可以为组织过程改进提供指南。

为什么模型是重要的?模型提供了解决问题的最初出发点、公共语言和共享视图,以及区分行动优先级和综合团队经验的框架。^①

模型对过程需求提供了一个公共集合,综合了最佳实践和实际经验知识,并以一种可以用来指导过程改进优先次序的格式来表达。借助于模型,组织机构可以通过已经证明能提高过程能力的实践来修改或创建过程。组织还可以利用模型实现两个目的:建立改进基线;使用度量技术来度量和分析过程改进活动和进展。^②

一般来说,基于模型的过程改进是从管理的承诺和评估开始的。评估结果又可以作为制定下一步行动计划的基础。当完成这些计划后,再进一步进行评估。如此反复迭代,循环改进不止。其目标是使组织成熟,使其持续地监督和改善过程,始终一致地生产高质量产品,在市场竞争中灵活自如,并随时进行调整以满足客户的需求。

现在业界最成熟的过程化评估和改进方法是美国卡耐基·梅隆大学的软件工程研究所(SEI)提出的过程能力成熟度模型集。这些模型描述了有效的过程单元框架,为组织描述了从混乱的、不成熟的过程向成熟的、有纪律的过程进行改进的一条途径。过程成熟度针对具体的过程明确定义、管理、度量、控制的程度。成熟度表明工程能力增长的潜力,表明组织过程的丰富性及其项目应用的一致性。改进组织过程要求组织加强管理机制,并能长期、始终地关注过程改进。

概念 什么是 CMM (能力成熟度模型):能力成熟度模型是特定学科中成熟实践的参考模型,被用于评估组织实施该学科的能力。CMM 模型之间区别主要依赖于:学科(软件、系统、获取等)、结构(阶段式和连续式)、成熟的定义方式(过程改进路径),以及能力的定义方式(制度化)。

中国工程院院士何新贵教授领导的一个研究小组根据中国的国情,提出了 CSCMM(中国 CMM),并在某大型工程中付诸实施。东大阿尔派、联想、鼎新、亚信等国内企业的有关部门也通过了 CMM 一定级别的认证。2000 年 6 月国务院颁发了《鼓励软件产业和集成电路产业发展的若干政策》,该文件的第 5 章第 17 条明确提出“鼓励软件出口型企业通过 GB/T19000-ISO9000 系列质量保证体系认证和 CMM(能力成熟度模型)认证。其认证费用通过中央外贸发展基金适当予以支持。”由此可见,我国科研机构、企业和政府对

^①选自于 Mike Phillips 在 April 2002 于 European Software Process Improvement Group Conference 会议上的发言。Mike Phillips 是 CMMI Program Manager。

^②选自于 CMMI DISTILLED—A Practical Introduction to Integrated Process Improvement, 作者是 Dennis M. Ahern, Aaron Clouse, Richard Turner。

软件过程化改进的态度和认识。

1.1.4 软件过程改进的效果

每种投资都希望获得相应回报。现实生活中实施软件过程改进带来商业收益的例子越来越多。过程改进的投资回报率一般在 5:1~13:1 之间。虽然我们都在尝试确定软件过程改进的投资回报率的定量证据,因为这样做可以使过程改进更易获得管理部门的资助和软件过程参与人员的配合。但是,需要把管理部门的注意力集中于软件过程改进的定性收益上,诸如提高员工士气、保持软件工程的性能、提高软件项目进度的可视性、减少缺陷,以及更加关注项目或者商业目标等。

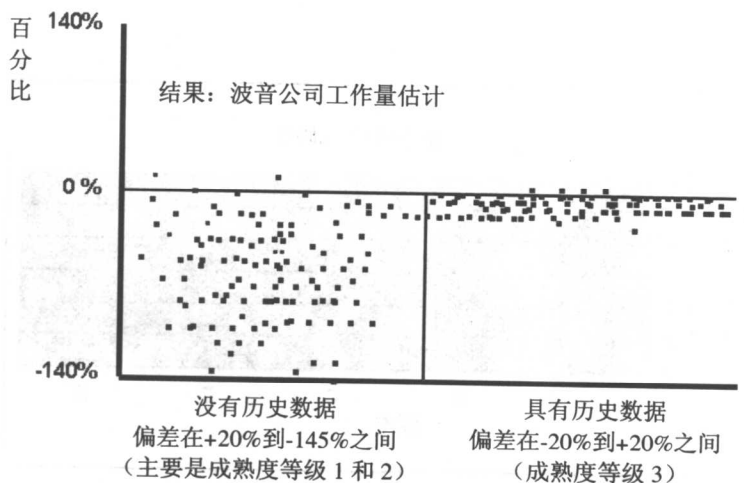
在 1996 年的 SEPG 会议上, Motorola 公司的副总裁 John Major 在演讲报告中总结了从 1992 年进行软件过程改进以来获得的商业收益:

