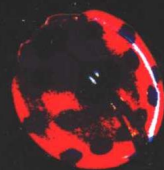


观点工作室 编著



CMM

实践之路

5



机械工业出版社
CHINA MACHINE PRESS



CMM 实践之路

观点工作室 编著

软件能力成熟度模型已成为软件工业事实上的标准，是一条提高我国软件产品质量，增强国际竞争力，促进软件产业快速发展的有效途径。它给软件企业带来的成功已经为许多国内、外著名软件厂商所证明。

本书内容分为三个部分。第一部分主要阐述软件能力成熟度模型的概念、体系结构、实施方法、策略以及先行者的成功经验；第二部分主要阐述软件能力成熟度模型 2 级和 3 级主要关键过程域的详细实施方法和案例，为 2、3 级的实施提供了全方位解决方案；第三部分主要阐述软件能力成熟度模型 2 级内部评估的一套完整方法和案例，为软件过程改进人员作内部评估提供很有价值的信息。

本书可以作为软件过程改进人员随时参考的指导书，也可以作为软件开发人员和其他软件从业人员在软件工程方面的参考书。

图书在版编目 (CIP) 数据

CMM 实践之路/观点工作室编著. —北京: 机械工业出版社, 2003.3
ISBN 7-111-11741-7

I. C... II. 观... III. 软件工程 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2003) 第 012527 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 韩 菲

责任印制: 路 琳

北京蓝海印刷有限公司印刷·新华书店北京发行所发行

2003 年 3 月第 1 版·第 1 次印刷

787mm×1092mm $\frac{1}{16}$ ·16.75 印张·410 千字

0 001—5 000 册

定价: 26.00 元

凡购本图书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话: (010) 68993821、88379646

封面无防伪标均为盗版

前 言

1999年以前,如果有人提起CMM这个新名词时,大多数的IT人士都会摇头说没听说过。的确,IT业发展太快,各种新名词层出不穷,要留下深刻的印象,非得要有不凡的影响力才行。然而很快,CMM就让国人领教了它的不同凡响。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》即18号文件,该文件的第五章第十七条明确提出“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”。表明政府鼓励企业实施CMM认证,各地政府也随之对CMM认证采取了各种激励措施,如珠海开发区规定了通过二级认证的一次性奖励50万元的政策;北京市科委也宣布,通过CMM2认证的软件企业将获得20万元奖励,通过CMM3认证的企业将获得30万元奖励。由此开始,CMM引起了IT业尤其是软件行业的广泛关注,众多软件企业对实施CMM认证充满了激情。仅2001年,北京市就有神州数码、联想软件、用友、东方通、北佳、方正电子、中软网络、亿阳信通等8家软件企业相继通过了CMM2认证,如果加上2001年以前通过CMM2认证的清华鼎鑫及CMM5认证的摩托罗拉这两家机构,目前北京市通过CMM2以上认证的企业或机构已超过10家。不但是纯软件企业,连国内的两大通信设备制造商巨头华为和中兴也在紧锣密鼓地积极推进CMM的实施。媒体当然也不会放过这个机会,不少主流媒体都针对CMM认证进行了集中持续的报道。

究竟什么是CMM?为何CMM会受到如此的关注?它适合中国的软件企业吗?它能给软件企业带来怎样的好处?采取怎样的实施策略才能够获取最大的收益?

CMM(Capacity Maturity Model)是软件过程能力成熟度模型的简称,是卡内基—梅隆大学软件工程研究院为了满足美国联邦政府评估软件供应商能力的要求,于1986年开始研究的模型,并于1993年推出CMM 1.1版,这是目前世界上比较流行和通用的CMM版本。

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,已经成为国际软件过程改进事实上的工业标准。要增强我国软件企业的竞争力,提高国产软件的水平,实施CMM不失为一条最有效的途径。

