

中小软件企业质量管理体系研究

摘 要

随着中国加入 WTO，中国软件业面临的国内外两个竞争市场，既是挑战，同时也是机遇，而占我国软件业 70% 左右的中小软件企业面临的形势尤为严峻。缺乏质量管理意识、开发过程混乱、客户满意度低等成为中小软件企业的通病，也严重制约了其发展和壮大。因此，本文就如何提高我国中小软件企业的质量管理水平进行研究。

首先，本文系统地论述了国内外质量管理的理论研究现状和我国中小软件企业质量管理现状以及存在的问题。

其次，研究了软件质量管理的基础理论、方法和体系。从软件质量特性的定义研究入手，分析比较了 CMM、ISO9000 等质量标准。

接着，结合我国中小软件企业质量管理的特点，建立起一个适应性强，可执行性高的中小软件企业质量管理模型，包括组织结构和企业文化、过程管理、技术管理三个层次。通过组织结构和企业文化层次保证质量管理的制度化，通过过程管理层次保证质量管理的规范化，通过技术管理层次保证质量管理的标准化，多方面多渠道地开展质量改善活动。这三个层次以提高软件组织的质量管理水平为目标，共同构成质量管理的铁三角。

最后，进行我国中小软件企业质量管理的对策研究，包括国家的政策推动、企业自身的管控以及通过培训来提高软件工程师的能力和素质。

关键词：中小软件企业；质量管理；组织结构；企业文化；过程管理

Research on Architecture of Quality Management of Small & Middle Sized Software Business

Abstract

After China entered WTO, Chinese software industry has to face internal and abroad competition, it is both challenge and chance. For the small & middle sized software business which accounts for nearly 70% of the software industry, the situation is grim especially. Lack sense of quality management, the disorder of the process of developing, low degrees of satisfactory of customs and so on are the disadvantages of most small & middle sized software business. Therefore, this paper will study how to improve the level of quality management of small & middle sized software business.

Firstly, the thesis reviews the theories and methods of quality software management systematically, analyzes the situation and problems of small & middle sized software business.

Secondly, the thesis studies the basically methods, theories and standards of software quality management, including CMM, ISO9000 etc.

Thirdly, combining with characteristics of small & middle sized software business, a model of quality management is proposed. The structure of model contains three hierarchies, they are framework and culture of organization, management of process and management of technology. This model can make the management of quality standard, institutional and systemic.

Finally, the thesis researches the systematical method to improve the level of quality management of small & middle sized software business. Including the power of government policies, the management and control of business themselves and improvement of the ability and quality of software engineers and so on.

Key words: Small & Middle Sized Software Business, Quality Management, Framework of organization, Culture of enterprise, Management of Process

图表清单

图 2-1 质量特性结构图	7
图 2-2 因果图	8
表 2-1 CMM 关键过程域表.....	10
表 2-2 CMM 级别与软件质量关系表格.....	10
图 3-1 2002 年软件产品结构情况图	17
图 3-2 中小软件企业质量管理模型	21
图 3-3 中小软件企业质量管理模型层次映射图	21
图 4-1 中小软件企业人员组织设置图	25
图 5-1 XP 总体过程图	38

独创性声明

本人声明所提交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得合肥工业大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：

张玉

签字日期：2007年6月13日

学位论文版权使用授权书

本学位论文作者完全了解合肥工业大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权合肥工业大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

(保密的学位论文在解密后适用本授权书)

学位论文作者签名：张玉

导师签名：

梁昌勇

签字日期：2007年6月13日

签字日期：2007年6月14日

学位论文作者毕业后去向：

工作单位：

电话：

通讯地址：

邮编：

致 谢

在这里，首先要感谢我的导师梁昌勇教授，他严谨、求实的治学作风，扎实的理论功底，对科学问题深刻、敏锐的认识和理解，灵活的思路、开阔的视野和宽厚待人的学者风范，对我产生很大影响，使我终身受益。在我攻读硕士学位三年的学习时间里，在研究方向的把握、关键理论指导等方面，梁老师给了我非常重要的辅导。在应用系统的设计、开发以及论文的写作过程中，得到了导师梁老师特别的关心和细致的指导，本文是在梁老师的亲切关怀、热情鼓励和悉心指导下完成的。在此，我以最诚挚的心情向我的导师梁昌勇教授表示最衷心的感谢。我庆幸自己有这样一位导师，在此，也献上我深深的祝福。

此外，也要特别感谢蒋翠清老师和俞家文老师在论文修改方面给予的指导与帮助！

同时，衷心感谢合肥工业大学管理学院的各位领导、老师，他们在很多方面给了我极大的帮助与支持。

另外，还要感谢管理学院信息技术与工程管理研究所的各位老师和同学，他们在我读研的三年里给了许多关怀和帮助，让我在一个求知氛围浓郁的环境下完成了此篇论文，在此对他们真诚的友谊表示感谢。

非常感谢我的家人在我求学生涯中对我的关心、支持和帮助。他们对我的爱护、支持和鼓励，是我不断前进的动力，有了他们的奉献，才使得我的研究工作顺利进行。三年中，还有许多老师、同学、朋友为我提供了很多支持，限于篇幅，虽不一一列出，但我仍然从心底感激他们。

由于本人水平有限，论文中的缺点、错误在所难免，敬请前辈专家们给予批评指正。

作者：张玉

2007年6月

第一章 绪论

2002年中国共产党的十六大提出了“信息化带动工业化，工业化促进信息化”的战略指导思想，软件产业的发展直接影响我国社会信息化的进程。软件产品质量同制造业产品的质量一样，直接关系到经济发展和民众的生命、财产安全。软件质量的低下，带来了大量的问题，如医疗事故、金融混乱、飞行失事等，程序中一个小小的疏漏往往会造成上千万美元的损失甚至人员伤亡。例如，1996年欧洲航天局发射的阿丽亚娜5型火箭，发射40秒后火箭爆炸，发射场上2名法国士兵当场死亡，耗资10亿美元的航天计划严重受挫。事故发生后，专家组的调查分析报告表明，爆炸的根本原因在于惯性导航系统软件中的设计错误^[1]。今天的计算机系统，其硬件质量越来越高，而软件质量问题却越来越突出^[2]。提高软件质量已成为软件业求生存求发展的唯一途径。

1.1 问题的提出

截至2004年9月，我国从事软件和信息服务的企业已超过10000家。但我们要清醒地认识到，和美国、印度等软件大国相比，我国软件行业表现为“小、散、软”的特点。“小”是指国内软件企业规模小。我国绝大多数软件企业的规模不超过50人。“散”是指软件企业的管理存在着严重的漏洞。目前我国绝大多数小软件企业只是软件手工作坊，根本谈不上管理，更不用说软件生产的流程控制。“软”是指大多数软件企业没有自己的核心技术和产品，从而在市场上缺乏核心竞争力。由于管理不善、缺少资金、人才外流等原因，导致企业研发力量薄弱，产品和服务自然也就缺乏竞争力。

中小软件企业中软件产品或服务状况不容乐观，很大程度上体现在其质量管理水平的薄弱上。无论是从产业规模、企业实力、技术水平，还是从创新能力、市场竞争力和产品的市场占有率来看，中小软件企业都处于不成熟阶段，质量管理工作的水平较低，主要表现在以下几个方面：

(1) 开发过程管理不规范。

软件机构是反应型的，缺乏风险分析经验，对软件开发过程中可能遇到的困难准备不足，管理人员往往花费大量的精力去应付突发事件。软件项目的进度和经费预算由于估计得不切实际，所以常常超出计划。在交付时间紧迫的情况下，往往不得不削减软件的功能，降低软件的质量。开发过程的非标准化既不利于组织本身软件过程能力的提高，同时也妨碍了软件开发组织与国际先进开发管理规范的接轨。

(2) 组织结构不合理，质量管理制度不健全。

在中小软件组织中，往往以项目划分人力资源，各个项目组单独作业，接

近于“黑箱操作”，缺乏质量保证小组的监督和管理。并且在组织内部对质量问题认识不够，意识不强，还没有建立起以质量为生命，以质量求发展的企业文化观。

(3)忽视建模等技术手段对软件质量的影响。

数据和编程的质量是软件质量的基础，测试和评审是保证软件质量的重要环节，但在中小软件企业中，常常将软件测试和评审活动忽略，或由开发者本人简单敷衍了事，不仅降低了软件复用的可能性，也为软件质量带来了很大的隐患。

1.2 国内外研究现状

1.2.1 国外软件质量管理研究现状

19世纪50年代后期到60年代，高级语言相继诞生并得到了广泛的应用，软件规模也越来越庞大，结构越来越复杂。随之而来的是软件应用的越来越广泛，软件的质量问题就变得越来越突出。因此如何提高软件的质量成为软件工程研究的一个重点。20世纪70年代中期，美国国防部设专题研究软件项目做不好的原因，发现70%项目的失败是因为管理不善，而非技术实力不够。他们的结论是：管理是影响软件研究开发项目全局的因素，而技术只影响局部。这个结论历经验证并得到举世公认。软件是知识产品，是人类智慧的结晶，软件系统的复杂程度也是超乎想象的，不同于一般的生产过程。提高软件质量已经不再是纯粹的技术问题，而是工程的问题。

此后出现了很多研究如何提高产品质量的理论和方法，最为广泛接受和采用的就是美国卡内基—梅隆大学软件工程研究院(SEI)为了满足美国联邦政府评估软件供应商能力的需求，专门针对软件开发企业设计的软件能力成熟度模型SW-CMM(Software-Capacity Maturity Model)。CMM是一个动态的过程，企业在取得较低级别证书后，可根据更高级别的要求确定下一步改进的方向，申请更高级别的认证。其主要特点是提供了软件企业自我评估的方法和自我提高的手段^[3]。随着对CMM研究的不断深入，SEI又陆续推出了其他基于CMM的模型，包括：人力能力成熟度模型P-CMM；软件采购能力成熟度模型SA-CMM；系统工程能力成熟度模型SE-CMM；一体化生产研制能力成熟度模型IPD-CMM等等。

此外，还有ISO9000和SPICE等标准也在软件质量管理中得到了广泛的应用。ISO9000系列标准是国际标准化组织(ISO)于1987年发布的。目前使用的是2000版的ISO族标准。ISO9000软件质量标准系列为：ISO9001，ISO9000-3，ISO9004-2，ISO9004-4，ISO9002。ISO9001是ISO9000系列标准中软件机构推行质量认证工作的一个基础标准，是在软件设计、开发、生产、安装和维护时

质量保证的参考文件；ISO9000-3 是对 ISO90001 进行改造后，将其应用到软件工业中对软件开发、供应和维护活动的指导文件；ISO9004-2 是指导软件维护和服务的质量系统标准，指导和支持软件产品的维护；ISO9004-4 是近年来公布的很有用的附加标准，是用来改善软件质量的质量管理系统文件。除了 ISO9000 标准系列外，许多工业部门、国家和国际团体也颁布了特定环境中软件运行和维护的质量标准，如：IEEE 标准 729-1983、730-1984、Euro Norm EN45012 等。

1998 年，SPICE(Software Process Improvement and Capability Determination) 项目组织发表用于确定自我能力改造和进行软件供应商能力的国际标准 ISO/IEC15504。该标准覆盖了过程评估、过程改进和过程能力确认等指南和模型。它规定了软件过程评估的框架，可供从事软件策划、检验、控制以及改进软件的获得、提供、开发、运行、进化及支持的组织所使用。

近年来的理论研究证明，在一个软件企业中，如果要保持良性的发展，必须关注人员组织、技术和过程管理三者之间的关系。而近年来越来越多的学者侧重于对软件开发过程的管理和研究，特别是以 CMM 和 ISO9000 系列、SPICE 为代表的过程管理标准，强调企业持续的改进和过程能力成熟度的提高。

然而从各种理论中，我们也发现每一种方法或技术都不是绝对有效的，软件质量的提高应该是一个综合的因素，需要从各个方面进行改进，针对企业的实际情况和管理水平，同时还需要兼顾成本和进度。

1.2.2 我国软件企业质量管理研究现状

相对于国外软件业，我国软件企业起步晚，因此对于软件质量管理方面的研究也滞后于国外，但我国的学者仍在针对我国国情进行着积极的探索和研究。

中国科学院软件研究所的李浩、李明树在 2001 年《一种基于 workflow 控制的软件质量管理模型 SQMM》一文中提出了 SQMM 模型(Software Quality Manage Model)；上海交通大学的肖瑾、周朝民在 2003 年《软件开发中的项目质量管理研究》一文中提出了 PDCS 模型(计划(P)、执行(D)、控制(C)、固化(S))等一系列模型和方法；古蓉、左春在 2003 年提出了事务流程图法在 CMM 过程改进上的应用等。

我国也陆续发布了一系列质量管理的准则和规范，如 1991 年国家发布应用软件质量标准 ISO/IEC9126-1991，后发展为《GB/T16260-1996 信息技术 软件产品评价 质量特性及其使用指南》；我国信息产业部也于 2001 年 4 月发布了《SJ/T 11234-2001 软件过程能力评估模型》和《SJ/T 11235-2001 软件能力成熟度模型》两个标准。国家 863 计划设立了“软件产品国际化示范工程”，旨在促进 CMM 在我国的推广^[4]，北京、上海等地企业通过 CMM 认证，政府还会有 20 万—80 万不等的奖励，鼓励软件企业建立质量管理体系，提高质量管理水平。能力成熟度模型的引进势必为我国的软件产业提供一个更有效的设计和认证思

路，为进一步落实软件质量保证政策提供有效途径^[5,6]。

但是纵观我国软件质量管理方面的著作、文献和研究课题，我们发现并没能针对我国软件行业中 70%左右的中小软件企业提出系统的科学的质量管理体系。传统的质量保证标准或方法实施步骤繁琐，需要消耗大量的人力物力资源，并且执行的周期较长，往往超出了中小软件企业的可承受范围或是将质量保证流于形式，不能从根本上改善质量现状。

本文认为，理论来源于实际而高于实际，能为改善我国软件行业，特别是中小软件企业的质量管理状况，提高质量管理水平做出贡献的理论研究具有更深刻的现实意义和价值。关注我国软件业现实问题的软件质量研究将是软件质量理论研究的重点和发展趋势，也是本文的立意所在。

1.3 文章的主要内容及结构安排

本文研究的主要内容包括：

(1)软件质量管理的基础理论方法。研究典型的软件质量管理理论、标准、方法，分析其对我国中小软件企业的适用性。

(2)从企业规模、产品服务对象、内部分工等方面分析概括我国中小软件企业的特点，质量管理中存在的问题，提出质量管理的影响要素。

(3)结合质量管理基础理论方法和我国中小软件企业特点，建立质量管理模型，包括设计组织结构和文化、规范过程管理和开发技术管理三个层次。其中，组织结构和文化的设计包括软件组织中人员安排、职责划分和企业文化的建立三个方面；规范过程管理，包括制定软件质量保证计划、采用剪裁化的质量标准、采用极限编程的开发方法，文档管理和软件质量评估五个方面；技术管理层次，包括数据及编程质量管理、建立测试与评审机制，以及标准建模语言 UML 的使用三个方面。

(4)我国中小软件企业质量管理对策研究，从国家、企业、个人三个层面剖析改善软件质量的有效手段。

各章节具体内容如下：

第一章：介绍本论文选题的背景和意义以及国内外在软件质量管理理论上的研究现状，总结我国中小软件企业质量管理的现状和存在的问题，并阐述本论文的结构和内容安排。

第二章：分析软件质量管理中的基础理论和方法，为中小软件企业质量管理模型提供理论依据和支持。

第三章：分析我国中小软件企业的特点和影响其软件质量的各种因素，结合第二章内容，给出中小软件质量管理模型的三层架构。

第四章：介绍中小软件企业质量管理的必要活动——组织结构和文化的安排及设计。

第五章：介绍中小软件企业质量管理的核心活动——过程管理。

第六章：介绍中小软件企业质量管理的基础活动——技术管理层次。

第七章：我国中小软件企业软件质量管理的对策研究。

第二章 软件质量管理基础理论方法

随着本世纪六十年代“软件危机”的出现，软件质量成为影响软件业生存、发展的重要问题。人们也逐渐对软件质量开展了深入的理论研究，从 1976 年 Boehm 提出软件质量的特性开始，各国学者和软件机构又陆续研究了软件质量管理的工具、方法及标准，并强调了测试与评审在改善软件组织质量水平中的重要作用。

2.1 软件质量的定义

McCall 等认为，特性是软件质量的反映，软件属性可用做评价准则，定量化地度量软件属性可知软件质量的优劣。1976 年 Boehm 提出了软件质量特性，形成了现在软件质量特性的雏形。1978 年 McCall 提出了体现软件质量的 11 个特性，包括使用性、可测试性、正确性、维护性、可靠性、移植性、效率、重用性、完整性、互操作性和适应性(灵活性)。美国国家标准 ANSI / IEEE Std 729-1983 定义软件质量为“与软件产品满足规定的和隐含的需求的能力有关的特征或特性的全体”。它包含了四层含义：能满足给定需要的特性之全体；具有所希望的各种属性的组合的程度；顾客或用户认为能满足其综合期望的程度；软件的组合特性，它确定软件在使用中将满足顾客预期要求的程度。