- 超过 75% 的产品项目的成熟度等级确定为第 3 级或者更高。
- 在过去两年中,获得接近 10 倍的软件质量改进率。
- 生产效率提高两倍多。
- 软件周期缩短两倍多。

概括地讲,实施软件过程改进可以带来如下好处:

- 提高项目进度和财政情况的可预测性。软件过程改进的一个方面就是提高过程的可视性,增强过程能力的可度量性,能够使类似项目获得成功。

从图 1-1 中可以了解到,当具备成熟度等级 3 的过程能力时,波音公司对工作量的估计偏差只有 -20% 到 +20%。预测精度的改善,有助于管理部门的决策和公司商业长期目标的制定。



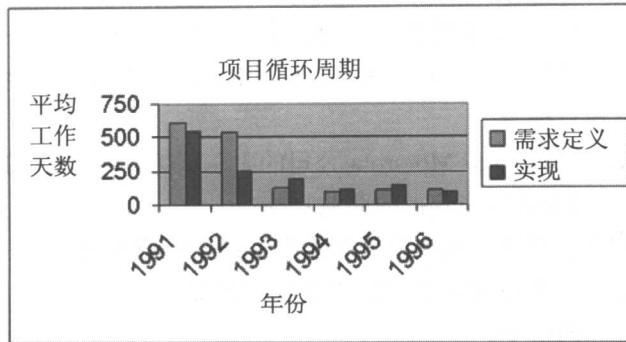
(基于对波音公司信息系统的 120 项目的预测估计结果)

源自: John D. Vu "Software Process Improvement Journey: From Level 1 to Level 5" 7th SEPG 会议, San Jose March 1997.

图 1-1 波音公司实施过程改进增强了可预测性

- 缩减循环周期。随着软件过程化的思想日渐深入人心，并成为每个人的自发行为，项目在培训、沟通方面的花费时间就会缩短，这也意味着更少的错误、更少的返工，所有这些都能提高生产率，从而最终缩短循环周期。

图 1-2 演示了 AFB 公司在 1991 年到 1996 年间，在逐步实现以过程为中心的管理思路 and 改革以来，项目循环周期逐渐减少的情况。

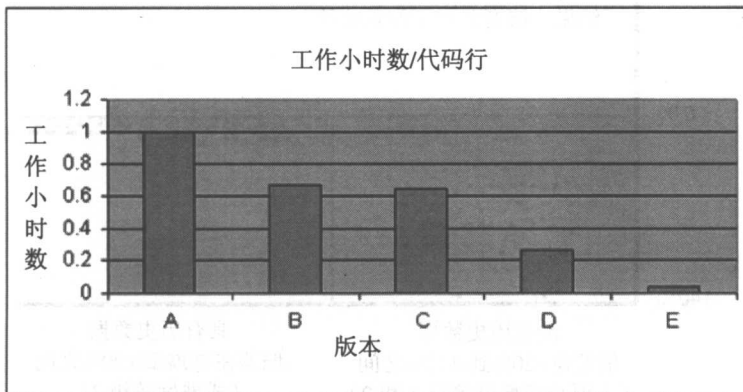


源自：Software Engineering Div., Hill AFB, Published in Crosstalk May 1999

图 1-2 AFB 公司实施过程改进缩减了项目开发循环周期

- 增强的生产率。软件过程改进在提高产品质量的同时可以减少发生的错误和返工。即使出现错误，也会在项目早期的各种审查活动和风险管理活动中得到妥善处理，同时软件过程改进可以提高项目人员的大局观，减少不同部门因为“老死不相往来”的陈旧观念所带来的沟通障碍。