本书作者是国内一家大型软件企业实施CMM的主要推进人之一,在该企业实施CMM2、3级的过程中积累了丰富的经验,希望能以自己的实践经验和亲身体会给大家更多的启发。本书向读者详细阐述CMM2、3级核心关键过程域实施过程中的种种问题和解决办法以及各种CASE工具的选择使用,给出了许多有实用性的规程和模板,并吸收了许多业界同仁精辟的观点和做法,希望给读者以指导和启发。CMM2级是企业从混沌到有序的一个质的转变,是整个CMM实施的基础,也是企业实施CMM最难着手的地方,它的顺利实施对于CMM在企业的成功应用具有非常重要的意义。而本书恰好是一本极具实战意义值得一看的CMM2级的实施指导手册。

本书的顺利出版,要感谢机械工业出版社的大力支持和帮助。

由于时间仓促,书中难免存在不妥之处,请读者原谅,并提出宝贵意见。

作 者

III

目 录

前言

第 1 章 CMM 概论	1
1.1 CMM 概况	1
1.1.1 CMM 概述	1
1.1.2 国内外实施 CMM 的情况	4
1.1.3 CMM 实施的成功策略	5
1.1.4 CMM 与 UML、XP	8
1.1.5 先行者的经验	9
1.2 CMM 的体系结构	15
1.2.1 能力成熟度模型概述	15
1.2.2 软件过程成熟度的五个等级	17
1.2.3 理解成熟度等级	17
1.3 CMM 的进一步认识	19
1.3.1 成熟度等级的内部结构	19
1.3.2 关键过程域	20
1.3.3 关键实践	23
1.3.4 目标	23
1.3.5 共同特点	24
1.4 实施 CMM 的方法——IDEAL 模型	24
1.4.1 初始阶段	25
1.4.2 诊断阶段	29
1.4.3 建立阶段	31
1.4.4 行动阶段	35
1.4.5 学习阶段	38
第 2 章 需求管理	40
2.1 需求管理概论	40
2.1.1 需求管理在 CMM 中的地位	40
2.1.2 需求管理的定义	40
2.1.3 需求管理和需求工程的关系	41
2.1.4 需求开发的实施	41
2.1.5 需求管理的典型活动和实施步骤	49
2.1.6 需求管理面临的阻力	51
2.1.7 需求管理的度量	52
2.2 需求管理的深刻认识	53
2.2.1 需求管理的规则	53
2.2.2 变更控制	53

2.2.3	SCCB的作用和活动	54
2.2.4	需求的跟踪	55
2.2.5	需求的追踪	57
2.2.6	需求变更波及分析	60
2.2.7	需求管理的评审	63
2.3	典型案例	63
2.3.1	需求管理规程的制订	63
2.3.2	需求管理计划的制订	67
2.3.3	需求管理的工具 Rational RequisitePro 和 DOORS	67
2.3.4	需求管理的整体部署方案	70
第3章	软件项目计划与管理	72
3.1	项目计划和管理概论	72
3.1.1	项目计划——预测则立	72
3.1.2	项目计划在 CMM 中的地位	73
3.1.3	项目计划的主要内容	73
3.1.4	项目计划的典型步骤	74
3.1.5	让计划执行变得可控的项目管理	75
3.2	项目计划的深刻认识	77
3.2.1	确定项目生存周期模型	77
3.2.2	项目估算	79
3.2.3	WBS 工作分解结构	83
3.2.4	制订进度计划	87
3.3	项目管理的深刻认识	91
3.3.1	项目计划的跟踪和监督	91
3.3.2	项目状态报告的产生和作用	96
3.3.3	风险管理	96
3.4	典型案例	102
3.4.1	软件项目计划的制订	102
3.4.2	软件项目跟踪计划的制订	104
3.4.3	软件项目计划的评审	104
3.4.4	怎样让项目计划更好地体现实际	105
3.4.5	工作量数据的收集	106
3.4.6	项目管理工具——MS Project2000 和 ProjectCentral 介绍	106
第4章	软件配置管理	128
4.1	配置管理概论	128
4.1.1	配置管理的定义	128
4.1.2	配置管理在 CMM 中的作用	129
4.1.3	配置管理能解决的问题	129
4.1.4	配置管理的主要活动	129

