



软件学院推荐教材

软件开发管理的实践

——超越CMM5的企业案例分析

张少仲 李远明 等 编著



清华大学出版社

软件开发管理的实践

——超越 CMM5 的企业案例分析

张少仲 李远明 等 编著

清华大学出版社
北 京

内 容 简 介

本书将 6 σ (六西格玛)与 CMM 的评审策略与软件研发管理实践系统融合,集中介绍了在软件工程中研发的核心内容和策略,主要内容包括软件危机和软件成熟度模型、软件度量、需求管理、项目计划与跟踪、合同管理、质量保证、配置管理、培训管理。将理论与实践应用案例紧密结合,具有很强的实践指导意义。

本书适于软件工程专业学生作为教材和教学参考书使用,也可作为软件企业管理,研发技术人员的参考书。

版权所有,翻印必究。举报电话:010-62782989 13901104297 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

软件开发管理的实践——超越 CMM5 的企业案例分析/张少仲,李远明等编著. —北京:清华大学出版社,2005.1

ISBN 7-302-10132-9

I. 软… II. ①张… ②李… III. 软件开发—项目管理 IV. TP311.52

中国版本图书馆 CIP 数据核字(2004)第 131039 号

出 版 者:清华大学出版社

<http://www.tup.com.cn>

社总机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

责任编辑:丁 岭 许振伍

印 刷 者:清华园胶印厂

装 订 者:三河市李旗庄少明装订厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:12.25 字数:236 千字

版 次:2005 年 1 月第 1 版 2005 年 1 月第 1 次印刷

书 号:ISBN 7-302-10132-9/TP·1056

印 数:1~3000

定 价:22.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

前 言

国内软件业发展到今天，面临着需要转折的局面。一方面，以前的开发方式存在很多弊端，随着一些较大软件项目的提出，这种弊端越来越明显，为此软件企业需要改变软件的开发方式；另一方面，对于如何以一种新的、高质量的方式来开发大的软件项目却不大清楚。虽然国外提出了用于提高软件质量的软件过程成熟度模型(CMM)，但该模型只提出了“做什么”的要求，却没有“如何做”的指南。这是因为每个企业的情况不同，必须根据自身情况决定如何去做。为提高软件产品质量，国外很多著名大公司推崇 6 σ 管理法。但 6 σ 管理法几乎适合于任何行业，因此它的重要意义在于其原理。而两者的原理(以下称“质量原理”)极其相似，都认为质量的提高是企业自身能力提高的过程。

作为一家通过了 CMM5 级评审的软件企业，海辉科技公司在几年的高速发展中一直在思考这样一些问题。比如，在技术上哪些东西需要保留，哪些东西必须摒弃，从国内外业界同行中能吸取什么等，对于国外的经验只能批判地接受，必须走出一条符合自己特点的路，这条路也应符合国内大多数软件企业的实际，具有一定的普遍意义。将“质量原理”付诸实践，把软件开发的质量提高到一个新高度，也为了与业界同行在这方面交换心得，编写了本书，本书也是海辉科技管理丛书之一。事实上，正如 6 σ 管理法和 CMM 中所表述的那样，海辉科技公司对开发质量的追求也是在实践中探索，在探索中前进。同时参照了“质量原理”，不断改进自身的开发方式，提高开发质量。对于一个企业来说，质量意味着生命，追求质量的道路永无止境，要不断努力，力争达到质量的新高度。

本书得到了北京大学软件研究所副所长、国家 863 高技术计划专家组成员、中国计算机学会软件工程专业委员会主任梅宏教授的指点。梅教授在百忙之中抽出时间，对本书提出了许多中肯的建议和意见，在此深表感谢！

本书还得到了大连海辉科技股份有限公司同事们的大力协助，在此也致以深深的谢意！同时郑重声明，本书中来自大连海辉科技股份有限公司的资料均已得到该公司的许可。任何单位或个人如果引用，须标明出处，并得到大连海辉科技股份有限公司的许可，否则将承担法律责任。