随着市场经济的发展，卖方市场向买方市场的转变，本文认为影响软件质量的特性分为三组，反映用户在使用软件产品时的三种不同倾向，核心是满足客户需求。这三种倾向是：产品运行、产品修改和产品转移。信息系统作为一个产品，也可以参照这三种倾向来定义，如图 2-1 所示。

本文认为，软件质量应以客户满意为中心，以能否实现客户需求为成功的标准，软件质量就是软件产品和服务能够实现客户明确和潜在需求，能够完成客户既定目标及后期维护的能力。它包括四个维度：

(1)以客户满意为中心。顾客就是上帝，顾客的满意就是软件产品的最终目的。如果不能使客户满意，功能再强大的软件也是低质量的产品。在软件产品的生命周期内，要时时体现出这一原则，才能体现软件的质量。

(2)符合客户需求。实际的需求包括用户明确说明的和其隐含的需求。由于专业知识、工作时间等条件的限制，有些客户不能完全全面的表达自身的需求，这就要求软件项目的承担者能够利用丰富的经验和负责的工作态度，在开发过程中引导客户，不断完善软件的功能，更好地实现客户的明确需求和隐含需求。

(3)软件产品不仅要满足用户当前的要求，还要考虑到维护工作。在它的生命周期内必须保证维护工作简单，要具备功能扩展能力，且修改容易。

(4)软件开发必须遵循规定的标准。开发过程的标准化有利于提高软件质量水平。

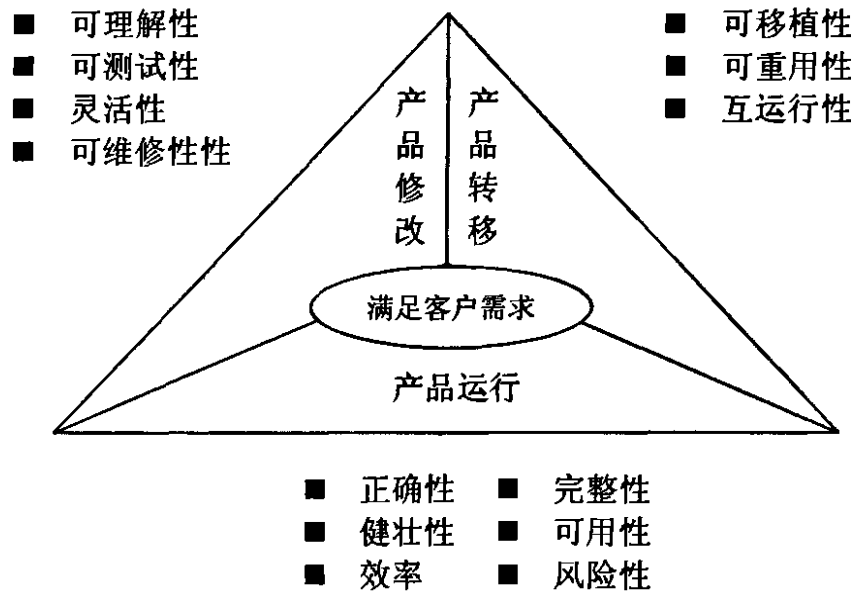


图 2-1 质量特性结构图

可见，从用户最感兴趣的角度来说，软件质量可以从三个方面来把握：如何使用软件、使用效果如何、软件性能如何；从软件开发的团队的角度来说，不仅要生产出满足质量要求的软件，也要能够运用最少的资源、以最快的进度生产出质量最优的产品；从软件维护者的角度看，要对软件维护方面的特性感兴趣；对企业的管理层来说，注重的是总体效益和长远利益，就是说质量好的软件一般可以帮助企业扩大市场；反之，质量差的软件一般会造成企业市场萎缩。

2.2 软件质量管理的工具、方法

软件质量管理的方法主要由质量管理的老七种方法和新七种方法演变而来，其中，老七种方法是指：因果图、流程图、直方图、检查单、散布图、排列图、控制图法。新七种工具是指：关联图法、亲和图法(KJ 法)、系统图法、矩阵图法、矩阵数据分析法、过程决策程序图法(PDPC 法)、矢线图法。老七种工具的特点是强调用数据说话，重视对制造过程的质量控制；而新七种工具则基本是整理、分析语言文字资料(非数据)的方法，着重用来解决全面质量管理中 PDCA 循环的 P(计划)阶段的有关问题。

在软件质量管理中，强调“无缺陷”的产品质量方针，越来越多的将统计控制理论和度量理论的一些基本方法融入到质量管理工作中来，例如：SPC(统计过程控制)、头脑风暴法、鱼刺图、PARETO 原理、层次图等方法。质量管理

方法在中小软件企业中的应用强调以简便性、易实施性和高效性为原则。

(1)头脑风暴法(Brainstorming)。

头脑风暴法的一般步骤可以概括为::

- 1) 确定质量问题。
- 2) 尽可能找出影响质量问题的因素。
- 3) 找出各原因之间的关系, 在因果图上以因果关系箭头连接起来。
- 4) 根据对结果影响的程度, 将认为有显著影响的因素标出来。
- 5) 在因果图上标出必要的信息。

利用头脑风暴法可以分析缺陷产生原因, 并对原因进行系统整理、归类, 即先放开思路, 进行开放式、创造性思维, 然后再根据概念间的层次关系整理成型。

(2)因果图

鱼刺图(Cause-effect diagram)又叫鱼骨图、石川图等, 最先由日本东京大学石川馨教授于 1953 年提出。它是用来表示质量波动特性与其潜在原因的关系, 即表达和分析因果关系的一种图表。主要用来分析和说明各种因素和原因如何导致或者产生各种潜在的问题和后果, 其分析原理如图 2-2 所示:

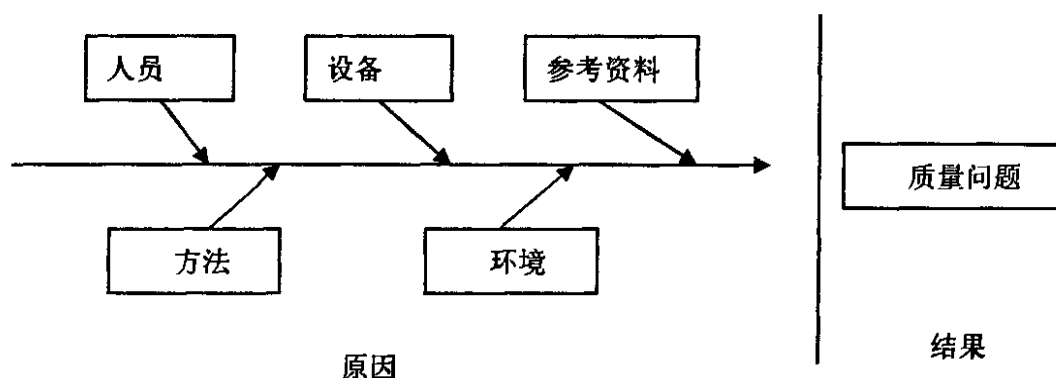


图 2-2 因果图

利用因果图法可以对项目进行分析, 确定需要监控的关键元素, 设置合理的见证点(W 点)、停工待检点(H 点), 并制定质量标准。

(3)PARETO 原理

Pareto 分析是一种识别消耗了最多成本的少部分的角色的统计分析方法。J.M.Juran 首先提出了术语 vital few 和 trivial many, 并应用到 Pareto 原理中。Pareto 原理强调了将精力集中在少数重要的事情上(vital few), 而不是在多数琐碎的事情上(trivial many)。

产品质量是由质量形成过程中许多因素共同作用的结果。质量有了问题, 特别是主要问题, 就要进一步找原因, 并针对原因采取措施, 才能解决问题。有些质量问题的原因一目了然, 而有些质量问题的原因, 就必须借助图形的方

式来分析，这样才能帮助我们理清思路，从而对症下药。

而发现问题后，识别关键、把握关键是项目成败的关键，也是我们日常工作的重要内容。如不要为过于细节的讨论耗费太多时间。在 CMM 模型中提到的“关键成功要素”(Critical Success Factors)实际上就是一种基本的统计控制方法。

2.3 软件质量标准

随着软件规模的扩大，客户对质量要求的提高，软件开发人员尝试设定一系列质量标准来规范开发过程，改善软件质量，主要包括 CMM/PSP/TSP 体系、ISO9000 系列、ISO/IEC15504 体系以及近年来开始运用的六西格玛管理法。以下将按照标准产生的时间序列予以详细的介绍和综合比较。

(1)CMM /PSP/TSP 流派

软件能力成熟度模型(CMM)是美国软件工程研究所(SEI)首先提出的，基本思想是基于已有 60 多年历史的产品质量原理。CMM 的本质是通过软件开发过程中一些关键域的精确定义，使软件开发各环节从一个不确定的“黑箱”操作过程变为可视、可对偏差随时控制的“透明”操作过程^[7,8]。这样一来，就可以把软件开发及生产过程中成功或失败的经验教训变成今后可以借鉴和吸取的依据，大大加快软件生产的成熟度的提高。在软件开发过程中，一些关键的过程域(KPA)可以被识别出来，成为某些 CMM 模型方法中的衡量基准。利用过去软件工程发展的成果，侧重这些关键过程域的实施，将会有效地建立一个过程，加快软件企业成熟度的提升。

CMM 一共有 5 级，18 个关键过程域(KPA)，52 个目标，300 多个关键实践^[9]。纵观整个模型的 18 个关键过程域中，有 9 个是管理方面的，5 个是组织方面的，4 个是工程方面的^[10]。CMM 的各等级特点和包含的关键过程域如表 2-1 所示。

采用 CMM 标准不仅可以提高企业的质量管理水平，而且也是供应商对软件承包商的资格考核的重要标准之一。同时，SEI 也对采用 CMM 后对于软件质量的提高数据进行了统计，如表 2-2 所示。

由于 CMM 并未提供有关实现 CMM 关键过程域所需的具体知识和技能，因此，SEI 研究与开发了个体软件过程 PSP(Personal Software Process)和群组软件过程 TSP(Team Software Process)，形成 CMM/ PSP/ TSP 体系。

PSP 为基于个体和小规模群组软件过程的优化提供了具体而有效的途径，例如如何制订计划，如何控制质量，如何与其他人相互协作等等。它指导软件工程师如何设计、跟踪自己的工作，利用已定义的和已度量的过程，建立可度量目标，根据目标跟踪性能，如何从一开始就控制质量，如何分析每个工作的

表 2-1 CMM 关键过程域表

过程能力等级	特点	关键过程域
1. 初始级 (Initial)	软件过程是无序的，对过程几乎没有定义，成功取决于个人的努力。管理方式属于反应式。	无
2. 可重复级 (Repeatable)	建立了基本的项目管理过程来跟踪费用、进度和功能特性。制定了必要的过程规则和纪律，能重复早先类似应用项目取得的成功。	需求管理 软件项目计划 软件项目跟踪和监督 软件子合同管理 软件质量保证 软件配置管理
3. 已定义级 (Defined)	已将软件管理和工程两个方面的过程文档化、标准化，并综合成该组织的标准软件过程。所有项目均使用经过批准、剪裁的标准软件过程来开发和维护软件。	组织过程定义 组织过程焦点 培训大纲 集成软件管理 软件产品工程 组织协调 同行专家评审
4. 已管理级 (Managed)	收集对软件过程和产品质量的详细度量，对软件过程和产品质量都有定量的理解和控制。	定量过程管理 软件质量管理
5. 优化级 (Optimizing)	过程的量化反馈和先进的新思想、新技术促使过程不断改进。	缺陷预防 技术变更管理 过程变更管理

表 2-2 CMM 级别与软件质量关系表格

软件过程成熟度等级	每千行软件的缺陷数目	软件准时提交的百分比	每人每月生产的程序行数	软件需要返工的百分比	平均软件失效时间(近似)
初始级	大于 10	≤ 50	Z	≥ 45	2 到 60 分钟
可重复级	小于 10	90	1.5Z	20	1-160 小时
已定义级	小于 1	99	2.5Z	10	不确定
管理级	小于 0.1	降低开发时间到 1/2	5Z	5	不确定
优化级	小于 0.01	降低开发时间到 1/4	10Z	≤ 2	近似完全可靠

结果，如何利用这些数据结果调整和改进下一项目的过程。PSP 的策略概括来讲，在开始阶段对整个过程进行计划，在结束阶段对收集的数据进行分析，以便调整过程。具体就是：①收集过程中涉及的数据(时间、缺陷等)，提高软件项目的计划和跟踪能力；②涉及和代码均进行回顾，以尽早发现并排除缺陷；③提炼 CMM 的 12 个关键过程域(KPA)作为指导。

TSP(Team Software Process 小组软件过程或团体软件过程)指导项目组中的成员如何有效地规划和管理所面临的项目开发任务，并且告诉管理人员如何指导软件开发队伍始终以最佳状态来完成工作。它所采用的方法是对群组软件开发过程的定义、度量和改进。TSP 实施集体管理与自我管理相结合的原则，最终目的在于指导所涉及的一切人员如何在最少的时间内，以预定的费用生产出高质量的软件产品^[11]。CMM 提供了有效工程工作的全局过程提高框架；PSP 提供了软件工程师一致的已定义的、已计划的、已度量的所需的规范；TSP 结合了集成产品小组与 PSP 和 CMM 方法以产生有效的团体。CMM、PSP、TSP 偏重于管理、组织，在整个软件开发过程中，CMM 提供了“做什么”，而 PSP 和 TSP 则提供了“如何做”，因此为了实现软件组织软件能力成熟度的不断提高，软件组织必须通过在早期阶段先实施 PSP、TSP，而后实施 CMM 的循环过程。

(2)ISO9000 系列

ISO9000 系列标准是国际标准化组织(ISO)于 1987 年发布的。目前使用的是 2000 版的 ISO 族标准。ISO9001《质量体系要求》是其中之一，它规定了质量管理体系的要求，包括管理职责、资源管理、产品实现、测量分析和改进等各个方面，可供组织证实其有能力稳定地提供满足顾客和适用于法律法规要求的产品，鼓励组织在建立、实施和改进质量管理体系时，采用过程方法，以提高顾客满意度，达到质量管理体系的持续改进的目标。

最初的软件质量保证系统是在 70 年代由欧洲首先采用的，其后在美国和世界其他地区也迅速地发展起来。目前，欧洲联合会积极促进软件质量的制度化，提出了如下 ISO9000 软件标准系列：ISO9001、ISO9000-3、ISO9004-2、ISO9004-4、ISO9002。这一系列现已成为全球的软件质量标准。

ISO 9000 系列标准现有 20 个标准，分为五类：质量术语标准：ISO8402-1994；质量保证标准；质量管理标准；质量管理和质量保证标准的选用和实施指南；支持性技术标准。

ISO 9000 的 2000 版的认证只是对从事软件工作能力最低的认可，是对软件质量最低的保证，并且对软件产品的适用性也不是很强。

(3)ISO/IEC15504 体系/SPICE

1998 年 SPICE(Software Process Improvement and Capability Determination)项目组织发表用于确定自我能力改造和进行软件供应商能力考核的国际标准 ISO/IEC15504。ISO/IEC TR 15504 包括以下内容：概念与介绍指南；过程和

过程能力的参考模型；完成一次评价；完成评价指南；评价模型与指标指南；评价者资格指南；用于过程改进指南；用于确定供应者的过程能力指南。

该标准由 9 个部分组成。第 1 部分阐述了标准构成，并为选择和使用提供了指导；第 2 部分规定了一个用于过程评估和过程能力的二维参考模型；第 3 部分规定了评估要求；第 4 部分对不同评估环境给予指导；第 5 部分提供了过程评估示范模型；第 6 部分阐述了评估员的能力、教育状况、培训和经历及其表述方法；第 7 部分阐述了如何确定评估输入和运用评估结果，以及过程改进实例；第 8 部分阐述了如何确定各种软件承包方，并规定了各种情况下的过程能力测定。第 9 部分收入所有术语并给出了定义。

ISO/IEC TR 15504 不仅重视软件过程，还关注人员、技术、管理、质量、用户支持和软件开发维护等问题。它可用于以下三个方面：①能力评定，即帮助软件组织在购买软件产品时参照国际公认标准，衡量软件供方的潜在软件能力；②过程改进，即帮助软件组织改进自己的软件开发和维护过程；③能力自我评估，即帮助软件组织确定自己承担新的软件项目的能力。其提供的软件过程评估方法主要用于过程改进和过程能力测定，为软件组织采用共同的评估方法提供了基础，使得基于不同但相互兼容的模型和方法的评估结果，可以进行某种程度的比较。