4.2 配置管理的深刻认识	130
4.2.1 配置管理的典型体系结构	130
4.2.2 配置管理中的角色	131
4.2.3 SCCB的建立	132
4.2.4 配置管理——计划先行	135
4.2.5 变更控制——配置管理的核心	141
4.2.6 典型配置管理工具 RationalClearCase 和 ClearQuest 详解	143
4.2.7 配置审核——让配置管理更有效	146
4.2.8 配置状态报告——配置管理的目的体现	148
4.3 典型案例	150
4.3.1 怎样合理部署配置管理	150
4.3.2 选择合适的配置管理工具	152
4.3.3 配置管理计划的制订	155
4.3.4 配置管理审核报告的制订	157
4.3.5 配置管理最佳实践	158
4.3.6 为什么会这样——解析实施过程中的误区	160
第5章 软件质量保证	163
5.1 软件质量保证概论	163
5.1.1 SQA 工程师的诞生	163
5.1.2 软件质量保证的定义	164
5.1.3 软件质量保证的作用	164
5.1.4 软件质量保证的主要活动	165
5.2 软件质量保证的深刻认识	167
5.2.1 软件质量保证参与的过程审核	167
5.2.2 软件质量保证参与的产品审核	172
5.2.3 软件质量保证参与的其他活动	175
5.2.4 软件质量保证计划的制订	176
5.2.5 软件质量保证的验证和确认	177
5.3 典型案例	177
5.3.1 建立有效的软件质量保证机构	177
5.3.2 SQA 人员如何建立自己的权威	179
5.3.3 软件质量保证规程的制订	179
5.3.4 软件质量保证计划模板	183
5.3.5 SQA 审核单的制订	185
5.3.6 SQA 工作量的度量	185
第6章 同行评审	187
6.1 同行评审概论	187
6.1.1 同行评审的概念	187
6.1.2 同行评审在 CMM 中的作用	188

6.1.3	认清现有的评审手段	189
6.1.4	哪些工作产品需要同行评审	190
6.2	同行评审的深刻认识	191
6.2.1	同行评审的主要方法	191
6.2.2	同行评审的组织管理	205
6.2.3	同行评审的数据收集	210
6.3	典型案例	212
6.3.1	正式与非正式评审的区别	212
6.3.2	选择合适的同行评审方式	212
6.3.3	同行评审计划的制订	213
6.3.4	如何提高同行评审的效率	214
第7章	组织过程焦点	216
7.1	组织过程焦点概论	216
7.1.1	组织过程焦点的定义	216
7.1.2	组织过程焦点的基础	217
7.1.3	组织过程焦点的主要活动	218
7.2	组织过程焦点的深刻认识	219
7.2.1	软件过程的评估和改进	219
7.2.2	合理使用软件过程数据库	220
7.2.3	选择和推广新工具	221
7.3	典型案例	222
7.3.1	软件过程改进和维护计划的策划	222
7.3.2	软件过程数据库的作用	225
7.3.3	组织过程焦点的度量、分析和验证	227
第8章	组织过程定义	229
8.1	组织过程定义概论	229
8.1.1	组织过程定义的定义	229
8.1.2	组织过程定义的基础	229
8.1.3	组织过程定义的主要活动	230
8.2	组织过程定义的深刻认识	233
8.2.1	软件工程过程组的建立	233
8.2.2	适合组织的标准软件过程的制订和维护	234
8.2.3	软件过程财富的建立和维护	238
8.3	典型案例	239
8.3.1	软件工程过程组参与的主要活动	239
8.3.2	软件工程过程组规程	240
8.3.3	标准软件过程的指南和准则	240
8.3.4	软件过程数据库要收集的过程和产品数据	241
第9章	CMM2 级评估(CBA-IP1)实践	248

9.1 CMM2 级各关键过程域的关联关系	248
9.2 CMM2 级过程评估总体方案	250
9.3 CMM2 级各关键过程域评估实施	250
9.4 CMM2 级过程审核问题清单	252
9.5 评估结果总体分析	255
9.6 结束语——实施 CMM 是一条没有终点的路	256
参考文献	257