作者对清华大学出版社计算机第二事业部的丁岭主任和许振伍编辑为本书的顺利出版所付出的辛勤劳动表示感谢。

本书主要面向有一定软件开发经验的软件工程师、软件项目负责人、软件公司的管理者以及一切有志于软件开发工作的大专院校学生，本书也可作为大专院校的软件工程课程的实践类参考书籍。

由于作者水平有限，加上时间仓促，难免存在错误和疏漏，欢迎读者批评指正。

作 者

目 录

第 1 章 软件危机和软件过程成熟度模型.....	1
1.1 软件的发展经历.....	1
1.2 软件危机.....	2
1.2.1 手工作坊式软件的特点.....	2
1.2.2 软件危机的发生.....	2
1.2.3 软件危机的举例.....	4
1.2.4 对软件的新认识.....	4
1.2.5 软件工程过程.....	4
1.3 软件成熟度模型介绍.....	5
1.3.1 不成熟的与成熟的软件机构的对比.....	5
1.3.2 软件过程成熟度模型.....	6
1.3.3 软件过程的可视性的提高.....	10
1.3.4 提高软件过程成熟度级别的意义.....	11
1.4 6 σ 管理法介绍.....	12
1.4.1 6 σ 管理法的定义.....	12
1.4.2 6 σ 管理法的核心.....	13
1.4.3 6 σ 管理法的优点.....	13
1.4.4 6 σ 管理法的步骤.....	13
1.4.5 实施 6 σ 管理法的优势.....	16
1.4.6 关于 6 σ 管理法组织的特征.....	17
1.5 应用举例.....	17
第 2 章 软件度量.....	22
2.1 直接的度量.....	22
2.2 间接的度量.....	23

2.2.1	功能点的度量	23
2.2.2	复杂度的度量	25
2.2.3	生产率、成本的度量	27
2.2.4	正确性、可维护性、完整性、可使用性和错误排除率的度量	27
2.3	项目度量	28
2.3.1	历史数据基线的建立	28
2.3.2	技术的度量	30
2.4	过程度量	31
2.5	应用举例	32
第 3 章	软件开发管理的概要	35
3.1	软件开发管理的要素	35
3.1.1	管理的对象	35
3.1.2	管理的要素	36
3.2	软件开发管理的概要	37
3.2.1	计划的管理	37
3.2.2	跟踪与报告	38
3.2.3	风险管理	38
3.2.4	合同管理	39
3.2.5	技术上统一控制开发内容和方法	39
3.3	软件开发管理的整体视图	40
3.4	应用举例	42
第 4 章	需求管理	46
4.1	获得用户需求	46
4.1.1	获得用户需求的一般方法	46
4.1.2	用户需求不明时的原型化方法	48
4.2	需求分析文档	49
4.2.1	需求规格说明书项目	49

4.2.2	书写需求规格说明书的注意点	50
4.2.3	初步的用户手册的编写	50
4.2.4	用户需求的分析	50
4.3	需求的评审	51
4.3.1	评审的内容	51
4.3.2	评审的注意点	51
4.4	需求的更改	52
4.5	应用举例	52
4.5.1	项目需求调研表和需求分析报告的评审项	52
4.5.2	用户满意度调查表	53
4.5.3	需求的跟踪能力矩阵	54
第 5 章	项目计划	55
5.1	软件的估算	55
5.1.1	估算的意义	55
5.1.2	估算的风险	56
5.1.3	估算的步骤	56
5.2	项目进度安排	59
5.2.1	准确安排进度的重要性	59
5.2.2	进度安排的方法	59
5.2.3	注意点	59
5.3	软件的风险评估	60
5.3.1	风险的分类	60
5.3.2	风险评估的项目	61
5.3.3	风险的估计方法	62
5.3.4	风险的评估	62
5.3.5	风险的消除	62
5.4	制订项目计划	63
5.4.1	制订项目计划的目的	63