(4)六西格玛管理法

六西格玛(Six Sigma)管理法是一种以数据为基础，追求完美的质量管理方法。统计学用西格玛(希腊字母 σ 的中文译音)来表示标准偏差，即数据的分散程度。对连续可计量的质量特性，用" σ "度量质量特性总体上对目标值的偏离程度。六个西格玛可解释为每一百万个机会中有 3.4 个出错的机会，即合格率是 99.99966%。

六西格玛管理法是全面质量管理的继承和发展，其核心是，将所有的工作作为一种流程，采用量化的方法分析流程中影响质量的因素，找出最重要的因素加以改进从而达到更高的客户满意度，即采用其 DMAIC 模型对组织的关键流程进行改进。这个模型的五个阶段分别是：D(定义)，M(评估)，A(分析)，I(改进)，C(控制)，与其他许多改进方法一样，DMAIC 模型也是建立在 PDCA 循环的基础上的。而 DMAIC 又由下列四个要素构成：最高管理承诺，有关各方参与，培训方案和测量体系。因此，六西格玛管理法为组织的质量管理工作带来了新的，垂直方法体系。

现在，六西格玛以自身的质量活动为基础，将概念和工具映射到软件系统开发的各个方面，如：传统的瀑布模型，快速应用开发，原型法，遗留系统的支持等等，改进了软件过程，从而提高了软件质量。

(5)中小软件企业实施各种标准的困难

CMM 他仅指明该做什么，而没有指明如何做，它不是方法论。但我们在学

习 CMM 时,可以从中学到分析问题的方法。CMM 为改善整个企业的软件过程提供了指南,而并非针对某个具体项目^[12]。CMM 并不能保证在这个过程框架下,产品开发百分之百的成功。产品的成功是多种要素的组合,例如市场等要素。并且,CMM 是针对大型软件企业(500 人以上)的,基于 CMM 的过程改善投资力度大、周期长,每一级评估的费用需要几十万甚至上百万,评估周期约需要 12~30 个月^[13]。对中小型的软件企业(50 人以下)需要根据自身条件和需要进行裁减。

ISO 面向合同环境,站在用户立场,要求对质量要素进行控制,规定了质量管理体系的最低标准。其广泛使用于各个行业,对软件行业的针对性较弱。其审核结果也只有“通过”或“不通过”两种,不能保证对中小软件企业的质量管理水平进行动态的循序渐进的改进。

ISO/IEC15504 体系/SPICE 是在包括 CMM 在内的一些软件过程改进模型的基础上制定出来的。如果要评估软件组织的软件过程更适合开发哪些类别产品,或针对某种特定软件产品评估由哪个组织来承担风险会更小,则采用 SPICE 方法针对各个基本软件过程进行评估更为适当^[14]。可见,SPICE 侧重于过程和风险评估,对于中小软件企业质量管理水平的提高针对性不够。

综上,各种典型的软件质量标准或方法体系有着自身的适用范围和条件,并不是“放之四海而皆准”的通用准则和灵丹妙药。中小软件企业质量管理水平的提高不仅依赖于采用根据自身资源和特点剪裁化的质量标准来保证过程的规范性,更需要以质量管理为核心的组织结构和文化理念,以及围绕提高软件质量而采用的技术手段。

2.4 测试与评审

产业界的大量研究(TRW 、 Nippon Electric 和 Mitre Corp 以及其它一些公司)表明设计活动引入的错误占软件过程中出现所有错误(和最终的缺陷)数量的 50%到 65%。根据 IBM 的研究表明,假定在分析阶段发现的错误其改正成本为 1 个单位的话,那么在测试之前(设计编码阶段)发现一个错误的修改成本约为 6.5 个货币单位,在测试时(集成测试,系统测试和验收测试)发现一个错误的修改成本约为 15 个货币单位,而在发布之后(已经交到用户手上)发现一个错误的修改成本约为 60 到 100 个货币单位。同样该比例也适用于发现一个错误需要的时间。可见,通过软件测试和评审及早发现并修正软件中的错误,不仅能提高软件质量,而且能大幅度的降低开发成本。软件测试和评审是软件开发的重要环节,同时也是软件质量保证的重要环节。

2.4.1 软件测试

软件测试就是用已知的输入在已知环境中动态地执行系统(或系统的部件)。测试的种类一般包括单元测试、模块测试、集成测试和系统测试。测试的主要方法包括:

(1)黑盒测试

黑盒测试检验是否符合系统需求,也称功能测试或数据驱动测试。它是在已知产品所应具有功能的基础上,通过测试来检测每个功能是否都能正常使用。在测试时,把程序看作一个不能打开的黑盒子,在完全不考虑程序内部结构和内部特性的情况下,测试者通过程序接口进行测试,它只检查程序功能是否能按照需求规格说明书的规定正常使用,程序是否能适当地接收输入数据而产生正确的输出信息,并且保持外部信息(如数据库或文件)的完整性。

常见的黑盒测试包括:集成测试和系统测试。

集成测试是在单元测试的基础上,将所有模块按照设计要求(如根据结构图)组装成为子系统或系统,进行集成测试。实践表明,一些模块虽然能够单独地工作,但并不能保证连接起来也能正常的工作。程序在某些局部反映不出来的问题,在全局上很可能暴露出来,影响功能的实现。

系统测试的目的在于通过与系统的需求定义作比较,发现软件与系统定义不符合或与之矛盾的地方。系统测试的测试用例应根据需求分析说明书来设计,并在实际使用环境下来运行。系统测试的内容极其广泛,包括功能测试、协议测试、性能测试、压力测试、容量测试等等。

(2)白盒测试

也称结构测试或逻辑驱动测试,它是在知道产品内部工作过程的情况下,按照程序内部的结构测试程序,检验程序中的每条通路是否都能按预定要求正确工作,而不顾它的功能。白盒测试的主要方法有逻辑驱动、基路测试等。

(3)灰盒测试

灰盒测试介于白盒与黑盒二者之间,关注输出对于输入的正确性,同时也关注内部表现,但这种关注不像白盒测试那样详细、完整,只是通过一些表征性的现象、事件、标志来判断内部的运行状态,有时候输出是正确的,但内部其实已经错误了,这种情况非常多,如果每次都通过白盒测试来操作,效率会很低,因此需要采取这样的一种灰盒的方法。

值得强调的是最后一个步骤——验收测试。软件验收测试是产品最终交付到用户之前的最后一道防线,有着举足轻重的地位。然而,做好软件测试却是不容易的,一方面要求测试者掌握软件开发的技能和软件测试方面的技能;另一方面产品必须给予测试充分的独立性和资源保证。

2.4.2 软件评审

需要强调的是,软件质量评审并不是在软件开发完毕后进行评审,而是在

软件开发的各个阶段都要进行评审。因为在软件开发的各个阶段都可能产生错误，如果这些错误不及时发现并纠正，会不断地扩大，最后可能导致开发的失败。软件评审是相当重要的工作，也是目前国内中小软件企业开发中最不重视的工作。在进行软件质量评审时，需要注意：

(1) 设定评审目标

软件质量评审的目标是：发现任何形式表现的软件功能、逻辑或实现方面的错误；通过评审验证软件的需求；保证软件按预先定义的标准表示；已获得的软件是以统一的方式开发的；确保项目更容易管理。

(2) 制定评审准则

评审的准则包括：评审产品，而不是评审设计者(不能使设计者有任何压力)；会场要有良好的气氛；建立议事日程并维持它(会议不能脱离主题)；限制争论与反驳(评审会不是为了解决问题，而是为了发现问题；指明问题范围，而不是解决提到的问题；展示记录(最好有黑板，将问题随时写在黑板上)；限制会议人数和坚持会前准备工作；对每个被评审的产品要建立评审清单(帮助评审人员思考)；对每个正式技术评审分配资源和拟定时间进度表；对全部评审人员进行必要的培训。

(3) 设计评审内容

评审的内容主要围绕产品质量的特性展开，包括评审软件是否具有互连性、复用性、可测试性、可移植性、互换性、可扩充性、可靠性、可修改性以及性能实现情况、操作特性实施情况、保密措施实现情况、软件的规格说明是否合乎用户的要求等内容。

2.5 本章小节

本章总结了不同学者不同机构对软件质量特性的定义，并给出了本文对软件质量的定义；接着总结了软件质量管理的发展历程和软件质量管理的方法体系和已被国际软件业接受并使用的质量标准体系，同时也分析了中小软件企业采用这些标准时面临的问题；最后强调了测试和评审活动的定义、作用以及对中小软件企业质量改进的重要性。

第三章 中小软件企业的质量管理模型的建立

美国软件工程专家 Brodman 和 Johnson 认为少于 50 人的为小型企业, 50~200 人的为中型企业, 200 人以上的为大型企业。根据中华人民共和国信息产业部《2002 年软件产业发展状况》统计, 截至 2002 年, 我国拥有软件研发人员 15.7 万人, 分布在 4700 家各类软件公司。其中, 67% 的软件企业员工少于 50 人, 26% 的软件企业在 50~300 人之间; 软件企业员工 1000 人以上的企业有 25 家, 50~1000 人的有 45 家, 200~500 的有 170 家。因此在我国, 大多数软件企业属于中小企业, 提高中小企业的软件水平对整个软件业来说具有深刻的意义。如何针对中小软件企业在质量管理上的特点, 并通过设立软件组织结构, 实施过程管理和采用技术手段等质量管理方法予以改善是目前研究的重点问题。

3.1 我国中小软件企业的特点

据有关资料统计, 2000 年我国从事软件开发、生产、销售、维护和服务的企业已发展到 10000 家以上, 其中专门从事软件开发的企业达 3000 多家^[15]。软件业内一位权威人士在分析我国软件质量时指出, 我国软件企业绝大多数是 50 人左右的民营小企业, 管理大多具有随意性, 其企业制度上的不完善是我国软件企业与发达国家的差距所在^[16]。

由于规模小, 资源有限, 我国的大部分中小型软件企业还处于作坊式的生产状态, 管理水平低下, 开发不规范; 短期行为多, 产品及过程可重用性差, 缺乏可持续发展的潜力; 软件开发过程是无序的, 软件组织一般不能提供开发和维护软件的稳定环境。在中小软件企业中, 软件项目的成功完全依赖于一个杰出的管理人员或一个有经验的、有战斗力的软件开发队伍, 但是当他们离开该项目后, 他们的这种能使过程稳定的影响力也就随之消失。在开发遇到危机时, 项目组就抛弃临时拼凑的软件开发过程, 并且组织的过程能力是不可预测的, 因为随着进度、预算、资源消耗和产品质量一般是不可预测的。就我国中小软件企业来说, 具体的表现出以下特点:

(1) 规模小, 人力、物力资源有限

中小型软件企业的规模特征首先是企业的人员较少, 资金规模相对小。资源是进行软件过程改善的先决条件, 不同规模企业进行软件过程改善, 其资源的占用并不是一个简单的线性关系, 如 500 人的大型软件企业和 50 人的软件企业实施软件过程改善所需的投入绝对不是 10:1 的关系, “麻雀虽小, 五脏俱全”, 除对软件过程改善培训和支持以外, 两种规模的企业进行过程改善的核心活动

基本上是一样的。不同的是，大型企业可以大手笔和大投入，中小型企业就必须精打细算地利用有限的资源做最需要的软件过程改善。

(2)软件生产领域的用户有限

中小型企业的人员和资金规模的限制，使很多的企业通常采用特性软件开发的开发类型，即为特定的用户定制软件产品，为特定用户对象做系统集成开发，并且产品的领域也较为单一。根据中华人民共和国信息产业部《2002年软件产业发展状况》调研结果显示，软件企业还是的主要产品以应用软件为主，其次是工组软件，中间件软件的份额最少。2002年软件产品的结构状况如图3-1所示：

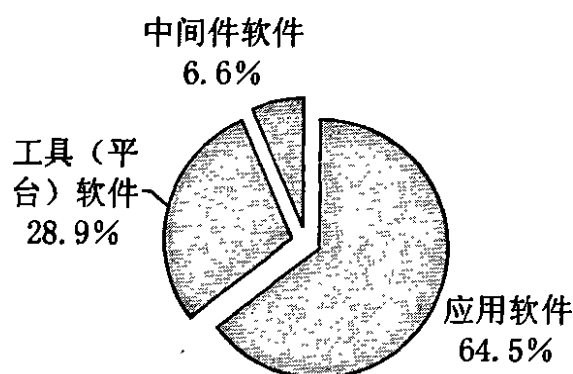


图 3-1 2002 年软件产品结构情况图

(3)企业内部分工不明确

中小型软件企业和大型软件企业的区别还体现在企业文化的显著差别。在中小型软件企业中，软件工程人员可能参与软件生产的各个方面^[17]，每个人都是“全才”，他(她)可以进行需求分析，同时也可以进行设计、开发和测试。在软件生产过程中，中小型企业在职责划分上更侧重于以人员为中心而不是以角色为中心，软件开发缺乏层次性。分工的不明晰，责、权、利不能很好的划分为质量管理带来了组织结构上的缺陷。

(4)软件生产模式不规范

在中小型企业中的软件生产，项目的成功更多地依赖于少数人员的杰出技术能力和项目管理能力，企业缺乏明确定义的软件过程。成功项目的经验不能得到最大限度的继承，软件生产的可重复性相对较差。在中小软件企业的软件项目中，一般在启动阶段缺少质量保证计划，在开发过程中也没有定期的进行测试和评审，因此突发状况较多，软件质量安全不能保障。

(5)企业员工质量管理意识淡薄

在大多数中小软件企业中，重技术轻管理的思想根深蒂固。企业负责人和技术骨干往往只重视技术，认为只要工具使用熟练，精通程序设计语言，就能开发出顾客满意的软件，对质量管理问题的重视不够。

(6)开发人员流动性大

在中小型软件企业中，人员变更相对频繁。工资低、压力大是我国大部分程序员的生活状态，也是造成软件人才容易跳槽甚至离开软件行业的原因。而中小软件企业的项目成功关键往往是凭借个人的杰出能力，一旦核心人物离开项目组，质量管理就会陷入一片混乱。这也决定了软件过程改善就不能以具体的某个人为中心，而是以软件过程及其活动为中心。

综上所述，我国中小软件企业的规模小，人数少，软件生产领域和使用对象有限，企业人员可能参与软件生产的各个方面，并且项目的成功更多地依赖于少数人员的杰出技术能力和项目管理能力，项目的成功经验难以复制，软件生产模式不规范，造成软件生产的可重复性相对比较差。在人员变更频繁的市场环境中，软件质量不稳定。中小软件企业的特点是本文质量管理体系建立的基础和依据。如何让中小软件企业扬长避短，充分利用已有资源提供更高的质量，创造更大的利益也是本文的研究目的。

3.2 中小软件企业质量管理的影响要素

据有关专家的分析，中国软件业在基础研究、技术前瞻性、市场前景的把握上都要比印度更胜一筹，在软件发展的社会环节、地区发达程度上，印度也落后中国十年。但就在最近的十年中，印度却一跃而成为除了美国以外最大的软件出口国，它所生产的软件产品远销 75 个国家，出口量是中国的 100 倍之多。那么，是什么因素促使印度软件业由小到大，由弱到强呢？

在创业初期，印度软件产业的规模与我国 80 年代的中关村相似，可是，不论公司规模大小，都能做到在文字方面打破民族偏见，在经营方面以 IBM 等公司为参照，在开发过程中更是采用严格的国际标准。例如印度的 Infosys 公司^[18]，这是一家从作坊式中小软件企业发展到如今拥有约 4500 名员工，并在 1999 年通过 CMM5 级评估的软件公司。在 1991 年前该公司软件开发没有执行标准，软件开发水平取决于开发人员个人的努力和高级主管对未来的预测。随着软件开发领域的扩大，公司决策人意识到在公司中必须实施软件规范。于是公司于 1992 年开始按 ISO9001 标准规范软件开发过程，1993 年开始 CMM 评估，由 1 级逐渐提升到 5 级。大大提高了软件开发能力，保证了向美国和欧洲软件出口的高增长。Infosys 公司的软件质量管理专家 K.Dinesh 先生介绍到：质量管理是否实施成功，取决于人的努力。在实施质量管理的过程中，职工要改变原来的习惯，每个人都要制定自己的目标。软件质量由人、技术、过程决定。三个因素如同一张桌子的三条腿，每条腿都很重要。

国外软件业发达的很重要原因就是：无论规模大小，绝大多数企业都按照规范化的工作方法管理软件循环过程，始终把最终用户放在软件产品供应优化和质量控制的中心，建立严格的项目管理和质量保障体系，通过技术创新塑造

企业的核心竞争力，并不断加强对员工的质量生命观。

通过对我国中小软件质量管理特点和印度软件企业通过质量管理将软件产业做大做强案例分析，总结得出质量管理的影响要素包括以下三个方面：

(1)组织结构和企业文化

要求以质量为中心，并在软件企业建立起以质量保证组为核心，以测试和评审组为有效支撑的组织结构，培养软件组织员工的质量意识和质量控制能力，逐渐将质量生命观融入企业文化中去，从员工思想上保证质量管理活动的开展，而不是流于表面和形式。