第1章 CMM 概论

1999 年以前,如果有人提起 CMM 这个新名词时,大多数的 IT 人士都会摇头说没听说过。的确,IT 业发展太快,各种新名词层出不穷,要留下深刻的印象,非得要有不凡的影响力才行。然而很快,CMM 就让国人领教了它的不同凡响。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》即 18 号文件,该文件的第五章第十七条明确提出“鼓励软件出口型组织通过 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。”表明政府鼓励组织实施 CMM 认证,各地政府也随之对 CMM 认证采取了各种激励措施,如珠海开发区规定了通过二级认证的一次性奖励 50 万元的政策;北京市科委也宣布,通过 CMM2 认证的软件组织将获得 20 万元奖励,通过 CMM3 认证的组织将获得 30 万元奖励。由此开始,CMM 引起了 IT 业尤其是软件行业的广泛关注,众多软件组织对实施 CMM 认证充满了激情。仅 2001 年,北京市就有神州数码、联想软件、用友、东方通、北佳、方正电子、中软网络、亿阳信通等 8 家软件组织相继通过了 CMM2 认证,如果加上 2001 年以前通过 CMM2 认证的清华鼎鑫及 CMM5 认证的摩托罗拉这两家组织,目前北京市通过 CMM2 以上认证的组织已超过 10 家。不但是纯软件组织,连国内的两大通信设备制造商巨头华为和中兴也在紧锣密鼓地积极推进 CMM 的实施。媒体当然也不会放过这个机会,不少主流媒体都针对 CMM 认证进行了集中持续报道。

究竟什么是 CMM? 为何 CMM 会受到如此的关注? 它适合中国的软件组织吗,它能给软件组织带来怎样的好处? 采取怎样的实施策略能够获取最大的收益? 伴随着这种种疑问,我们开始了探索 CMM 的实施之路。

1.1 CMM 概况

1.1.1 CMM 概述

CMM(Capability Maturity Model for Software)中文译为“软件过程能力成熟度模型”。最早的前身是 1986 年由美国卡内基—梅隆大学的软件工程研究所(Software Engineering Institute, SEI)受美国国防部委托研究制订的一种评估其软件承包商能力的办法。1987 年 9 月,SEI 发表了过程成熟度框架的简要描述和成熟度提问单。在对软件过程成熟度框架和成熟度提问单的初始版本经过四年的提炼后,SEI 将软件过程成熟度框架进化为能给软件组织在实施过程改进方面提供有效指导的软件能力成熟度模型(CMM)。

SEI 将 CMM 定义为:对于软件组织在定义、实现、度量、控制和改善其软件过程中各个发展阶段的描述。这个模型便于确定软件组织的现有过程的能力和查找软件质量及过程改进方面最关键的问题,从而为选择过程改进战略提供指南。

在 1991 年和 1992 年期间,推出了 CMM 的最早版本 1.0。1992 年 4 月在举办了 CMM 1.0 版本的研讨会后,推出了目前在全世界推广实施的 CMM 1.1 版本,也称其为 SW-CMM

1.1 版,主要用于软件开发过程和软件开发能力的评估和改进。

随着 CMM 的不断发展,它的家族也越来越庞大,先后出现多种类型的 CMM,如:

- SW-CMM:软件 CMM。
- SE-CMM:系统工程 CMM。
- SSE-CMM:安全工程 CMM。
- SA-CMM:采购 CMM。
- P-CMM:人员管理 CMM。
- IPD-CMM:集成化产品开发 CMM。

CMM 的内容开始包罗万象。为了方便使用和推广,SEI 推出了集成的 CMM 版本——CMMI,并宣布自 CMMI 发布起,将在三年内完成 CMM 到 CMMI 的过渡。2000 年 8 月,CMMI V1.0 发布。2002 年 1 月,CMMI V1.1 发布。2003 年秋,CMMI 将全面取代 CMM。但是由于目前在 CMMI 实施方面尚缺乏经验,建议软件组织先实施 CMM 到 3 级,再转换成 CMMI。因此本书中提到的 CMM 皆指 SW-CMM 1.1,即软件 CMM。

1. CMMI 与 CMM 的总体区别

从总体来说,CMMI 框架看起来给人的感觉比 CMM 更清晰,可操作性更强。它实际上是将每个 PA(Practice Area)要做的工作分成了两大部分,一块是每个 PA 都需要做的工作,即所谓的一般目标(Generic Goals);另一块是不同 PA 特定要做的工作,即所谓特定目标(Specific Goals)。一般目标是通过共同实践(Generic Practices)来完成;特定目标通过特定实践(Specific Practices)来完成。