图 1-3 演示了 AFB 公司某项目的不同版本在实现以过程为中心的改进之后对项目工作效率的影响。



源自：Software Engineering Div., Hill AFB, Published in Crosstalk May 1999

图 1-3 AFB 公司实施过程改进提高了生产率

- 提高产品质量（根据缺陷进行度量）。从改进后的过程对软件质量的影响可以看到质量的提高，从而提高使用该软件的产品或者服务的质量。产品质量提高可以提高客户的满意度、减少产品的缺陷、减少退货，以及避免媒体压力等，这些都是商业成功的显著标志。
- 增强客户满意度。实施面向过程的方法能够使类似项目获得成功，改进任务周期，更加保证产品质量，同时更加注重来自于各方面的意见，因此使得客户的意见和改进能够尽快地实现和解决。
- 增强雇员士气。实施过程改进带来质量和效益的提升，加强产品品牌的知名度，也进一步增强公司职员的自豪感和责任感，增强职员对公司前景的信任感。同时，软件过程改进的重要一点就是持续改进，这也需要调动职员的创造力和创新精神，为过程进一步改进出谋划策。
- 增强投资回报率。产品质量提高和生产率提高自然会使得投资回报率提高。同时，减少项目中无谓失误的环节也会节省项目时间和成本。
- 降低质量成本。规范的软件过程实施和制度化之后，软件过程和制度深入人心，每个人都按照规范化过程进行操作。因为规范化过程是成功实践的总结，实施出现的错误机率比较小。即使出现错误和故障，实施各级审查制度可以保证在项目早期发现并解决这些问题，而不是像非规范化项目那样在项目后期发现问题而进行大量返工。

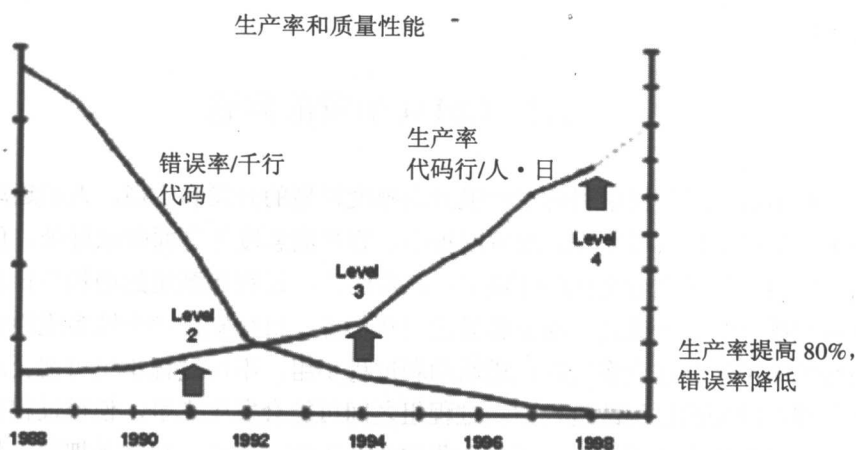


图 1-4 AFB 公司实施过程改进在生产率和产品质量方面的收益

下面给出来自于 DACS 的统计数据，以便于读者能够定量地把握过程改进带来的好处：^①

^①摘录自 A Business Case for SPI Revised – Measuring ROI from Software Engineering and Management, DACS, September 1999
<http://www.dacs.dtic.mil/techs/roispi2/>.

- | | |
|------------|--------|
| ● 开发成本 | 减少 73% |
| ● 返工量 | 减少 96% |
| ● 平均开发时间 | 减少 37% |
| ● 发布后错误 | 减少 80% |
| ● 发生风险的可能性 | 减少 92% |
| ● 投资回报率 | 21:1 |

有关成熟度等级（能力等级）和过程规范的几个问题

1. **成熟度等级（能力等级）是不是目的？**成熟度等级（能力等级）有助于我们将注意力集中于具有最高优先级的问題，但是人们容易忽略过程改进的真正目的，却把获取更高层次的成熟度等级（能力等级）当作自己的目标。当注意力集中于成熟度等级（能力等级）时，他们会认为评估是度量等级的一种方式。由于把成熟度等级（能力等级）看作目标，就非常关注于其准确性。但是，软件本身太过于复杂，以至于无法通过几个数字就可以完全度量清楚。我们真正需要的是一个框架，通过这个框架，可以使工程师和经理检查和讨论其过程，最后根据对组织情况的详细了解，确定关键问题，从而决定对什么进行改进，以及如何进行改进。

2. **成熟度等级的模型过于简单化？**成熟度模型的关键过程域最重要一点是目标。我们应该关注目标本身，并且使用各种可能的方法来实现这个目标，成熟度等级中提供的实践只是经过实践检验的成功实践，建议你剪裁和借鉴这些实践。但是成熟度模型从来都不拒绝新思想，SEPG 也会发现效率更高的实现目的的办法。软件工程是个新兴领域，充满新的思想，成熟度模型不会限制新思想的出现；但是过分详细的模型很有可能妨碍新思想的出现和实施。