(2)过程管理

软件质量管理的过程比结果更为重要。在软件开发前周密地进行质量保证计划，在开发过程中采用合适的质量标准，并实施地开展软件配置管理，将测试与评审活动贯穿于软件生命周期，才能保证最终软件产品的高质量。

(3)技术手段

包括软件分析、设计、实现的技术方法和技术手段等。选用不同的技术方法对软件产品质量有明显的影响。技术因素包括软件开发技术和应用领域技术等。软件开发技术涉及软件开发的诸多方面。编程技术是构成软件产品的基础，只有将技术与测试评审等环节有机的结合，才能从根本上开发出质量更好的软件产品。

在一个软件企业中，如果要保持良性的发展，必须关注人员组织，开发过程和技术三者之间的关系。人员组织是流程成功实施的保障，好的组织结构能够有效的促进流程的实施，尤其是对软件企业来说，人是最宝贵的财富，他们是技术的载体；开发过程对于产品的成功有着关键的作用，一个适合于组织特点和产品特点并且规范化的开发过程能够极大的提高产品开发的效率和产品质量，反之则会拖延产品开发进度，并且质量也无法得到保证；而技术层次的质量管理也对软件整体质量起着关键的作用。

3.3 中小软件企业质量管理模型架构

20世纪60年代时，软件产品的问题日益严重，软件产品或者不能按时交付，或者到时交付的软件产品错误百出，即所谓的“软件危机”。此后，人们提出了软件工程。软件工程是科学和数学的应用，使计算机设备的能力借助计算机程序、过程和有关文档成为对人类有东西，其核心问题是保证质量、按时完成软件项目。在这之前，普遍认为使用先进的开发工具加上充分的测试就可以解决软件的质量问题。后来，渐渐发现，更多的问题原来是起源于人的，也就是说员工管理的问题^[19]，如果缺乏有效的管理，再先进的开发工具也无法取得预期的效果，软件测试也是无法从根本上保证软件质量的，软件测试仅能发现一部分错误，并不能保证软件中没有错误，它只是软件质量保证的一个环节，软

件产品是过程的产物，所有的工作都是通过过程来完成的，真正的软件质量要靠各个开发环节来保证，软件的质量管理在于软件开发过程本身是否处于有序的受控状态。

3.3.1 模型的架构设计

20 多年来，运用新的软件开发方法和技术并没有满足人们对软件生产率与质量的期望，人们逐渐意识到影响软件发展的主要是管理问题而不是技术问题^[20]。通过多年实践，人们发现软件质量是人、过程和技术的函数，即 $Q=f(M, P, T)$ 其中，Q 表示软件质量；M 表示人；P 表示软件生产过程；T 表示技术。管理是影响软件研发项目全局的因素，技术只影响局部。而提高软件组织的生产和管理能力，必须关注技术、过程和人员这三个相关的因素，它们被称为软件产品的质量三角。所以，总的来看提高软件质量要走技术和管理相结合的道路。从管理的角度看，对于软件开发过程的管理，借鉴制造业中成熟的质量管理方法和经验是大有裨益的。但是由于软件开发过程有其特殊性，软件产品主要是员工脑力劳动的结晶，所有加强企业文化建设，确保各岗位人员发挥最大的效能也是提高软件组织质量管理水平的重要保障。

由几十年来的软件工程实践和理论研究结果我们可以发现，成功实施质量管理、积极开展标准化规范化的过程管理的中小软件组织逐渐变大变强，并使其软件产品和服务稳稳的立足于国际市场，例如中国的东软，印度的 Infosys 公司等；相反，开发过程混乱，没有质量保证意识的中小型软件企业日益衰退，客户投诉率高，满意度低，这些企业就会逐渐失去市场竞争力，甚至破产倒闭。软件组织结构和其文化，开发过程和技术是软件企业质量管理成功的铁三角，理想的情况下它们彼此促进，糟糕的情况下它们彼此制约。对于各类资源具有稀缺性的中小软件企业更需要合理的安排以上三个层次的关系，提高产品的质量。因此，本文针对中小软件企业的特点及其质量管理的现状和质量管理的重要因素建立起三层结构的质量管理模型。该模型以技术管理层次和过程管理层次为基础，重点突出强调组织结构和文化层次在质量管理中的重要作用。该模型结构如图 3-2 所示。

3.3.2 模型的逻辑性

在上述的中小软件企业质量管理模型中，通过组织结构和文化层次保证质量管理的制度化，通过过程管理层次保证质量管理的规范化，通过技术管理层次保证质量管理的标准化，多方面多渠道的开展质量改善活动。三个层次以提高软件组织的质量管理水平为目标，共同构成质量管理的铁三角。模型的三个

层次间相互依赖，并存在着映射关系如图 3-3 所示。

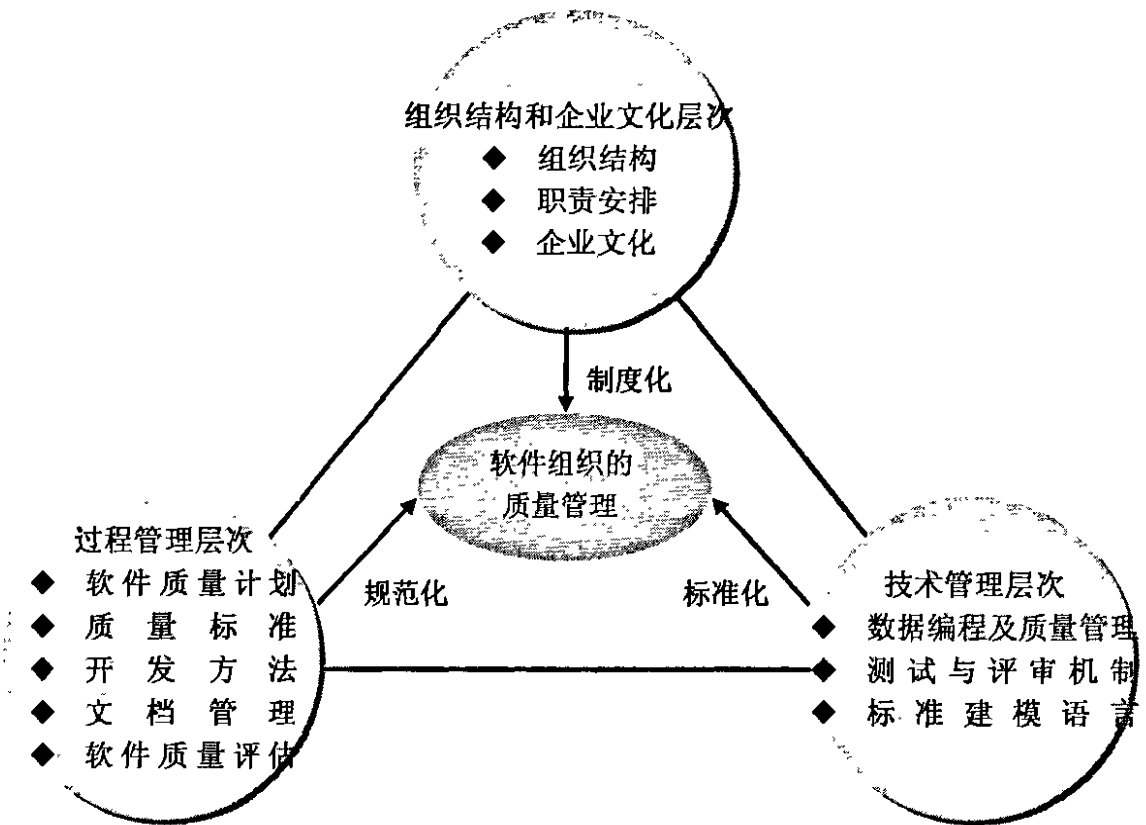


图 3-2 中小软件企业质量管理模型

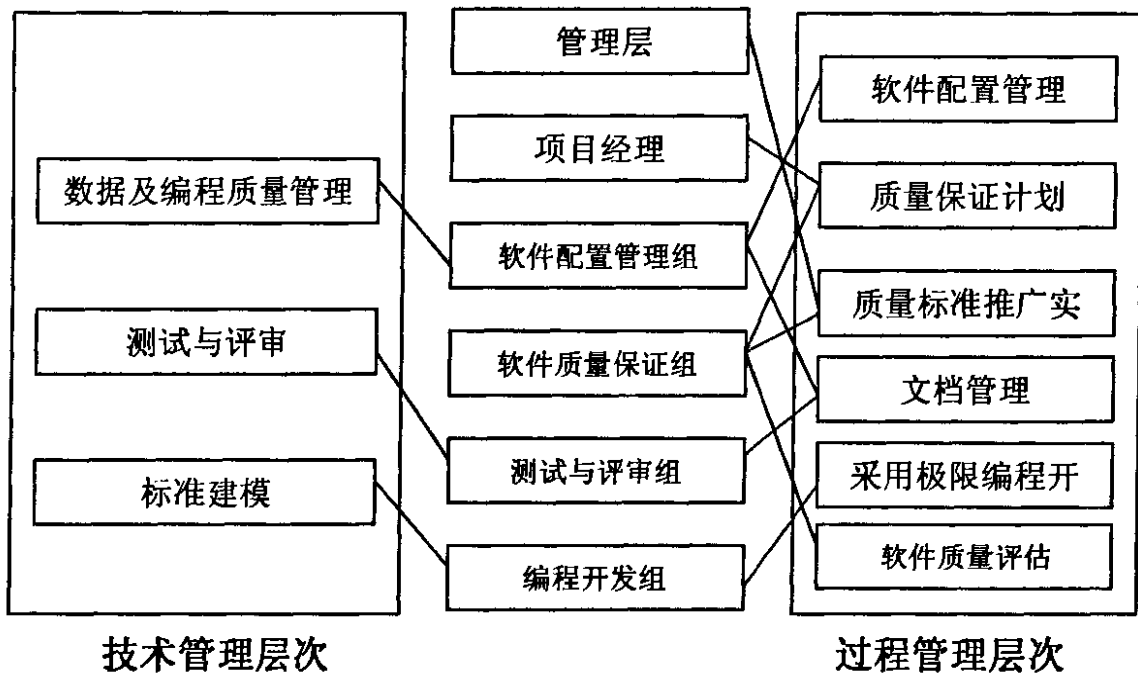


图 3-3 中小软件企业质量管理模型层次映射图

组织结构中的各个岗位都直接参与过程管理活动，并负有相应的职责，同样，技术管理也需要应用于过程管理的流程中去，以技术手段来保证质量。可见，模型是以组织结构和文化为实施条件，以过程管理为联结纽带，以技术管理为基础的系统化质量水平提高方案。其各层次的目的和内在联系在于：

(1)企业组织结构和文化的设计是开展过程管理和技术管理的前提和条件。

只有将质量管理制度化，明确规定每个岗位在质量管理活动中的任务和责任，并且逐步建立起质量生命观，使员工的质量保证行为由有意变为无形，时时处处约束自己，改变以往随意、不规范的工作习惯。

(2)过程管理贯穿于项目的整个生命周期，需要依靠技术手段来实现并关系到软件组织各岗位上的每位成员。

过程管理包括开发和维护软件及其相关的软件产品的一系列活动。从质量保证计划开始，到选择合理的开发方法，并在开发过程中采用剪裁化的质量标准，同时对过程中的文档等资料的编制、审核、保存严格规定，最后需要围绕客户满意度对软件的质量进行评估。过程的质量管理需要领导、质量保证组、软件配置组等人员的配合和协作。

(3)技术管理是提高质量水平的基础活动。

在软件组织承接的软件开发的全过程中通过设计技术、编程技术、测试技术等各种技术手段保证质量，可以较早的发现并解决质量问题，甚至预防并杜绝缺陷的产生，从源头上尽量做到“无缺陷”。

模型的主要逻辑性在于在设计以质量改善为目的的组织结构和文化的条件下，利用质量技术工具，在开发过程的各环节全面的保证最终软件产品的高质量，从而逐步提高整个软件组织的质量管理能力和水平。

3.3.3 模型的合理性

模型建立在对我国中小软件企业特点的分析上，结合典型的质量管理理论、工具、方法，具有针对性和可执行性。其合理性和与其他软件质量标准的优势在于：

(1)针对中小软件企业的特点

中小软件企业人力物力资源有限，软件产品领域和用户范围相对较窄，能够在质量管理上的投入较少，而被广泛采用的 CMM 等标准是针对 500 人以上的大型软件企业，认证步骤繁琐，投资周期长，短期内效果不显著。据调查 CMM 每一级向上一级认证时需要 7-10 万美金，耗时 18-30 个月，这样的花费显然超出了中小软件企业的承受范围。因此本文提出的模型相对简化了如 CMM 等质量标准中设置的某些岗位和活动，如将软件工程组和软件过程组合并，将高级经理、一线软件经理等角色取消，而将其职能融入到项目经理中；将 CMM2 级中的软件合同管理等关键过程暂不推广等等。模型中的组织结构以中小软件企

业的规模为对象，并且各组成员的任务分配相对灵活。

(2)质量改善活动系统而全面

从某种意义上说，CMM 等软件质量标准忽略了对人才、个人技能的要求和考核，特别是忽略了以质量为生命的企业文化的建设。而人为企业之本，尤其软件产品作为人类脑力劳动的结晶，更是需要加强对人力资源的管理。因此，本模型突出了组织结构和企业文化对于软件质量的重要作用，以及技术手段对质量的影响。模型将设置的岗位与过程管理和技术管理中包含的质量改进活动相互映射，保证各类人员在其位，谋其职，并确保将各项活动落实到人，责权分配到岗。

(3)详细阐述了具体的质量活动

一些质量标准强调“做什么”而并未指出“怎么做”，这使很多企业陷入执行的困惑和误区。因此，本模型给出了需要开展的具体活动或详细的指南，保证企业能按部就班的改善质量。从组织结构设置、岗位职责到过程管理和开发技术、测试工具等均给出了详细的说明，保障了企业在实施质量管理时有章可循。

3.4 本章小节

本章在总结我国中小软件企业特点的基础上，进而结合印度中小软件企业通过规范化管理做大做强的案例分析了影响其质量管理的因素，从而提出了针对中小软件企业的质量管理模型架构。并且详细说明了模型的逻辑性和合理性。值得注意和强调的是，中小软件企业若想提高软件质量，改善开发过程，急需的并不只是一套标准，一种新技术或是优秀的管理人才，而是针对其现状和特点而设立的一整套质量管理体系。

第四章 组织结构和企业文化的设计

有效的革命都是从人员结构开始的，也就是对人员的责任与义务进行调整，这是一切活动开展的基础。人、技术、过程是保证软件质量的铁三角。但本文认为，人是其中最重要的因素。一个缺乏对人力资源有效利用的软件组织根本无从开展质量提高活动。这一理论在 1987 年 Tom DeMarco 和 Timothy Lister 出版的软件开发领域的经典之作《人件》中有充分的说明，该著作研究了软件研发人员的心理和行为，并指出在软件开发过程中只有协调人际关系、建立一个好的激励机制才能保证一个高智商团队的高效工作。组织结构是一个组织为行使其职能按某种方式建立的职责、权限及其相互关系，组织结构问题是企业生产力高低的一个决定因素，组织文化是规范和约束企业健康发展的无形动力。所以只有设置以提高质量管理水平为目的的组织结构，并建立以质量为生命的企业文化，将奖惩机制与员工的工作绩效直接联系起来，才能促使员工重视软件质量，保证质量管理体系有效运行。

4.1 组织结构的设计

在中小软件企业中，工作人员较少，组织结构也相对简单。一般由总经理和管理层成员共同决策，协调各个项目组的运行，并负责财务、人事和设备管理等；其余成员以项目划分，同一成员也可以参与多个项目。中小软件企业人力资源有限，这就要求质量体系在人员安排上，把资源用在“刀刃”上，物尽其用，人尽其才。在人员安排上，既要保证企业的每一位成员都能有效地参与到质量提高的过程中来并做出贡献，同时也要保证关键的质量管理岗位的设置，并配备能够胜任的成员。

在软件企业内部，需要合理的安排成员，如图 4-1 所示，第一级为软件企业的管理层；第二级为是企业的各职能部门，包括财务部、人力资源部、软件项目部以及独立于各部门的软件质量保证组；质量保证组是独立于项目之外的组织，直接受第一层领导并处于同等的级别，项目经理作为项目的牵头人和领导，负责安排项目的具体分工和日常活动，以及员工的绩效考核等事宜；第三级为软件配置管理组、系统分析设计组、编程开发组和测试与评审组共四个小组，组员以项目需要组成虚拟团队，但各成员分工不随项目改变。中小软件企业的组织结构如图 4-1 所示。