在 CMMI 中,一般目标包含四个公共属性(Common Feature),即执行约定(Commitment to Perform)、执行能力(Ability to Perform)、执行指导(Directing Implementation)和验证执行(Verifying Implementation)。由于这四个属性针对每个 PA 都是共同要执行的方面,所以称之为公共属性,一般目标就是按这四个方面通过一般实践来得以完成的。

在 CMMI 中,每一个特定目标由一系列特定实践组合完成,这些特定实践包括了 CMM 活动中大部分内容(其中去掉的部分主要移到了 CMMI 的一般目标中,由一般实践来完成),同时又根据不同 PA 的情况,比 CMM 活动增加了一些实践内容。

CMMI 与 CMM 比较,公共属性(Common Feature)的含义和所包含的内容都发生了很大的变化,从上面可以看出 CMMI 中公共属性包含四个属性,并且只是每个 PA 要执行的一部分。而 CMM 中公共属性却包含了执行约定(Commitment to Perform)、执行能力(Ability to Perform)、执行活动(Activies Performed)、度量和分析(Measurment and Analyst)及验证执行(Verifying Implement)共五个属性。同时 CMMI 中的属性比 CMM 增加了一些新要求。表 1-1 给出了两者的区别。

表 1-1 CMM 与 CMMI 的区别

CMMI			CMM
Generic Goals (一般目标)		Specific Goals (特定目标)	Goal 1(目标 1)、Goal 2(目标 2)……
GG 1 (一般目标 1)	GG2 (一般目标 2)	SG1(特定目标 1)、SG2……	

(续)

CMMI		CMM
GP1(共同实践1)、GP2、GP3……	SP1(特定实践1)、SP2、SP3……	Co(执行约定)、Ab(执行能力)、Ac(执行活动)、Me(度量和分析)、Ve(验证执行)

CMMI 属性比 CMM 增加的新要求:

1. Ability to Perform(执行能力)
 - CMMI 在该属性中要求建立维护一个过程执行计划。
2. Directing Implementation(执行指导)
 - 要求过程工作产品要进行配置管理。
 - 要求对有关的干系人进行鉴别和计划。
 - 依据计划监督并控制过程执行,并对偏移计划行为予以纠正。
3. Verifying Implementation(验证执行)
 - 对过程活动的执行情况进行客观的评价。

2. CMM 今后的发展方向

本节中提到的 CMM 今后的发展方向并不是指 CMM 本身的发展前景,而是要重点探讨 CMM 在国内的应用方向。软件生产不同于汽车、纺织等制造行业的生产,它完全是由人进行创造的,外界无法观察,也很难进行监控,同时软件也不像制造业的产品那样可以方便地进行检测评估,这就给软件的管理和控制造成了很大的麻烦。国内的软件组织正是由于缺乏一套完善的管理办法,致使经常出现产品不能按时发布、质量难以保证、项目组中个人作用过于放大等问题。因此,中国的软件业要想真正成为产业,增强国际竞争力,就必须改进整个软件生产管理的过程。而对过程的控制和管理正是 CMM 的强项,它提出的过程监控以及文档先行,让组织软件生产的每个步骤都变得可控和可预见,让组织真正有效地控制软件生产各方面的问题,使软件产品的质量得到真正保证。所以每一个软件公司,只要是真正想有所作为、想持续发展、让组织具备竞争力,就应该学习 CMM 所倡导的工作方式,在软件生产过程中贯彻 CMM 的思想。