1.2 CMM 引发的问题

随着众多适合于不同学科的单能力成熟度模型的开发和实现，人们逐渐认识到过程改进在生产效率、产品可靠性、投资回报率、客户满意度等方面带来好处。但是在跨越不同组织、学科、公司或者文化的环境下，随着组织、过程和系统的增长与日益复杂，又滋生了新的问题——过程改进活动很容易迷失在任务、日程安排和个性多重性的环境中。因为大型组织中可能存在众多过程改进活动和过程小组，不同过程小组可能会选用不同的、有时甚至相互冲突的过程改进模型。过程组之间可能会发生竞争，使得过程改进成为跨越不同组织、相互没有联系，并且无需合作的修补工作。最后，通常是把更多精力用于过程改进的边缘活动，而不是放在过程改进活动本身。

对于小型组织也存在类似问题，在小型公司中通常一个人要同时执行多项工程功能，如果每个人对于不同过程都采用不同模型来处理，对于个人会带来多大的工作负担，这样的后果是与其改善，到不如顺其发展。试想一下，如果能够把这些过程组织在一个整体框架下，使用公共的接口来操作会节省多少无谓的工作量！

还有一个不容忽视的现实情况是近年来工程和产品开发的组织发生了巨大变革，变革目的是消除与分段开发有关的低效率。在分段开发中，前一阶段产生的中间产品传给下一阶段的工作人员继续处理，这就类似于工作线操作，这样带来的一个问题是有可能需要进行大量返工，以纠正前面出现的理解错误和实现错误。为了解决这个问题，人们逐渐提出了并行工程、交叉学科群组、交叉功能群组、集成化产品群组（IPT），以及集成化产品和过程开发（IPPD）等，这些都是在产品或者服务的整个生命周期的合适时间处理这类问题的方法。这些技术倾向于要求设计人员和客户要与制造人员、测试人员和用户共同工作，这种工作方式被称作“让所有相关人员坐在一起”，这意味着所有关键的相关人员要支持产品或者服务开发的所有阶段。这样就需要对组织结构和开发性质进行重新定义。传统的功能部门与交叉学科群组的高度交互工作方式严重抵触，因为每个功能部门就是一个过程组，它们可能采用各自不同的工作过程。

至此，我们就可以发现，众多单一学科的过程改进模型已经不适合于现代系统开发需求，不能支持并行工程环境。这就需把多种单一过程方式集成起来，~~提供~~提供共同的组织框架和共同的工作接口，以消除不同功能部门和过程组之间的屏障。

除此之外，从众多 CMM 模型发布和实施至今，人们在开发和应用 CMM 模型时也逐渐发现了成熟度模型存在如下问题：

（1）KPA 的焦点是与传统瀑布过程相关的活动和人工制品：需求规格说明、文档化的计划、质量保证审计和审查，以及文档化的过程。很少有 KPA 关注能够获得真实目标产品的进化结果（比如软件产品）和相关的工程人工制品（用例模型、设计模型、源代码或者可执行代码）。同样，也不关注体系结构/设计过程、评估过程或者部署过程，其中这些过程已经被证明是项目成功的关键。CMM 也忽略同行评审、审查和传统的质量保证“策略”方法。尽管手工评审和审查也许会发现 60% 的错误，但是很难发现体系结构方面的重大问题。并且，没有任何工程师、设计人员、测试人员或者项目管理者说审查和同行评审是项目成功的关键。

（2）另外一个问题是 CMM 的大多数实现都会驱使组织产生更多文档、更多检查点、更多人工制品、更高可追溯性、更多评审和更多计划，并认为输出更厚的文档、更详细的信息和耗费更长时间的会议会更好。实际上，人们普遍认为“越来越多的人工制品”并不是 CMM 的过错，但是对组织当前成熟度的度量也存在问题。CMM 采用基于活动的方法来度量成熟度；如果实施基本项目活动集合，则处于成熟度等级 2。没有任何东西可以限制实施这些活动是否达到预期结果。我更愿意使用结果驱动的程度模式：如果能够在多个项目按照可预测的成本、质量和时间表来重复该过程，则处于成熟度 2；如果可以在多个随后项目中提高成本、质量或者时间中的任何一维，则处于成熟度 3；在多维上的提高和增强则处于更高等级上。实际上，单独使用基于活动或者结果驱动的程度标准都是不完善的；我们需要综合两者方法才能正确度量成熟度。

我们经常看到组织严格按照 CMM 规范来定义过程，以便于建立与满足评估要求的清晰映射。度量框架也应该度量项目性能的实际改进，比如完成成本以及产品预期质量。软件开发标准（比如 RUP 或者 ISO12207）可以成为准确评估的基础，有助于