组织机构设置的合理性在于：

(1)中小软件企业的人力资源有限，因此本结构对 CMM 中设立的组织结构进行了精简，主要由软件质量保证组来完成设立标准、监督控制和执行的任务，

确保以最少的人力资源完成最全面的质量改善活动。

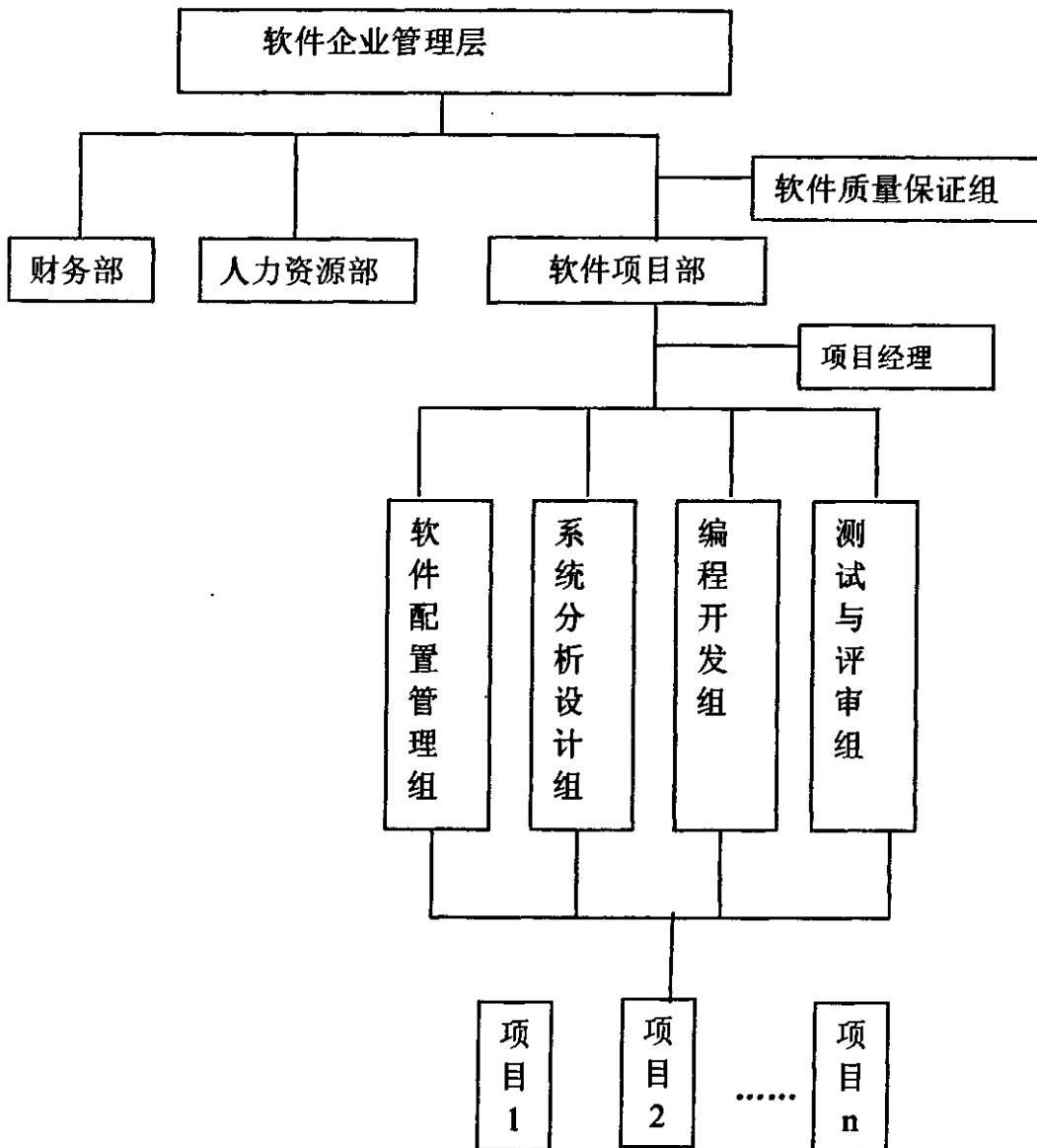


图 4-1 中小软件企业人员组织设置图

(2)该结构考虑到将活动规划到具体的小组，保证软件组织各成员在不同的项目中也从事专业的工作，改变以往多面手的状况。

(3)结构的独立性和完备性。结构中将软件质量保证小组独立出来，其具有直接向领导层汇报和建议质量管理活动的权利，并且结构中的岗位设置是与过程管理和技术管理中具体的任务相映射的。

4.2 职责划分

将组织结构设立之后，就需要对项目企业各人员的具体职责进行详细地划分，保证每项任务责任到人，划分如下：

(1)软件企业管理层

对于质量管理这样的一把手项目来说，软件企业管理层的态度直接决定着质量保证活动能否顺利开展和持续地进行下去。管理层人员需要明确地认识到软件质量的重要性，推广和监督质量管理活动，制定质量保证规范，将质量活动安排到各成员，在企业内部树立起“以质量为生命”的企业文化。

(2)软件质量保证(SQA)组

质量保证组是组织的核心所在，其对软件质量负有直接而重大的责任。因此，必须独立于软件项目组之外，能够直接向企业决策者汇报软件质量管理情况。软件质量保证组负责计划和实施项目质量保证活动，监督软件开发规范、软件过程的执行情况。他们的工作目的是保证软件过程的步骤和标准是否得到遵守。这个小组的成立，是为了让项目成员“用正确的方法做正确的事”，这也是 CMM 所强调的，尽量杜绝“用错误的方法做正确的事”，当然，前提是“用正确的方法一定能够做正确的事”。要求该组成员能够采用科学的工具和方法保证软件质量，例如头脑风暴法、鱼骨图法、黑盒测试、白盒测试等。

SQA 组织并不负责生产高质量的软件产品和制定质量计划，这些都是软件开发人员的工作。SQA 组织的责任是审计项目经理和软件工程组的质量活动并鉴别活动中出现的偏差。

软件质量保证的目标是以独立审查的方式监控软件生产任务的执行，给开发人员和高层管理者提供反映产品质量的信息和数据，辅助软件编程开发组得到高质量的软件产品，其主要工作包括以下三个方面：

- 1)通过监控软件的开发过程和周期性的测试来保证产品的质量；
- 2)保证生产出的软件和软件开发过程符合相应的标准与规程；
- 3)保证软件产品、软件过程中存在的不符合问题得到处理，必要时将问题反映给高层管理者。

因此，这个小组在软件过程改进中会起到很重要的作用，他们的工作是否能正常地开展与改进的成败与否有直接关联。他们的责任是审查软件设计、开发人员的活动，验证他们是否将选定的标准、方法和规程应用到活动中去，因此，SQA 工作的有效执行需要 SQA 人员掌握专业的技术，例如质量控制知识、统计学知识等，并且要求他们具有丰富的经验，为 SQA 人员进行专门的培训，以使他们能够胜任这项工作。

(3)项目经理

组织系统分析小组、系统设计小组人员制订开发计划，组织系统设计小组进行详细设计，根据详细设计进度组织编程人员进行开发活动，进行项目开发的跟踪与监督。项目经理的主要责任是对整个项目的总体业务负责，指导、控制、管理和规范某个软件或软/硬件系统建设，项目经理是最终对客户负责的人。在以项目为单位的机构中，大多数的项目参与人员要向项目负责人汇报，可能

有些还是逐级汇报。项目经理是整个项目的主导者。项目经理的职责更多的是从管理层面来定义，他不需要涉及到具体的项目开发中去，不需要具备优秀的开发编程能力，而是要能够利用科学的项目管理方法，如甘特图、鱼骨图以及头脑风暴法等统一规划、调度、控制整个项目的进度、质量、成本等因素。

在人力资源可行的情况下可设置项目组长。项目组长负责整个项目组内部人力资源分配和考核；负责项目组内所承担项目的计划、进度。项目组长是一个项目的具体负责人，对一个项目的软件活动完全负责，控制一个项目的软件资源，按照软件的约定与项目负责人沟通。在中小软件企业中，这样的角色往往是由开发小组中的某一个成员承担，或者就是项目经理自己担任。此外，由于人力资源的有限性，项目组长常常需要参与到具体的编程工作中去，除了参与系统的需求分析、设计以外，还常常负责系统的整体架构设计、模块集成以及技术难关的攻克等任务。因此，在中小软件企业中，对项目组长的责任和要求非常高。

(4)软件配置管理组

配置管理小组是在保证项目开发完毕时，同时完成内部文档和外部文档的编制。内部文档的及时产生和规范，是保证项目开发各小组能够更好的接口和沟通的重要前提，从另一方面讲，也是保证工程不被某个关键路径所阻塞而延滞的前提。如上所述，配置管理小组还是质量保证小组得以发挥作用的基础。配置管理小组的主要职责包括：完善各个部门发送需要存档和进行版本控制的代码、文档(包括外来文件)和阶段性成果；对代码、文档等进行单向出入的控制；对所有存档的文档进行版本控制；提供文档规范，并传达到开发组中。

软件配置管理组负责策划、协调和实施软件项目的正式配置管理活动，由经理和技术人员组成。由于软件开发的可视性相对较差，所以需要一个小组来维护软件的配置，才能让开发过程有条不紊。这个小组是保证软件过程改进顺利进行的一个重要小组。

(5)系统分析设计组、编程开发组

负责软件的需求调查、需求分析以及概要设计、详细设计和编程工作，根据详细设计进行编码，同时要负责软件后期的安装、维护和升级工作。为了保证软件开发的规范化，要求系统设计组与编程人员具备良好的规范性语言，使程序的可读性强，易理解易修改。

(6)测试与评审组

负责项目组内所承担项目的软件测试。系统测试组是包括负责人和技术人员的一个小组，负责计划和实施对软件的单独系统测试，以确定其软件产品是否满足用户的需求。具体的活动应包括：需求评审、设计评审、代码走查、单元测试、集成测试、系统测试以及验收测试等。

测试与评审组在前期创建项目管理阶段就应该成立，他们专门负责对软件

测试，对软件质量的保证起到十分重要的作用。通常他们是程序问题的第一个发现者，使得软件在交付前的错误发生率降到最低。同样，测试组在软件开发过程的不同阶段所做的工作也有所不同。在软件需求分析和设计阶段，测试组就应该着手编写概要设计计划和详细测试计划等，并且应该有一个详细的测试大纲和准备测试用例。在软件测试阶段，其主要的工作是对软件进行实际的测试，而测试的方法会因不同的系统而有所不同。测试中发现问题，要填写软件问题报告。软件测试的每一个阶段，测试组都应该提交一份软件测试报告，以对软件测试过程中所发现的问题等进行总结分析和汇报。

需要强调的是，在以上罗列的组织结构和职责划分中，在具体的公司环境中，角色和个人并不是一一对应的关系。一个人可以承担多项责任，而一项工作也可由多个人共同承担。例如，软件项目经理可以担任的职务有：项目经理、软件配置管理经理、系统分析师等。而项目组长可以担任的工作有：系统分析师、软件配置管理、软件质量保证等。系统在实际应用中，可以根据项目的任务分解进行合理的调配。

4.3 企业文化

在设置完企业的组织机构并划分职责之后，还需要在企业中营造浓厚的质量管理氛围，提高组织的质量意识。企业中绝大部分成员的观念以及相应的行事方式形成了企业文化。软件的过程改进实际上是企业文化的转变，而企业文化的转变归根结底是人的观念、思想、认识、做事方式的转变，而不仅是个人能力的提高。由于软件本身凝结了开发者的智慧、经验、甚至是感情在其中，包含了太多的人的因素，所以隐藏在产品内部的质量缺陷不可能完全杜绝，单从技术的角度去解决是不够的。

4.3.1 以质量为生命的企业文化的建立

以生产高质量产品为目标的组织文化将为软件项目组成员的质量活动带来无形的约束和保障。本文认为需要在软件组织树立的质量文化应包括以下几点：

(1) 以质量为生命

随着中国加入 WTO，中小软件企业面临的是国内、国外两个市场的激烈竞争。没有高质量的软件产品，没有规范化的质量管理，就很难立足和发展。软件企业中的每位成员，特别是企业的管理层能够视质量为第一要素，以高标准规范自己的工作，把握质量的每一关。在企业内部，自上而下地形成“以质量为生命”的观念，是开展质量管理工作的前提。

(2) 质量管理，从我做起

软件质量管理是体现在每一位软件项目成员的每一项活动中的。只有每一

位成员树立起强烈的质量意识，并提高自身改善质量的能力，将质量保证活动融入日常工作环节中，从而形成良好的工作习惯，才能保证组织在质量管理上的良性发展。

对于中小软件企业的管理层，需要正确认识质量管理带来的额外工作量，配备必要的资源，质量管理最直接的表现就是带来了很多的事务性管理工作，这要求配备适当的工具和人力物力资源。实施体系就是打破旧平衡，建立新平衡，这个过程中体系会相对混乱，对正在运行的项目会产生一定的影响，这些情况，高层管理者都应该心理上有所准备，对质量管理抱有坚定的信心和决心。

对于项目经理，首先需要主动调整心态以适应新的情况；其次必须接受系统的软件过程培训，因为需要从技术为主转变为管理为主；同时项目经理必须积极参与软件过程改进中来。软件企业管理层通过资源和政策驱动，项目经理以实际问题驱动。项目经理要做过程改进的“明白人”，认识到过程改进工作正是为项目服务的，而不单单是被动的服从高层领导人的安排和质量经理的规划等等。

对于其他人员，编程人员需要编写出规范化的程序，测试组人员需要进行模块测试、系统测试并建立相应的文档，系统分析设计组成员需要保证流程图的规范性。

(3)积极开展质量培训活动

软件工程师的工作是逐日的“机械”过程^[21]，没有严格的培训计划以不断提高自己的过程能力，随机性强，没有统筹规划，而事实表明必须有合理的训练才能做规范的工程工作^[22]。积极在企业中开展软件质量管理的培训工作，向员工介绍推广 CMM 等质量标准的知识。通过让顾客满意和公司所有成员及社会受益，从而达到长期成功的一种质量管理模式。定期的质量管理知识培训，不仅为提供企业的质量打好理论基础，对员工个人而言，也是提高其质量能力和知识资本的有效途径，是一项双赢的活动。培训的内容应包括：软件过程理论、软件生命周期、软件开发方法、CMM 等质量标准理论、质量管理办法、软件过程规范等。

如果每个员工都意识到质量对企业成功的重要性，都在寻求提高质量的新途径，都在为达到完美无缺而奋斗，那么这样的软件企业才会充满生命力和可观的发展前景。同时，企业在客户中树立良好的质量形象和声誉，通过优越的设计质量来满足客户需要，也为企业的可持续性发展打下了坚实的基础。

4.3.2 企业文化的建设途径

(1)发端于少数人的倡导——高层领导的作用

在导入质量管理文化的过程中，公司高层领导的支持是成功的关键。只有得到公司高层领导和业务骨干的有力支持，才具备开展质量管理的资源和条件，

其实施才有保障。

高层管理者在来展质量管理的方面，是“参与者和实践者”，而不是“领导者和旁观者”，高层管理者决不只是提供资源、资金，更深层次意义在于“责任”和“参与”。质量管理实施成败的责任人是高层管理者，而不是项目经理和 SQA，如果高层管理者没有亲身实践，就不可能成功，高层领导的主要活动是动员成员最大限度发挥能力。

在实施质量管理体系时，高层管理者要统一全体开发人员的思想，使他们真正理解质量管理的含义，不光要知道做什么，还要领会为什么这样做。

(2)企业文化建设包括三个层次的活动

企业文化的三个层次包括精神层面、制度层面和物质层面，三者是紧密联系的。物质层是企业文化的外在表现和载体，是制度层和精神层的物质基础；制度层则约束和规范着物质层和精神层的建设，没有严格的规章制度，企业文化建设无从谈起；精神层的是形成物质层和制度层的思想基础，也是企业文化核心和灵魂。

1)观念层次上的文化建设

价值观是企业文化的核心，企业文化的实施从改造企业价值观开始。在导入质量管理时，要首先统一全体开发人员的思想，树立整体思想，正确认识产品或项目的质量、功能、进度和成本的平衡关系，避免质量管理仅停留在表面现象。

2)制度和行为层次上的文化建设

从管理的角度来讲，软件企业的管理分为三个发展阶段：用人管理、制度管理、标准管理。我国的中小软件企业目前大多还处于用人管理发展阶段，因此通过制度管理，向更高层次开展标准管理是必要的。因而，在制度和行为层次上，建立完善科学的管理制度，完善员工的行为规范，形成独具特色的管理模式。