显然,很多国内的软件组织也已经认识到了这一点。但是,不同的组织对 CMM 的导入有着不同的策略和目的。从表面上来看,CMM 只是一种资质认证,它可以证明一个软件组织对整个软件开发过程的控制能力。得到了这份认证,意味着良好的组织形象和更多的订单。有些组织正是出于这种目的而进入软件业“应试教育”的误区,以通过认证为导向,生搬硬套,期望拿到“毕业证书”后可以大肆宣传自己的“学历”和“水平”,但折腾一番后发现组织的真正水平并没有提高,依旧不能向客户提供高质量的产品。因此,CMM 的应用方向决不是以拿到认证证书为终极目标,而是应该把 CMM 当作镜子对照自己,作为尺子来衡量自己,找出差距,找出问题,采取措施切切实实提高软件组织的过程能力和软件过程的成熟度,这才是 CMM 应用的真正意义所在。组织借助 CMM 锻炼内功、增强组织的核心竞争力,最终会体现在自己所提供产品的优势上,客户的满意度和美誉度是一个组织实力的最好佐证。相反,如果把精力花如何尽快地通过评估、尽快地“获证”,又会有什么真正的实效呢?

3. CMM 不能做什么

CMM 为软件过程改进提供了一个有秩序、有纪律的框架,告诉用户软件过程改进要做什

么。CMM 的主要作用有如下 3 个方面：

- 用于软件过程的改进(Software Process Improvement, SPI)。帮助软件组织对其软件过程的改进进行计划、制订以及实施。
- 用于软件过程评估(Software Process Assessment, SPA)。在评估中,经过培训的专业人员详细了解一个组织软件过程的状况,找出该组织所面对的与软件过程有关的、最迫切的所有问题,以及取得组织领导层对软件过程改进的支持。
- 软件能力评估(Software Capability Evaluation, SCE)。在能力评鉴中,经过培训的专业人员鉴定软件承包者的能力资格,或者是监理正用于软件生产的软件过程的状况。

很明显 CMM 并不是无所不包的,它并不阐述所有的对保障项目成功来说是重要的问题。例如, CMM 并未谈及用户应该如何去做,更没有提到特定应用领域内的内容:如当前流行的软件建模方法、软件开发方法、各种 CASE 工具或者如何选择、雇佣、激励有能力的开发人员。虽然这些问题对项目的成功至关重要,所以怎样解决这些问题还需根据组织的结构和产品的特性而定。

CMM 给用户提出了明确的目标和方向,用户需要不断地探索,来得到 CMM 有效实施的最佳实践。

1.1.2 国内外实施 CMM 的情况

据 SEI 公布的数据,2001 年 10 月底,全球共有 73 个 CMM 4 级组织,66 个 CMM 5 级组织;而 2000 年 12 月全球通过 CMM 评估的组织,通过 2 级的为 449 家,通过 3 级的为 236 家,通过 4 级的为 60 家,通过 5 级的为 39 家。这 139 家组织中,美国占 59 家,印度占 72 家,其他国家占 8 家。在全球的分布如表 1-2 所示。

表 1-2 国外 CMM 评估情况

国家名称	CMM4	CMM5
中国	0	2
澳大利亚	2	0
法国	1	0
印度	29	43
以色列	1	0
俄国	0	1
新加坡	1	0

根据国内的最新统计,截止 2002 年 8 月,CMM 的评估情况如表 1-3 所示。

表 1-3 国内 CMM 评估情况

公司名称	通过级别	通过时间
摩托罗拉	5 级	2000 年 9 月
化为印度所	4 级	2001 年 12 月
东大阿尔派	3 级	2001 年 6 月
托普软件	3 级	2001 年 12 月
联想软件事业部	3 级	2002 年 1 月
鼎新公司	2 级	2002 年 1 月
用友软件	2 级	2001 年 6 月

(续)

公司名称	通过级别	通过时间
东方通科技	2 级	2001 年 12 月
浪潮通软	2 级	2001 年 11 月
新太科技	2 级	2001 年 12 月
博通公司	2 级	2001 年 4 月
神州数码	2 级	2002 年 4 月