5.4.2	主要针对的事项	63
5.4.3	开发对象和环境的优化	64
5.4.4	对象系统的记述	65
5.4.5	合同和约定	66
5.4.6	制订项目计划对各方的依赖	67
5.4.7	软件生命周期的选择、开发方针的确定	68
5.5	项目计划的文档	69
5.5.1	编写项目计划的文档的注意点	69
5.5.2	项目管理计划的目录(示例)	69
5.5.3	项目管理计划的评审	71
5.5.4	项目管理计划的变更	71
5.6	应用举例	72
5.6.1	项目启动表和小组创建表	72
5.6.2	项目开发计划书	73
5.6.3	进度管理表和项目结束分析报告	83
第 6 章	项目的跟踪	85
6.1	跟踪的目的和对象	85
6.1.1	跟踪的目的	85
6.1.2	跟踪的对象	85
6.1.3	跟踪的基本条件	86
6.1.4	跟踪的基本策略	87
6.2	规模的跟踪	87
6.2.1	规模跟踪的目的与方法	87
6.2.2	规模跟踪的要点	88
6.3	工作量、成本的跟踪	89
6.3.1	工作量、成本跟踪的目的与方法	89
6.3.2	工作量、成本跟踪的要点	90
6.4	关键计算机资源的跟踪	91

6.4.1 关键计算机资源跟踪的目的	91
6.4.2 关键计算机资源跟踪的要点	92
6.5 进度的跟踪	93
6.5.1 进度跟踪的目的	93
6.5.2 进度跟踪的要点	93
6.5.3 进度管理	94
6.5.4 报告	96
6.6 风险的跟踪	96
6.6.1 风险的标识和评价	96
6.6.2 跟踪风险的时间	97
6.6.3 跟踪风险的要点	97
6.7 软件工程技术活动的跟踪	97
6.7.1 跟踪的目的	97
6.7.2 跟踪的要点	98
6.8 报告	99
6.8.1 报告的目的	99
6.8.2 报告的注意点	99
6.8.3 报告的种类	100
6.8.4 报告的内容	101
6.8.5 进度状况报告(小组周报示例)	102
6.9 应用举例	102
6.9.1 跟踪和监控项目会议记录表、问题表和行动跟踪表	102
6.9.2 缺陷预防举例	103
6.9.3 用挣得值分析方法跟踪项目成本举例	106
6.9.4 用关键路径方法跟踪和控制项目进度举例	106
第7章 合同管理	108
7.1 与用户的合同管理	108
7.1.1 合同的准备、签署、变更	108

- 7.1.2 合同的履行 109
- 7.2 分包管理..... 113
 - 7.2.1 合同与订货 114
 - 7.2.2 计划、进度、质量和配置管理 116
 - 7.2.3 交付、验收、支付、评价 120
- 7.3 应用举例..... 122
 - 7.3.1 内部合同书与完成项目统计表 122
 - 7.3.2 客户服务品质协议书 124
- 第 8 章 质量保证 126**
 - 8.1 质量保证计划..... 126
 - 8.1.1 质量保证计划内容 126
 - 8.1.2 模块结束标准 128
 - 8.1.3 测试计划 128
 - 8.1.4 系统完成标准 130
 - 8.1.5 质量保证计划、测试计划、项目结束标准的完成时间和
准备事项举例 131
 - 8.2 质量保证的实施..... 132
 - 8.2.1 走查 132
 - 8.2.2 检查 133
 - 8.2.3 正式评审 134
 - 8.2.4 测试 135
 - 8.3 问题管理..... 136
 - 8.3.1 问题管理的要点 136
 - 8.3.2 问题的种类 137
 - 8.3.3 问题管理的功能 138
 - 8.3.4 问题管理的顺序 138
 - 8.4 应用举例..... 139
 - 8.4.1 测试与评审缺陷表和缺陷原因分类表 139