质量管理的实质是制度化体系，质量管理也是实施全面制度化的有效途径，没有良好的制度保障，企业文化建设将困难重重。由于人们对待变革的态度势不一样的，从而成为变革的障碍。所以变革的落实需要一个强有力的制度化推力。在推行制度化的文化建设中，企业应根据不同的经营环境，确定企业文化建设的目标。首先，目标必须是确定并且可度量的，如，细化的计划、表现评定、质量指标、日期表的阶段目标等，每一个小组成员都要接受这些标准，作为工作的目标；其次，小组目标必须代表一种严肃的挑战，虽然好的个人特性和成为一个小组的愿望可以培养小组工作的价值，但这些特征是无法自动产生一个有凝聚力的小组的；最后，目标必须被跟踪，进度必须被明显展现出来，这样小组成员就能看到他们是怎样向他们的目标前进的。

3)物质层次上的文化建设

在物质层次上，应构建良好的文化传播网络，成为企业理念的物质载体，具体表现为通过管理工具和文档化的管理，将质量管理思想灌输到每个人的心里，成为其工作的一部分了。一方面是管理工具的应用。管理工具作为思想、方法的载体，它可以将管理有形化、客观化，降低劳动强度，解决手工无法解决的问题，一方面规范软件开发过程、统一人员的工作形式，另一方面会简化软件开发规范化的难度，并且易于为开发人员、管理人员所接受。充分利用管理工具来推行质量管理是一个很好的策略。管理工具有配置管理工具(SI)、项目计划与跟踪工具(MS PROJECT)、功能测试工具以及缺陷管理与跟踪等一些管理工具。通过这些管理工具，可以建立和健全技术管理的科学量化考核、控制体系，变隐性管理为显性量化管理；根据以往项目的经验和数据，估计开发软件所需成本和时间，从而提高软件项目的计划性、可预见性；按照技术管理目标，展开跟踪研究，确保技术管理的实效性。

4.3.3 企业文化变革的方法

企业文化变革的方法主要有：舆论导向法、教育输入法、利用事件法、行为激励法、建立礼仪法、造就楷模法、领导垂范法、活动感染法、创造氛围法等^[23]。这些方法在使用中不是孤立的，我国中小型软件企业应根据企业文化变革的范围和难度，可以以一种方法为主、其他方法为辅，也可以把几种方法结合在一起使用，使之相互渗透、相互补充、综合发挥作用。教育和沟通。这一策略假定，阻力的来源是由于不良的沟通。质量管理推进者与利益相关的各个方面沟通不足是很常见的，因而沟通的内容很重要，但沟通的形式同样重要；支持性措施。组织可以调整其人力资源的考评制度和提升制度，对质量管理提供支持；调整组织架构。这与改变管理者具有同样的效果，可以向组织传递管理当局变革的决心；变革的过程就是协助个人克服阻力的过程，企业可以通过“了解——接受——拥有”的阶梯方法，推动员工由漠不关心和观望转变为知道，继而接受，并全身心投入变革。

4.4 本章小节

本章根据中小软件企业的特点，设计了以提供质量为核心的组织结构和文化。人、技术、过程是保证软件质量的铁三角。但本文认为，人是最重要的因素。一切活动的开展都应以人为本。只有合理的利用有限的人力资源，建立良好的质量管理制度的文化才能规范开发人员的工作流程。该层次包括组织结构的设计、职责划分、以质量为生命的企业文化的建立三个环节，强调从人事管理方面加强对质量的控制。

第五章 过程管理的规范化研究

当前国内软件企业普遍效益低下，其根本原因是软件过程存在缺陷，因此需要通过对软件过程的研究，提出一套适用于国内软件企业过程改进的方法，提高软件研发效率和质量。卡内基梅隆大学的软件工程研究所(SEI)对软件过程的定义是：人们用于开发和维护软件及其相关产品的一组活动、方法、实践和革新^[24]。

任何一个软件企业，只要开发产品，就一定有一个软件过程。SEI 提出的软件能力提高是指对软件开发过程的有效控制和管理，保障最终产品质量。软件过程的不断改进才是提高软件机构软件开发能力和提高软件质量的第一要素。早在 80 年代中期，美国就已进入以过程为中心的软件开发与生产的时代，软件开发平台和过程质量管理模型和平台的研究受到广泛的重视，一些典型和通用的质量标准纷纷出台，其中以美国国防部支持的 CMM/TSP/PSP 的研究最为深入，在软件领域的应用也最为广泛^[25]。我国信息产业部也于 2001 年 4 月发布了《SJ/T 11234-2001 软件过程能力评估模型》来加强对企业开发过程的监督和控制。但是在中小软件企业中全面实施这些针对大型企业而设立的软件质量标准显然不合时宜，也不具备相应的条件。因此需要抓住软件过程中的关键活动予以改进，再逐步推广到过程中的所有环节。

5.1 软件质量保证计划

软件开发过程的质量管理从质量保证计划开始。凡事预则立，不预则废。在软件开发的过程中，更是得到有力的验证。无论何种规模的企业，无论何种规模的软件项目，在软件开发项目启动之前，都有必要制定详细的《软件质量保证计划》，为后期的质量管理和控制做出很好的规划，从而有效地提供质量保证。编制项目的质量计划，首先必须确定项目的范围、中间产品和最终产品，然后明确关于中间产品和最终产品的有关规定、标准，确定可能影响产品质量的技术要点，并找出能够确保高效满足相关规定、标准的过程方法。编制质量计划通常采用流程图、因果分析图等方法对项目进行分析，确定需要监控的关键元素，设置合理的见证点、停工待检点，并制定质量标准。本文建议中小软件企业采用目前各个软件组织较常用的是 ANSI/IEEE STOL 730--1984，983--1986 标准，包括以下内容：

- 1.计划目的
- 2.参考文献
- 3.管理

- 3.1.组织
- 3.2.任务
- 3.3.责任
- 4.文档
 - 4.1.目的
 - 4.2.要求的软件工程文档
 - 4.3.其他文档
- 5.标准和约定
 - 5.1.目的
 - 5.2.约定
- 6.评审和审计
 - 6.1.目的
 - 6.2.评审要求
 - 6.2.1.软件需求的评审
 - 6.2.2.设计评审
 - 6.2.3.软件验证和确认评审
 - 6.2.4.功能评审
 - 6.2.5.物理评审
 - 6.2.6.内部过程评审
 - 6.2.7.管理评审
- 7.测试
- 8.问题报告和改正活动
- 9.工具、技术

就中小软件企业而言，还需要在质量保证计划中添加模块测试、系统测试和软件质量评估等活动来保证计划的完整性。制定软件质量保证计划是过程管理的第一步。在开发活动正式进行之前，需要由项目经理、项目组长及相关人员进行头脑风暴，讨论出详细而周密的计划，为日后开展工作提供可执行的标准和指导方向。在实际工作中也需不断的调整和完善质量保证计划，作为今后开展项目的参考依据。

5.2 采用剪裁化的质量标准

遵循软件质量标准是保证和提高软件质量的必要条件，但是现有的标准如CMM、ISO9000等均是针对500人以上的大型软件企业，认证步骤繁琐，周期长且需要大量的资金投入。因此，在实施质量标准时，必须经过一定的剪裁。根据中国软件行业协会的统计，到2005年6月，我国共有接近11000家软件企业，其中仅有204家通过了CMM 2到5级的认证，而规模在50人以下的软件

企业约占 70%，年淘汰率在 15%左右。并且 CMM 认证费用昂贵、周期长、步骤繁琐，在中国中小软件企业中教条式地实施 CMM 各关键过程域和关键实践未必能对过程真正地改进，并且也不可行。但是各个企业可以参照其来提高开发软件的过程能力和管理流程的科学性，灵活选择、优先实施其中相对于其业务发展和具体项目需要有重要意义的过程域，再逐步完善。利用有限的人力、物力资源，切实地改善软件过程，实现企业和个人利益最大化。

5.2.1 剪裁标准的原则

中小软件企业要根据自身的实际需求和资源配置情况，选用最需要、最能改善软件质量、最能创造价值的质量标准。以 CMM 为例本文认为企业在采用剪裁化的质量标准时，应以以下几点为主要原则：

(1) 按需所取

中小软件企业可以参考已实施 CMM 的成功企业取经，再参照 CMM 模型来改善自身的软件开发过程，选择 18 个关键过程域中最能直接产生效益和效果的首先实施，不要盲目的耗费人力和物力。

(2) 以点带面

中小软件企业可以选择一两个小项目作为样本，尝试性的实施某些关键过程域，若取得成功，即可大范围地推广到更多更大的项目中去，若效果不理想，则可以从中总结经验和教训，为以后开展项目做积累。

(3) 切忌形式主义

国务院于 2000 年 6 月颁发的“18 号文件”第五章第十七条明确提出鼓励软件出口型企业通过 ISO9000 系列质量保证体系认证和 CMM 认证，其认证费用通过中央外贸发展基金适当予以支持。目前各省市、高新区、软件园都有对通过 CMM 的企业给予资金奖励的制度。因此，很多软件企业在实施 CMM 时犯了照搬照抄的形式主义错误，误以为拿到 CMM 认证就可以在国际软件市场中具备竞争力，但是却忽略了自身软件开发过程能力的真正提高和改善。

(4) 全员参与

实施 CMM 这样的软件质量标准是系统工程，它需要从企业主管到一般开发人员的每一位成员的理解和配合。取得高层主管的支持，才能顺利的开展标准实施工作；取得开发人员的认可，才能让质量管理工作卓有成效。

5.2.2 中小软件企业的 CMM2 级剪裁

在《基于粗糙集的 CMM 关键过程域的选择》一文^[26]中，选取对 CMM2 级——可重复级作为剪裁的对象，因为 CMM2 级是软件企业着手进行过程改进和提高的入门阶段，也是我国大部分企业正在参照改进的级别，CMM2 级关键过

程域的实施，可以对成本、进度和功能特性进行追踪，并可以重复以往成功的开发经验，指导今后的实践。该文选取近年来硕士、博士论文中所记录的五家软件组织：天津三友软件公司、上海的P公司、联想软件公司软件事业部、大唐微电子公司远程写卡项目组、深圳保网信息技术公司这五个组织实施 CMM2 级的案例。这五家软件公司规模均在 50 至 100 人左右，是典型的中小软件企业。并且都是分阶段逐步地实施某些关键过程域，并取得良好效果，例如生产率的提高，缺陷率的下降等等。而粗糙集算法可以通过分类和属性约简的方法，客观地分析出海量数据中潜在的规则和重要属性。采用粗糙集来分析软件企业中对于 CMM 关键过程域的优先选择问题，能够客观地找出最重要的活动，使企业在资源有限的情况下优先开展更能提高产品质量或增加利润的活动。分析步骤如下：首先通过调查访问得到其他成功开展 CMM 企业实施的关键过程域；然后汇总归一化处理成标准地信息表；最后通过区分找到最重要属性，即优先实施的关键过程域。通过粗糙集分析，我们发现对于中小软件企业来说，在参照 CMM2 级改善过程时，需求管理、软件项目计划、软件质量保证和软件配置管理是最重要关键过程域，应当优先实施；软件项目跟踪与监控、软件转包合同管理是次重要的活动，可以稍后再开展。

(1)需求管理

需求管理是对获取，组织并记录系统需求的系统化方案，以及一个使客户与项目团队对不断变更的系统需求达成并保持一致的过程。需求管理包括以下几个流程：

1)需求分析

问题分析可以通过了解问题及涉众的最初需要，并提出高层解决方案来实现。它是为找出“隐藏在问题之后的问题”而进行的推理和分析。问题分析期间，将对“什么是面临实际问题”和“谁是涉众”等问题达成一致。而且，您还要从业务角度界定解决方案，以及制约该解决方案的因素。您应该已经对项目进行过商业理由分析，这将便于您更好地预计能从构建中的项目中得到多少投资回报。

2)理解涉众需要

需求来自各个方面，比如来自客户，合作伙伴，最终用户或是某领域的专家。您需要掌握如何准确判断需求应来源于哪方面，如何接近这些来源并从中获取信息。提供这些信息主要出处的个人在本项目中称为涉众。

3)定义系统

定义系统指的是解释涉众需求，并整理为对要构建系统的意义明确的说明。在系统定义的初期要确定以下内容：需求构成，文档格式，语言形式，需求的具体程度(需求量及详细程度)，需求的优先级和预计工作量(不同人在不同的实践中通常对这两项内容的看法大不相同)，技术和管理风险以及最初规模。系统

定义活动还可包括与最关键的涉众请求直接联系的初期原型和设计模型。系统定义的结果是用自然语言和图解方式表达的系统说明。

4)管理项目规模

为使项目高效运作，应仔细根据所有涉众的需求确定优先级，并对项目规模进行管理。要慎重选择需求，以确保每次增加都能缓解项目中的已知风险。要达到目的，您需要和项目的涉众协商每次迭代的范围。通常，这要求具备管理项目各个阶段的期望结果的良好技能。

(2)软件项目计划

制订软件项目计划的目的在于建立并维护软件项目各项活动的计划，软件项目计划其实就是一个用来协调软件项目中其它所有计划，指导项目组对项目进行执行和监控的文件。一个好的软件项目计划可为项目的成功实施打下坚实的基础。

软件项目计划包括对于要进行的工作的发展评估，确定必要的责任，详细说明进行工作的计划。软件计划起始于陈述那些要进行的工作和其他对于软件项目(由 KPA-需求管理的活动建立)进行定义和约束的限制和目标。软件计划过程包括评估软件工作产品和所需资源，制定进度表，鉴定和评估软件风险以及责任磋商。重申这些步骤也许对于制定软件项目计划(即软件开发计划)是必要的。

这一计划为执行和管理软件项目活动以及根据软件项目的资源、限制和能力对软件项目的客户定义责任提供基础。

(3)软件质量保证

目标是提供在软件过程中对于软件项目和软件产品的恰当监督和管理。软件质量保证包括审核软件产品和活动，验证其是否符合适用程序和标准并为软件项目和其他相关经理提供审核结果。软件质量保证部门在软件项目初期协同制定计划、标准和程序。通过这种协同工作，软件质量保证部门确保其符合项目要求并可用于贯穿软件生命周期的审核的进行。软件质量保证部门在整个生命周期审核项目活动和软件工作产品并为调查软件项目是否符合其计划、标准和程序提供可视性管理。

(4)软件配置管理

根据《ISO/IEC 12207(1995)信息技术--软件生存期过程》的定义，配置管理过程是在整个软件生存期中实施管理和技术规程的过程，它标识、定义系统中软件项并制定基线；控制软件项的修改和发行；记录和报告软件项的状态和修改申请；保证软件项的完整性、协调性和正确性；以及控制软件项的储存、装载和交付。

软件开发过程中随着工作的进展会产生许多信息，如：需求分析说明、设计说明、源代码、可执行码、用户手册、测试用例、测试结果和这些内容形成

的相应的技术文档；以及合同、计划、会议记录、报告等管理文档。另一方面，软件开发过程中出现变更是不可避免的。软件配置管理将如此庞大且变动中的信息集合规范化，保证开发活动能够有条不紊地进行。它为软件开发提供了一套管理办法和活动原则，成为贯穿软件开发始终的重要质量保证活动。软件配置管理作为 CMM 二级中的一个关键过程域，越来越多的被质量管理采用。实施软件配置管理对于人员变动较大，开发过程混乱的中小软件企业起着基石的作用。合理地设置各级项目组成员的权限、唯一地标识每个软件项的版本、标识共同构成一个完整产品的特定版本的每一软件项的版本、控制由两个或多个独立工作的人员同时对一给定软件项的更新、控制由两个或多个独立工作的人员同时对一给定软件项的更新、按要求在一个或多个位置对复杂产品的更新进行协调、标识并跟踪所有的措施和更改等方面才能保证开发活动的井然有序。

5.3 采用极限编程的方法进行开发

常见的开发方法一般包括原型法、结构化开发方法、面向对象的开发方法以及非常适用于中小软件企业的极限编程(XP)方法。XP 以过程简单为准则，以人员之间的信息沟通为保证、以提高软件质量为目标，可以有效应对软件开发中的需求变化。他不仅是一种软件开发思想，也是一套完整的解决方案。

2001 年美国成立了 Agile Alliance 发表的《敏捷软件开发宣言》中指明敏捷的价值观是：注重个人及互动胜于过程和工具；注重可用的软件胜于详尽的文档；注重客户协作胜于合同谈判；注重响应变化胜于恪守计划。XP 是众多敏捷方法中最常见的一种，它以简单为准则，以人的沟通为重心，而同样以关注质量为根本，它不只是一种思想，更是提供了很多种方法为获得快速而高质量的软件确定了解决方案。XP 消除了大多数重量型开发过程中的不必要产物。从而为开发人员减轻了负担。

极限编程对于中小软件企业的适用性体现在以下几个方面：(1) XP 适用于小规模的项目组，一般在 2 到 10 人之间，估计工作量少于 24 人/月，多于 10 人/月，而开发周期通常少于一年。(2)在 XP 方法中，开发人员编写几乎所有的产品代码都是成对进行的。所有设计决策都牵涉到至少两个人、至少有两个人熟悉系统的每一部分、几乎不可能出现两个人同时疏忽测试或其它任务、代码总是由至少一人复查。这就保证了软件的质量。开发的过程中改变各对的组合则可以在团队范围内传播知识，可以提高团队整体的素质。图 5-1 展示了 XP 的软件过程。