1.1.3 CMM 实施的成功策略

以下是从实施 CMM 过程中所积累的大量经验教训中概括出来的 6 条策略。

1. 自底向上和自顶向下有机结合

在进行软件过程改善的时候,通常有两种做法,称之为自顶向下与自底向上。在自顶向下的做法中,组织成立一个推进小组,一般称为 SEPG(软件工程过程组),他们是组织里“过程规范”制订的组织者。SEPG 组织一些开发人员成立各种任务小组,由这些任务小组根据进行过程改善参照的标准编写各种各样的组织的标准与规范,经过一系列的评审、培训,然后让开发人员去执行。在执行过程中最常见的阻力是来自于开发人员,他们往往会抱怨制订的组织开发规范不符合组织的实际情况,标准太高,无法达到。这种做法,费时费力不讨好,大家的意见都比较大,标准定得比较完美,而且在评审时还要大家表面上都要认可。制订标准的人花费了很大的精力,对标准的评审浪费了大家很多的时间,执行时还难以贯彻下去。因此建议将自顶向下和自底向上方法有机结合,即由 SEPG 深入到开发人员、项目经理中去,充分调研,从开发人员、项目管理人员中吸取可执行性强的改进措施。在这种办法中,不需要管理人员花费太多的精力进行标准的制订和改进的推动,这些工作都是由开发人员自己去做的,管理人员仅仅是起到了监督、整理、归纳的作用,按照这个过程逐步完善形成标准与规范。

2. 未雨绸缪,设计良好的过程架构体系

实施改进之前,组织应对自身当前所有的软件能力水平及过程状态有尽可能的、客观的、详尽的了解。对自己的现状有了充分的了解后,就要设计一个适合组织实际情况的过程架构体系,为组织过程改进提供宏观的指导,指明前进的方向,如图 1-1 所示。当然,这是一个比较粗的过程,它的组成部分需要进一步的细化。

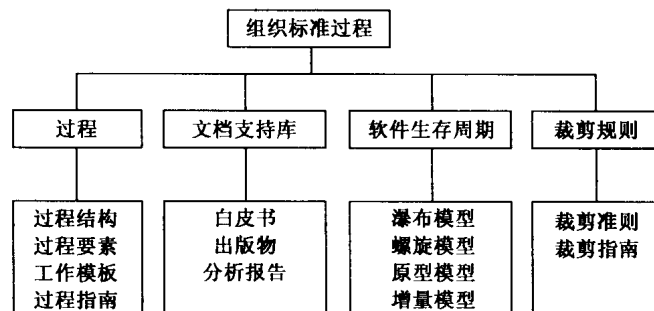


图 1-1 过程架构

3. 循序渐进,由易到难,由粗到细,由松到严

CMM 的一个核心思想是分级改进。在 CMM 模型中将软件组织的过程能力分成了 5 级,有很多组织很可能违背了分级改进的思想,搞了一场革命,期望短时间内提高管理水平,那显然是不现实的,用户需要的是改良而不是革命。分级改进实际上就是要循序渐进,能一步做到 2 级吗?不可能的!对于 2 级的每个 KPA,可能先实现了每个 KPA 的一部分活动,稳定了,再实施另外一部分活动,如果现在 1 级,想一下子将 2 级的所有 KPA 的所有活动都满足是不现实的。在实施 CMM 的过程中一定要根据组织的实际情况量力而行,千万不要期望值太高,要一步一步来。先定出最基本的改进方案,然后逐步提高,要把握分级改进的思想。

要做到循序渐进,首先要对组织现状有一个明确清醒的认识,在分析现状时,下面的四个问题是必须要解答的:

- 当前我们存在哪些问题(当然,问题可能很多)?
- 哪些问题是我們迫切需要解决的?
- 哪些问题是我們目前能够解决的?
- 哪些问题是我們当前无法解决,需要打好基础后才可以解决的?

接下来要对照标准,提出解决方案。按照“力所能及,有所提高”的原则对问题排出优先级。

对 SPP、SPTO 这 2 个 KPA 来说,用户可以采取 5 次循环达到 CMM2 级的要求:

- 第一次循环:从无到有,使项目组成员熟悉做计划的过程,熟悉项目计划跟踪的重要性。

第一步:要求每个项目组都要用 Project 2000 做项目计划,该项目计划要满足一定的条件,如:任务分解的颗粒度不能太大;任务负载要均衡;任务尽可能并行等等。第二步:对每个项目组按计划进度进行跟踪,在计划执行过程中及时发现问题,解决问题。第三步:总结本次循环执行过程中存在的问题,如:项目计划中任务识别不全;计划的任务工作量估算不准;在项目进行过程中,发现问题后采取措施不及时等等。