8.4.2	软件测试计划和测试内容一览表	142
8.4.3	软件质量保证报告和软件质量保证跟踪记录	144
8.4.4	缺陷防范和过程改进计划	146
第 9 章	配置管理	148
9.1	标识软件配置项	148
9.1.1	软件配置项	148
9.1.2	标识软件配置项	149
9.1.3	软件配置项的联系	150
9.2	建立软件基线库	150
9.3	版本控制	150
9.4	变更管理	151
9.4.1	变更管理的目的与对象	151
9.4.2	变更管理的操作顺序	152
9.4.3	管理文档和管理项目	154
9.5	基线状态报告	155
9.6	应用举例	155
9.6.1	配置项分类列表和基线库的访问权限列表	155
9.6.2	配置审计报告和配置审计的会议记录	158
9.6.3	软件配置管理计划	159
9.6.4	变更申请表、变更登记表、变更状态报告表和变更登记一览表	163
第 10 章	培训管理	166
10.1	对开发人员的培训	166
10.1.1	培训计划的确定时间	166
10.1.2	培训计划的制订与实施	166
10.1.3	培训计划的变更	168
10.1.4	培训的文档	168
10.2	面向用户的培训	170

10.2.1	培训计划的目的	170
10.2.2	培训计划的制订时间	170
10.2.3	培训计划的要点	170
10.2.4	培训计划的确定与实施	172
10.3	应用举例	173
10.3.1	培训需求表、培训计划表与培训的反馈	173
10.3.2	培训的索引与培训结果评价表	174
10.3.3	用户培训计划、培训实施指导表和培训确认表	177
附录	参考文献	180

第 1 章

软件危机和软件过程成熟度模型

1.1 软件的发展经历

20 世纪 40 年代中出现了世界上第一台计算机以后，在这几十年中，从全球的观点来看，计算机软件经历了 3 个发展阶段。

- 程序设计阶段：20 世纪五六十年代。
- 程序系统阶段：20 世纪六七十年代。
- 软件工程阶段：20 世纪 70 年代及以后。

从表 1.1 中可以看到 3 个发展时期主要特征的对比。

表 1.1 软件发展的 3 个时期及其特点

时期 特点	程序设计 (20 世纪五六十年代)	程序系统 (20 世纪六七十年代)	软件工程 (20 世纪 70 年代及以后)
软件的范围	程序	程序及说明书	程序、文档、数据
程序设计语言	汇编、机器语言	高级语言	面向对象等软件语言
软件工作范围	编码	设计、编码和测试	软件生存期
软件的需求者	程序设计小组	少数用户	市场用户
开发软件的组织	个人	开发小组	开发小组及大中型软件开发机构
软件规模	小型	中小型	大中小型
决定质量的因素	个人技术水平	小组技术水平	管理水平
开发技术和手段	子程序、程序集	结构化程序设计	数据库、开发工具、开发环境、工程化开发方法、标准和规范、网络及分布式开发、面向对象技术
维护责任人	程序设计者	开发小组	专业维护人员
硬件特征	价格高 存储容量小 工作可靠性差	性能价格比、速度、容量及工作可靠性有明显提高	向超高速、大容量、微型化及网络化方向发展

续表

特点 \ 时期	程序设计 (20 世纪五六十年代)	程序系统 (20 世纪六七十年代)	软件工程 (20 世纪 70 年代及以后)
软件特征	规模小，不受重视	软件技术的发展不能满足需要，出现软件危机	开发技术有进步，但未获突破性进展，价高，未完全摆脱软件危机

1.2 软件危机

1.2.1 手工作坊式软件的特点

从 20 世纪 50 年代到 60 年代，人们曾经把程序设计看做一种任人发挥创造才能的技术领域。当时的程序开发方式称为手工作坊式，具有如下特点：

- 计算机软件开发被看成只是编写程序。
- 程序只要能在计算机上得出正确的结果，写法可以不受任何约束。
- 通篇充满了程序技巧或使用了许多窍门的程序被认为是好程序(尽管这些程序很难被别人看懂)。
- 从理解需求到拿出结果都由一人完成，他人无法介入。
- 程序质量全靠软件人员个人来保证，别人无法维护他的结果。
- 程序不是一种单独的产品，除代码外，不形成文档。
- 没有标准或规范，无法使多人共同完成一项大型软件。

这种开发方式，对于小的程序，仅供极小范围使用(例如只是程序设计者本人或只有几个人)，尚可“孤芳自赏”。对于稍大的程序，并需要较长时间为许多人使用的程序，情况就完全不同了。