其中：椭圆形框中表示的是 XP 采用的方法，矩形框表示的是软件过程。XP 的软件开发过程是通过将整个项目划分为多个迭代周期来完成，每次提供尽可能小但又比前一版本有所改进的版本供客户考虑，通过多次修改达到客户要求。并且，极限编程方法中所强调的结对编程、测试过程、多次迭代等都为提

高软件质量提供了开发方法上保证。

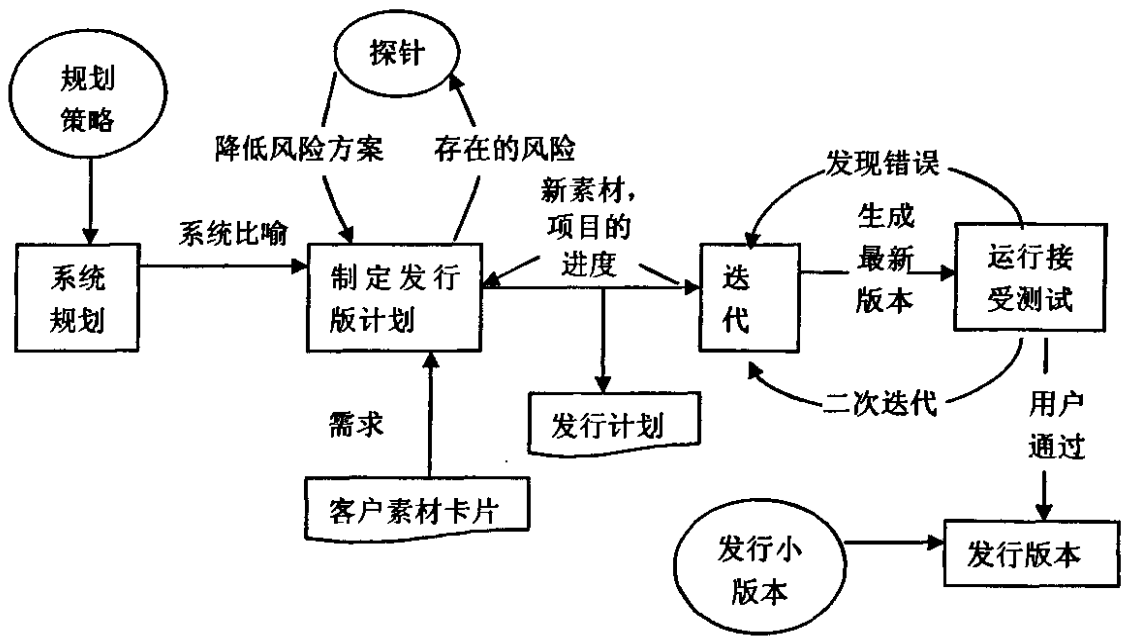


图 5-1 XP 总体过程图

在以上的十余种实践活动中，中小企业可以优先开展的包括：

(1)计划制定(Planning Game)

计划是持续的、循序渐进的。项目计划制定后，应根据进展情况及时调整。在实践中，最好每天早晚各开一个简短会议，早晨明确当前开发周期的目标及每个成员当天的任务，晚上检查当天的任务完成情况并做出必要的调整。

(2)系统隐喻(System Metaphor)

系统的结构由开发者和用户之间的“隐喻”来定义。隐喻是要求开发者将整个系统联系在一起的全局视图，他是系统的未来影像，是他使得所有单独模块的位置和外观变得明确直观。如果模块的外观与整个隐喻不符，那么该模块是错误的^[27]。

(3)结对编程(Pair Programming)

让项目组中的两位成员同时坐在计算机前，一人编程，另一人审查和思考。这将产生一种持续的、动态的交流。对于中小软件企业，这样做有以下好处：结对编程可以保持正在编程人员集中精力和高效率；一个优秀程序员可以培养其他程序员；结对的程序员中的任何一个都可以到用户那里进行现场维护；如果有程序员离开项目组，也不会影响项目的进行。

(4)测试驱动开发(Test Driven Development)

测试比设计更优先考虑。在进行编码之前，先制定测试标准和测试用例。编写单元测试是一个验证行为，更是一个设计行为，也是一种编写文档的行为。

单元测试能及时告诉开发人员系统在某一点是否正确工作，而验收测试则保证系统实现了用户要求的功能。

(5)持续集成(Continuous Integration)

将新的代码不断集成到已经完成的系统中，并进行测试。不能通过测试的单元立即抛弃。每当单元测试完成后都要集成。

(6)小版本发布(Small Release)

尽量以小的版本发布软件，只要增加了满足用户需要的新功能，就可以交付给用户。这样可以及时听取用户的反馈意见，并由此决定下一版本的发布方向。对我国中小软件企业而言，这往往会赢得用户的好感，用户会认为开发人员尊重他们的意见，他们的要求得到了及时的实现。

而关于其他的XP实践活动，例如集体代码所有权，重构等还需根据企业实际情况进一步商榷和研究。

5.4 文档管理

在软件开发的过程中，除了程序以外，也会产生相应的文档资料。认真、规范的编制文档并分门别类的保存是文档管理的主要任务，也往往是编程人员容易忽视的环节。文档的编写和存贮是为了保证开发工作的有迹可训，有章可循。本文认为，企业应在办公系统中建立存放各类文档的模块，方便企业内部人员提交并讨论，为不同的项目，不同组的成员建立起互相交流、互相促进的平台。文档(包括模型)是软件开发过程中的中间成果，这些中间结果关系到软件需求的准确性完整性、设计的合理性，对软件系统的最终结果有决定性作用。文档的管理也是常常被开发人员忽视的重要环节。

文档维护主要是配置管理小组的工作。文档从用途上分主要分为内部文档和外部文档。内部文档包括：项目开发计划；需求分析；体系结构设计说明；详细设计说明；构件索引；构件成分说明；构件接口及调用说明；组件索引；组件接口及调用说明；类索引；类属性及方法说明；测试报告；测试统计报告；源代码；文档分类版本索引；软件安装打包文件；软件开发总结报告。

外部文档主要包括：软件安装手册；软件操作手册；在线帮助；系统性能指标报告；系统操作索引。

在编制文档时有如下要求：

(1)分清读者对象

按不同的类型、不同层次的读者，决定怎样适应他们的需要。例如，管理文档主要是面向管理人员的，用户文档主要是面向用户的，这两类文档不应像开发文档(面向软件开发人员)那样过多地使用软件的专业术语。

(2)精确性

文档的行文应当十分确切，不能出现多义性的描述。同一课题若干文档内容

应该协调一致。

(3)清晰性

文档编写应力求简明，如有可能，配以适当的图表，以增强其清晰性。

(4)完整性

任何一个文档都应当是完整的、独立的，它应自成体系。例如，前言部分应作一般性介绍，正文给出中心内容，必要时还有附录，列出参考资料等。同一课题的几个文档之间可能有些部分相同，这些重复是必要的。例如，同一项目的用户手册和操作册中关于本项目功能、性能、实现环境等方面的描述是没有差别的。特别要避免在文档中出现转引其它文档内容的情况。

(5)灵活性

对于较小的或比较简单的项目，可做适当调整或合并。比如，可将用户手册和操作手册合并成用户操作手册；软件需求说明书可包括对数据的要求，从而去掉数据要求说明书；概要设计说明书与详细设计说明书合并成软件设计说明书等。

(6)可追溯性

前后两个阶段生成的文档，随着开发工作的逐步扩展，具有一定的继承关系。在一个项目各开发阶段之间提供的文档必定存在着可追溯的关系。例如，某一项软件需求，必定在设计说明书，测试计划以至用户手册中有所体现。必要时应能做到跟踪追查。

5.5 软件质量评估

软件质量评估是质量管理中一个非常重要而又容易忽视的环节。

软件质量评估不仅仅是在项目完成后进行，还包括对项目实施过程中的各个关键点的质量评估。项目质量评估看起来属于事后控制，但它的目的不是为了改变那些已经发生的事情，而是试图抓住项目质量合格或不合格的精髓，以使将来的项目质量管理能从中获益。

项目终止阶段，是在决策项目终止后，检查项目文件资料完备，包括项目施工质量验评表、竣工报告等，同时进行项目总结。项目总结是一个把实际运行情况与项目计划不断比较以提炼经验教训的过程。通过项目质量计划和总结，项目过程中的经验和教训将得到完整的记录和升华，成为“组织财富”。

评估要以客户满意度为主要考核指标，并包含软件质量特性的内容，即使用性、可测试性、正确性、维护性、可靠性、移植性、效率、重用性、完整性、互操作性和适应性(灵活性)。

评估的主要活动包括：

(1)计划准备阶段

界定评估范围确定评估目标制定；调查问卷选择组织参与者；制定评估计

划组织和培训评估团队；宣传评估的意义、目标、基本方法和过程；进行问卷调查并分析汇总，进行文档的前期查阅以确定准备内容的完整性。

(2)评估执行阶段

召开动员会明确目标取得组织参与者的支持；与组织参与者进行深度会谈、文档查阅及必要的演示说明；对收集到的信息进行确认归类与汇总；分析总结与归纳通过评估发现的组织过程的强项和弱点；将初步发现的强项和弱点向组织参与者报告说明并征求反馈意见；根据反馈意见做进一步验证评估和必要的修正补充。

(3)评估报告阶段

向组织参与者报告说明评估结果取得共识；向组织管理层报告说明评估结果取得认同以及对进一步质量管理计划的支持。

5.6 本章小节

开发过程是影响软件质量的重要方面。本章就如何在软件开发过程的活动保证软件质量进行了分析，包括制定软件质量保证计划，采用剪裁化的质量标准、选择正确的开发方法、加强对过程中产生的文档的管理以及进行软件质量评估五个方面。

第六章 技术管理的研究

除了前两章介绍的组织结构和企业文化层次以及过程管理层次，技术管理也是很重要的一个环节，是开展质量管理的基础活动。开发人员的技术水平直接关系到软件的优劣。SEI 的研究表明，经过软件质量培训的开发人员编程的错误较少，能减少到 50% 以上，研究还表明，稍稍经过培训便会加速经验的获得。通过技术可以从源头上尽量杜绝缺陷的产生。软件的开发过程中设计到多种技术，建模技术、测试技术、编程技术等等，各种技术的选择和使用都会对软件质量产生重要的影响。本章将阐述技术层次的管理中数据及编程质量管理、建立测试与评审机制、运用标准建模语言三个方面。

6.1 数据及编程质量管理

6.1.1 数据质量管理

多数情况下，软件系统的最终目的是对用户关心的各类数据(信息)完成各种各样静态或者动态的处理或管理任务，为用户创造他们所期望和额外的价值。因此数据质量是用户最为关心的，数据质量也反映了软件系统产品的质量。数据质量是数据抽取、数据转换、数据整合、数据仓库以及管理信息系统开发等项目中质量控制和质量保证必须考虑的主要工作。数据质量管理可分为人工比对、程序比对、统计分析三个层次。

(1)人工比对

为了检查数据的正确性，测试人员打开相关数据库，对转换前和转换后的数据进行直接的比对，发现其不一致性，通知相关人员进行纠正。

(2)程序比对

为了自动化地检查数据的质量，更好地进行测试对比，程序员编写查询比对程序给测试人员使用。测试人员使用此程序对转换前和转换后的数据进行比对，发现其不一致性，通知相关人员进行纠正。

(3)统计分析

为了更加全面地从总体上检查数据的质量，需要通过统计分析的方法，主要通过对新旧数据不同角度、不同视图的统计。为了更加全面地从总体上检查数据的质量，需要通过统计分析的方法，主要通过对新旧数据不同角度、不同视图的统计对数据转换的正确程度进行量化的分析，发现其在某个统计结果的不一致性，通知相关人员进行纠正。

6.1.2 编程质量管理

软件系统是靠“编”出来的，为了确保软件产品的质量，就必须确保软件程序代码的质量。为了提高编程质量，应检查源码的逻辑、属性、对象命名标准、语言代码布局等内容；代码的编译、链接、集成和构建必须得到验证和确认。编程质量管理层次可分为编译检查、编程规范、编程逻辑、编程优化。

(1)编译检查

使用开发工具所带的编译功能或专门程序对软件源码进行检查，分析和寻找源码存在的问题。

(2)编程规范

通过人工源码检查判断源码是否符合企业已经制定的相关编程规范。制定编程规范，在企业内形成一个开发约定和规则，有利于整体风格统一、代码的可读性、可维护性和可扩展性。

(3)编程逻辑

所编写的源码是否考虑周全，无矛盾或遗漏之处。常见问题如：忘记定义变量就使用、变量没有赋值就直接使用、输入输出的数据类型与所用格式说明符不一致、没有注意数据的数值范围造成数组越界或数据溢出、输入时数组的组织方式与要求不符、循环语句可能会造成死循环、条件语句只考虑符合的情况而没有考虑例外的情况、读取文件或数据库中的数据没有考虑例外情况，等等。

(4)编程优化

通过人工或软件检查判断是否可进一步提高源码总体性能和运行可管理性。总体性能如内存管理、数据库组织和内容、非数据库信息、任务并行性、网络多人操作、关键算法、与网络、硬件和其他系统接口对性能的影响等等；运行可管理性如便于控制系统运行、监视系统状态、错误处理；模块间通信的简单性等等。

6.2 建立测试和评审机制

6.2.1 建立有效的测试机制

完善的测试机制，包括丰富的软件测试经验，强大的测试工具，优秀的测试管理水平。尽早测试、连续测试、自动化测试，就是 IBM 在软件自动化测试技术核心的三个最佳成功经验。并在此基础上提供了完整的软件测试流程和一整套的软件自动化测试工具，组建一个测试团队，基于一套完整的软件测试流程，使用一套完整的自动化软件测试工具，完成全方位的软件质量验证。

(1)扩大软件测试队伍和资源投入的问题。

某著名国际软件企业的软件测试人员与软件开发人员的比率达到 3:5 左右，并且在实践过程已经证明了这种人员结构的合理性。但国内公司显然一时很难达到，但更重要的是重视程度和资源投入，在这个基础上壮大软件测试队伍，提高测试人员的素质。

(2) 软件测试应贯穿于整个软件产品生命期。

一方面，软件测试也要经历测试计划、测试用例的设计和实现，以及测试运行一系列的阶段，因此，早在软件需求阶段，甚至更早，软件测试的工作就要开始了。软件产品的错误和缺陷往往发现得太晚，大部分是在用户已经将系统投入使用时才发现，而此时的维护成本比早期发现高数十倍^[28]。在开发之前要了解测试需求，不断进行模型测试；在应用程序完成后还得测试是否能完成应用需要。只有在整个生命周期中坚持测试，才能不断提高应用程序质量。加大软件测试在软件生命周期中的地位是保证软件质量的关键。另一方面，软件测试越早进行越好，因为 BUG 越早发现，BUG 造成的影响和修改的代价就越小。而且，软件测试并不仅仅针对程序，软件的需求、设计等等也要被测试编译检查。

(3) 选择自动化测试工具。

在进行单元测试时，可以选择 IBM 的 Rational Purify Plus 工具，其适合要在短期内快速提升单元测试能力和运行分析能力的团队；在客户端功能测试时，可选择 Mercury 公司的 winrunner；compuware 的 qarun 或是 IBM Rational 的 SQA robot 工具；在实时检测时，可采用 IBM 的 Rational Purify Plus 或是 IBM 的 Rational Test RealTime，这两种检测工具用于实时/嵌入式产品的全面测试和运行时，可以完整的运行时分析工具可以提高应用程序的可靠性和性能。

此外，也要重视第三方的测试力量。第三方的专业测试企业是靠技术与服务来赢得客户信任的，也因此更加注重测试方法与质量。对于中小软件企业来说，从无到有地去建立测试部门，并完善测试体系，需要较大投入，将研发出来的软件产品交给实力强劲的第三方专业测试公司，在提高软件产品的质量问题的同时，也可节约了产品测试成本。

6.2.2 确保评审会议制度的形成

对于质量计划中设置的见证点、停工待检点，软件企业相关人员即需要召开质量评审会议，按照作业程序及时进行测量检查(其中对于停工待检点必须由监理人员签字认可后才能进入下一道工序)，以确定项目成果(或阶段成果)是否符合相关的质量标准。参会的人一般可包括：公司技术专家、用户以及项目组小组成员一起讨论项目计划的可行性，会议通常采用头脑风暴法，各抒己见，会后由指定的记录员形成质量记录，发送给相关人员。应对质量监测的结果应采用相应的统计方法进行分析，如帕累托图法(按发生频率排序的直方图，它显

示了可识别原因的种类和所造成的结果的数量)等。通过统计分析对人员、设备、参考资料、方法、环境等影响项目质量的因素进行监控,确定项目实施过程是否在控制之中,同时进行趋势分析,对一些偏向于不合格的趋势及早进行控制。质量控制阶段应根据验收数据做出验收决定,确定是否进入下一步工序。对于质量监测中发现的不合格,应及时利用“因果分析图”等方法分析原因,并进行适宜的处置,保证不合格得到识别和有效的控制。不合格处置包括返工、返修、降级、让步放行、报废等形式。

质量监测分析时,对于已发现的不合格或潜在不合格,应制定相应的纠正措施或预防措施,以消除不合格或潜在不合格的原因,防止不合格的发生。纠正措施或预防措施制定后,应对质量计划进行相应的调整,保证项目的顺利实施。

6.3 标准建模语言 UML 的使用

一个好的设计基本上决定了产品的最终质量。设计是把需求转换成系统的一个关键步骤,它需要从自然语言描述的需求中寻找出设计的基础单元,构建出整个系统的构架。UML 是国际上比较流行的一种建模语言^[29]。UML 的重要性在于,表示方法的标准化有效地促进了不同背景人们的交流,有效地促进软件设计、开发和测试人员的相互理解。无论分析、设计和开发人员采取何种不同的方法或过程,他们提交的设计产品都是用 UML 来描述的,这有利地促进了相互的理解。

此外,UML 适用于系统开发过程中从需求、规格说明到测试等的不同阶段。UML 适用于系统开发过程中从需求到测试的软件开发生命周期的全过程。包括:

(1)需求分析

UML 的用例图表示用户的需求,活动图对每个用例进行需求说明。

(2)分析

类图描述系统的静态结构,协作图、状态图、顺序图。活动图描述系统的动态特征。

(3)设计

设计阶段的任务是通过综合考虑所有的技术限制,扩展和细化分析阶段的模型,并得到可行的技术解决方案。

(4)实现

构造或实现阶段是对类进行编程的过程。可以选择某种面向对象编程语言作为实现系统的软件环境。

(5)测试

完成系统编码后,需要对系统进行测试,它通常包括:单元测试、集成测

试、系统测试和验收测试。

在需求分析阶段，可以用用例来捕获用户需求。通过用例建模，描述对系统感兴趣的外部角色及其对系统 (用例)的功能要求。分析阶段主要关心问题域中的主要概念 (如抽象、类和对象等)和机制，需要识别这些类以及它们相互间的关系，并用 UML 类图来描述。为实现用例，类之间需要协作，这可以用 UML 动态模型来描述。在分析阶段，只对问题域的对象(现实世界的概念)建模，而不考虑定义软件系统中技术细节的类(如处理用户接口、数据库、通讯和并行性等问题的类)。这些技术细节将在设计阶段引入，因此设计阶段为构造阶段提供更详细的规格说明。

6.4 本章小节

本章对中小软件企业在技术管理层次的三个方面进行了详细的说明。包括数据及编程质量管理、建立测试与评审机制、运用标准建模语言，旨在通过具体的活动和安排为中小软件企业在技术方面提供支持。

第七章 我国中小软件企业质量管理对策研究

对占我国软件行业 70%左右的中小软件企业的质量管理进行分析和理论研究具有现实的意义和价值。提高我国中小软件企业质量管理的水平非一朝一夕之事，也不可能一蹴而就。这是一项系统工程，是动态的发展过程，需要从国家到企业，软件工程人员以及高校等各软件教育机构的共同努力和协作。总结影响中小软件企业质量的关键问题和症结所在并针对其开展对策研究是本章的主要思路。

7.1 国家的政策推动

(1) 国家政策的推动是改善中小软件企业质量管理水平的原动力

例如国务院于 2000 年 6 月颁发的“18 号文件”第五章第十七条，明确提出鼓励软件出口型企业通过 ISO9000 系列质量保证体系认证和 CMM 认证。北京、上海等地还对实施 CMM 认证的软件企业给予一定财政补贴，这些政策很大程度上提高了中小软件企业加强质量管理，参与国际认证的热情。

(2) 制定软件质量评定标准

我国信息产业部参照 CMM 模型于 2001 年 4 月发布了《SJ/T 11234-2001 软件过程能力评估模型》和《SJ/T 11235-2001 软件能力成熟度模型》两个标准。这些为软件企业开展质量管理活动提供了参考依据，保证了企业有章可循、有章可依。如果政府或信息产业部门能出台一些针对不同规模、不同类型的软件企业质量管理细则，则能更好的保证软件行业的健康发展。

(3) 在计算机教育机构设立软件质量管理类课程

随着软件业的发展和需要，国家有必要改革现有的计算机教育体系，以适应软件开发的新形式。在我国软件企业中，目前员工的主力军是经过高等教育的大学毕业生，从业人员素质较高，而且年龄较轻。以中关村科技园区内的软件企业为例，园区内，软件企业人员中，硕士及以上学历为 1.5 万人，大学学历接近 5.9 万人，两者之和占从业人员的比例接近 70%。因此，如果能够在软件开发人员在校期间就树立其质量意识将能更好的改善我国企业质量管理水平。在我国教育机构设置的软件课程中，不乏软件开发工具、系统建模等课程，但是软件质量管理类课程比较匮乏。通过对即将成为软件工程人员的在校学生在质量管理知识方面的教育和引导，使其在走上工作岗位之前就树立起质量为本的意识，并具备较高的质量管理水平，才能从根本上解决我国软件企业质量管理水平薄弱的问题。

7.2 企业的自我管控

(1)领导牵头、全员参与

不仅需要领导重视，还要确实的落实，领导者需要积极参与。说到底软件过程改进首先改的是领导风范、领导人的行为。就像制造工业实行 ERP 一样，重要的不是软件，而是企业的工作方式。所以软件企业的领导层的确实参与是非常重要的。只有在领导的大力推动和督促下，并推动公司中每一位成员改进日常的工作习惯，才能保证软件质量管理工作日益形成规范和制度。

(2)重视软件质量管理

提高软件质量已经成为软件开发组织的共同目标。低质量的软件不仅意味着低的竞争力，更严重的是会给客户造成重大损失，甚至是灾难性的后果。在市场竞争的环境下，企业如果不能在管理上付出努力，就很难在竞争中有所作为。我们的企业管理落后于发达国家，致使为国际公认的我国人才优势和雄厚的智力资源无法充分发挥。具有聪明才智的技术人员在非规范化的管理环境下不能为工程项目的大厦添砖加瓦，管理薄弱反而会使无序的活动成为无效的活动或者导致力量互相抵消。企业先进的管理水平、良好的经营状况是开展质量管理的基础。

(3)重视软件开发的规范性

不少软件开发人员凭自己的经验，习惯了自己熟悉的一套开发方法和步骤，或者习惯了非规范、任意性很强的做法，不愿受规范化管理的约束。一些开发活动事先不作计划；活动过程中不作记录；项目临近结束补写资料，赶制文档。在开发进度由于各种原因延误的情况下，降低测试工作等等。这些不规范的开发行为要靠管理加以纠正。

(4)将质量管理的基本要求体现在开发规范中

目前，不少软件企业并未建立自己的开发规范，使得项目或产品的开发工作无章可循、无法可依。有的项目或产品开发获得成功完全是因为参与的技术人员或管理人员优秀，但他们的成功做法并未得到推广。特别是当他们由于各种原因一旦离开岗位以后，工作情况会发生很大滑坡。只要将质量管理形成准则，形成生产活动执行的标准，才能保证在任何状态都有章可依。

(5)建立良好的人才奖惩机制

首先建立合理的评估机制，不依靠个人感情，而依靠绩效考核。只要他达到了某个程度，就应该奖励，技术人员的薪酬不一定要低于管理人员。通过激励来培养企业内部积极开展质量活动的氛围。同时，也要对不負責任，对质量问题不够重视的个人采取一定惩罚，监督其改正，树立良好的质量观。

(6)建立有效的质量保证手段，使产品质量可预测、可测量

保证质量水平的前提条件之一就是设定合理的质量控制基线，制定规范化

的标准作为质量管理的依据。否则，质量管理活动仍是一片混乱，无从考察。通过质量保证计划和剪裁化的标准等手段使软件开发过程中其质量的可预测的、可测量的。

7.3 软件工程师能力和素质的提高

(1)通过职业再培训提供员工素质

软件工程师是制造软件的一线工人，他们的能力与素质与软件质量有着最直接的关系。我国并不缺少编程人才，但是严重缺乏质量管理人才。因此，企业在实施质量管理体系的过程中，各层次人员需要进行不同的定向培训，即针对保证软件质量的职业再培训。不仅是对开发人员，项目管理人员的培训对质量管理体系的实施，同样起着关键的作用。因为项目管理人员处在承上启下的位置，体系的落实、组内配置管理、组内质量数据的收集都离不开这一层管理人员的配合。通过培训使项目经理等管理人员提高对质量管理工作的重视和能力，才能使之主动配合，增强其质量管理的意识，保障质量改善活动顺利开展。软件工程师也应自觉学习质量管理知识，掌握 CMM 等质量管理标准并应用于开发活动中。

(2)转变质量管理的意识

质量管理作为持续的改进活动，首先需要软件工程师从心理上接受并认可，改变技术本位主义的传统观念。高层经理通过资源和政策来推动，项目经理从解决实际问题方面去推动。软件组织管理层和项目经理是实施软件质量管理的两大推动力。对于质量保证人员来说，要增强服务意识。质量保证小组要密切接触编程人员，采取多种灵活方式，多了解一线人员的意见，保证质量管理活动的团队协作性。质量管理不是个人的事，每个人都应积极参与，每个人都要明确应该尽的职责，凡事都要从我做起。

软件过程改善容易出现急功近利的现象，如果在改善过程中出现了失败很可能放弃所有变更再回到原来的状态，不仅阻碍了企业软件生产能力的提高，还挫伤了软件开发人员的积极性。其实软件过程改善是一循序渐进、长期实施的过程，中小企业员工应将质量管理活动逐步融入日常工作，建立长期的、阶段目标的软件过程改善计划^[30]。

7.4 本章小节

本章分析了提高我国的中小软件企业的质量管理的关键举措，并予以对策研究和发展趋势展望。通过对我国中小软件企业特点的了解和其质量管理特点的发现，我们发现，要想尽快地提高软件质量，减小与发达软件国家的差距，必须形成从国家到企业，再到软件工程师的协同努力。国家政策的推动，企

业自身的重视和软件工程专业人员自身能力和素质的提高是改善我国中小软件企业质量管理水平的关键措施。中小企业的质量管理之路任重道远。

结 论

在经济全球化的今天，高质量、高效率、低成本地开发出软件系统是对每一个软件企业的要求和挑战，中小软件企业面临的形式尤为严峻。工程化、工业化和标准化生产是软件企业生存的必要手段和途径，它已经成为国际软件业通行的趋势。面对市场竞争日益激烈的新形势，机遇和挑战同在，希望与困难并存。以质量求效益、求发展，是我国软件企业生存与发展的必由之路。我国软件企业应该牢固树立“质量第一”的长远观点，将加强质量管理作为当前工作的重中之重，并结合企业的实际情况，建立自己的质量管理体系，这样才能保证企业的可持续发展。

本论文在写作过程中，作者花费了大量时间对我国中小软件企业的质量管理现状进行调查和研究，并进行了详细的分析，找出症结之所在。文章结合我国软件业的实际情况和现代软件质量管理方法和标准等建立了一种适合我国中小型软件企业的软件质量管理模型，为我国中小软件组织走上软件开发规范化道路起到一定帮助作用。该模型包含三个层次：人员组织层次、过程管理层次、技术管理层次。三个层次相辅相成，缺一不可，其共同保障中小软件企业的质量水平。在三个层次中又详细地划分成 11 个方面，全面的构建起质量管控模型。本文的主旨就是强调质量及质量管理对于软件业，特别是我国广大的中小软件企业的重要性，希望引起管理层的重视，能够选用合适的人才，简便的方法做正确的事情，用最少的资源创造最大的效益。

但同时仍有一些问题有待探讨和商榷：

(1)关于质量标准的剪裁问题

针对各个中小软件企业在资源、条件、管理上的差异性，究竟如何剪裁过程标准使之符合企业实际需求还有待探讨。粗糙集的算法虽然可以分析一些成功开展 CMM2 级中小软件企业优先开展的关键过程域，但选取的数据量有限，不能全面的反映剪裁的方法。

(2)模型的适用性

本文的中小企业质量管理模型建立在现状分析和理论研究的基础上，具有很强的针对性和实用性。但限于条件等原因，无法在某中小企业中试点推行，因此，在模型的适用性上有待考察。

以上两点和对于我国中小软件企业质量管理发展趋势等问题的研究将是本文的后续课题，笔者将在今后继续努力，希望能为我国软件业质量管理的研究做出微薄贡献。

参考文献

- [1] 秦娥, 魏红娟. 加强软件质量管理的必要性[J]. 平原大学学报. 2003
- [2] 贲可荣, 马良荔, 刘孟仁. 软件可靠性分析及对策[A]. 第四届全国计算机应用联合学术会议论文集. 电子工业出版社, 1997
- [3] 左新斌, 彭志忠. MIS 质量管理及 CMM 认证[J]. 山东省青年管理干部学院学报, 2004
- [4] 殷新春, 汤克明, 严芬. CMM 实践中的软件开发过程与过程裁剪. 小型微型计算机系统, 2003(2)
- [5] 马良荔, 刘孟仁, 贲可荣. 软件工程能力成熟度模型研究[J]. 计算机应用研究, 1998, 15(6): 8-10
- [6] 曾浩, 贲可荣. 通过提升 CMM 级别改进软件质量[J]. 舰船电子工程, 2001, (2): 35-41
- [7] Carnegie Mellon University Software Engineer Institution. The Capability Maturity Model Guidelines for Improving Software Process [M]. Beijing: Post & Telecom Press, 2002
- [8] OLOTE P. CMM in Practice: Processes for Executing Software Projects at Infosys [M]. Beijing: High Education Press, 2002
- [9] Faizul Huq. Testing in the software development life-cycle: now or later. International Journal of Project Management. 2000, 8
- [10] Technical Report CMM/SEI- 9 3- TR- 25, Key Practice of the Capability Maturity Model SM, [S]. Version1. 1
- [11] Mark C Paulk, Bill Curtis ect. The Capability Model for Software. <http://www.sei.cmu.edu>
- [12] 试析 ISO9001 CMM SPICE 之异同, 李晓桦, 世界标准化于质量管理, 第三期, 2002(3): 33-36
- [13] 梁成才. CMM 二级 KPA 软件配置管理的设计及实施方案[J]. 计算机工程 2003(1): 83-85
- [14] Walls S Hurrphrey. The Team Software Process [R] TECHNICAL REPORT CMU/SEI-2000-TR-023 ESC-TR-2000 023, 2000
- [15] CCID 咨询顾问公司. 2000 年中国软件产品卖掉 230 亿. 中国经营报, 2001 (3)
- [16] 殷志鹤. CMM 访谈实录[J]. 软件世界, 2001 (3)
- [17] 李健, 金茂忠, 中小型企业软件过程改善方法研究[J]. 计算机工程与应用 2001(19): 107-112
- [18] Panka j, Jalote, CMM 实践 Infosys 公司实施软件项目的过程[M]. 高等教育

- 出版社[M], 2003
- [19]S. A. Freos. Motivated humans for reliable software products. *Microprocessors and Microsystems*. 1998, 4
- [20]Watts Humphrey. Three Process Perspectives: Organizations, teams, and People [J]. *Software Engineering*, 2002, Volume14: 39 -72
- [21]Walls S Hurrphrey. The Personal Software Process. <http://www.sei.cmu.edu>
- [22]Walls S Hurrphrey. The Personal Software Process: Overview, Practice and Results. <http://www.sei.cmu.edu>
- [23]谷文广. 国内软件企业实施过程改进的实用性方法探索[J]. *北京工商大学学报 (自然科学版)*, 2004, 22(3): 46-48
- [24]Declan P Kelly, Bill culleton. Process Improvement for Small Organizations [J]. *IEEE computer*, 1999: (10): 41-47
- [25]Mark C Paulk, B Curtis, M B Chrissies, C V Weber, Capability Maturity Model for Software[R]. Version1. 1, CMU/SEI-93-TR-24, Software Engineering Institution, Carnegie Mellon University, February1993
- [26]梁昌勇, 李兴国, 张玉. 基于粗糙集的 CMM 关键过程域的选择[J]. *运筹与管理*, 2006, (8), 134-138
- [27]Kent Beck, Martin Fowler. 规划极限编程[M]. 人民邮电出版社, 2002
- [28]罗圣仪 计算机软件质量保证的方法和实践[M]. 北京科学出版社, 1999
- [29]James Rumbaugh, UML 参考手册[M]. 机械工业出版社, 2000
- [30]李健, 金茂忠. 有效改善软件过程方法研究[M]. 人民邮电出版, 1999