1.2.2 软件危机的发生

这种开发方式有三大问题。

- 透明度很差。程序制作的全过程只在开发者个人的脑子里，别人无法进行控制和管理。
- 质量得不到保证。把许多人组织起来共同完成一项软件，必须按照一定的规范开展工作，编制程序只是一小部分工作，还有大量的精细设计、严格检测、文档编制和科学管理工作要完成。没有这些工作，软件的质量无法保证。

- 用户对软件的可靠性没信心。用户在对软件进行验收之前，对软件的研制过程一无所知，因此对交付软件的可靠性缺乏信心，只能在使用过程中对软件逐渐了解。

对于需要可靠性很高的软件，风险极大。

随着计算机硬件技术的进步，要求软件能与之相适应。一些复杂的、大型的软件开发项目被提了出来。然而软件技术的进步却不能满足形势发展的要求。在软件开发中遇到的问题无法解决，致使问题积累起来，形成了日益尖锐的矛盾。从软件工程的角度来看，这些问题归纳起来有：

(1) 软件开发无计划性

- 由于缺乏软件开发的经验，缺乏有关软件开发数据的积累，使工作计划很难制订。
- 只能主观盲目地制订计划，执行起来和实际情况有很大差距，致使常常突破经费预算。
- 对于工作量估计不准确，进度计划无法遵循，工期一拖再拖。
- 为了加快速度而增加人力，如果缺乏统一规范，结果适得其反，更加延误了工期。
- 软件开发的投资者和软件的用户对软件开发工作既不满意，也不信任。

(2) 软件需求获取不充分

- 软件设计所依据的需求，在开发的初期阶段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，使得一些问题不能及时解决而隐藏下来，造成开发后期矛盾的集中暴露。这时问题既难于分析，也难以挽回。

(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，参加的人员各行其事。
- 不重视文字资料工作，使设计和实现过程的资料很不完整。
- 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

- 未能在检测阶段充分做好检测工作，提交用户的软件质量差。
- 在运行中暴露出大量的问题。
- 这样的不可靠软件，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。

1.2.3 软件危机的举例

最突出的实例就是美国 IBM 公司在 1963—1966 年开发的 IBM360 机的操作系统。这一项目花了 5000 人/年的工作量，最多时有 1000 人投入开发工作，写出了近 100 万行源程序。尽管投入了这样的人力和物力，得到的结果却非常糟。据统计，这个操作系统每次发布的新版本都是从前一版本中找出 1000 个程序错误而修正的结果。可以想象，这样的软件质量糟到什么地步。难怪这个项目的负责人 F.D.Brooks 事后总结了他在组织开发过程中的沉痛教训时说：“……正像一只逃亡的野兽落到泥潭中做垂死的挣扎，越挣扎陷得越深。最后无法逃脱灭顶之灾，……程序设计工作正像这样一个泥潭，……一批批程序员被迫在泥潭中拼命挣扎，……谁也没有料到问题竟会陷入这样的困境……”。IBM360 操作系统的历史教训成为软件开发项目的典型事例为人们所记取。

1.2.4 对软件的新认识

许多计算机和软件学者尝试，把其他工程领域中行之有效的工程学知识运用到软件开发工作中来。经过不断实践和总结，最后得出一个结论：按工程化的原则和方法组织软件开发工作是有用的，也是摆脱软件危机的一个主要出路。在这一点上，软件开发与计算机硬件研制，甚至与建高楼没有本质的差别。计算机硬件的研制具有如下特点：

- 设计、生产与检验三足鼎立。
- 对修改、扩展有严格的技术状态管理制度控制。
- 可提供全套的资料。
- 检测需要很多设备、仪表、工具。
- 标准件在设计中占有重要地位。

1.2.5 软件工程过程

软件工程过程是为获得软件产品，在软件工具支持下由软件工程师完成的一系列软件工程活动。每个软件开发机构都可以规定自己的软件工程过程。针对不同类型的软件产品，同一软件开发机构也可能使用多个不同的软件工程过程。但无论是哪种情况，软件工程过程通常包含以下基本过程活动。

- 软件规格说明：规定软件的功能及其运行的限制。