- 第二次循环:增加完整的生命周期模型定义。

生命周期模型是项目管理的主线,定义一个好的生命周期是推行 CMM2 级的一个最关键的基础工作。

第一步:要求每个项目组首先要定义出自己的生命周期模型,做出项目计划模板。第二步:要求每个项目组按照项目计划模板制定项目计划。第三步:进行项目计划跟踪,用工具如 ProjectCentral 等进行跟踪。

- 第三次循环:增加规模、工作量等的度量。

第一步:要对项目的规模、工作量进行正式的估计,采用 Wideband Delphi 方法或功能点法等。第二步:按计划模板做项目计划。第三步:进行项目计划跟踪,提交项目状态周报告和月报告。

- 第四次循环:增加风险分析。

项目的风险管理对于国内的软件公司而言,一般都是一个难点,可能风险很多,风险可以预见,但是没有很好的规避措施。所以建议将风险管理放在较后的循环中来改进。

- 第五次循环:体系化,制度化。

开始时要求不要太严,对文档的要求要粗一些,要把握住实质而不是皮毛。

4. 教育与培训并重

任何一种标准的推广总会将培训放在一个很高的位置上,培训的关键目的不仅仅是让开

发人员掌握技能,更要改变他们的思想。从这个角度上讲,培训应当作教育来做。因此,首先要做的工作是教育。要让各个方面的人都能够接受这些先进的思想,以便于主观上认可、客观上配合。进行教育有各种方式,需要配合起来使用:

- 观摩学习:看看别的做得好的组织是如何做的。和别的组织的管理人员多沟通交流。尤其是对组织的高层管理人员交流这点很关键。
- 请外面的专家来讲:俗语讲得好“外来的和尚好念经”,这句话屡试不爽。组织内的人说多遍可能效果甚微,外面的专家讲一句,一下就点透了,接受了。
- 自我反省:开发人员自我分析找出自己的不足;并提出改进意见,大家互相沟通交流这些经验教训,自己教育自己。
- 补上“教”:很多软件工程师、项目经理的经验并不是很丰富,也可能比较固执,不撞南墙不回头。没关系,允许他犯错误,但是要有人帮他记录他的错误,提醒他认识自己的不足,加深他的印象。
- 培训:培训可以分成多种:技术型培训与管理型培训。这是最基本的工作,思想工作做通了,培训起来效果就会比较好。

SPI 不能仅仅定义为开发部门的工作,需要整个组织的所有人员的参与和重视。因此教育与培训的对象比较多,不要有遗漏。如:

- 高层管理人员:为什么要进行 SPI? 在过程中会出现什么问题? 对公司有哪些正面与负面的影响? 需要领导配合做哪些工作? 要认识到工作的艰巨性。
- 开发管理人员:技术与管理知识。
- 开发人员:技术与管理知识。
- 市场人员:要认识到 SPI 的重要性,对市场的影响,在此过程中如何配合?
- 客户:对于一个项目而言,需要提出合理的进度、质量与投入的要求,并把握好需求的范围。

5. “因材施教,各个击破”

在一个组织内可能有多个开发项目组或者开发部门,不同的组与部门原来就有不同的管理水平。在推行 CMM 时,不要一刀切,不要希望每个队伍同时达到 CMM2 级或更高的级别,应该区别对待。比如说做产品的部门,经常进行大大小小的各种各样的升级,产品的版本比较多,他们对版本管理的工作认识得很深刻,在工作中积累了一套行之有效的版本管理的方法,版本管理比较正规。对于这样的部门可以实施配置管理 KPA 的要求,进一步提高管理水平。而对于其他做系统集成的部门这方面的工作可能就差一些,没有很好的版本管理的基础。因此,如果一刀切,要求大家都在 3 个月内达到 CMM2 级的要求,这个目标对系统集成的部门讲,就定得太高了。当然,通过发动一场运动来实施 CMM 是可以尝试的,但往往不能得到本质的改变。

所以在进行改进时应针对不同的项目组、不同的部门定出不同的改进策略和推进计划。

6. 充分利用 CASE 工具

CASE 工具可以作为思想、方法的载体,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,易于为开发人员、管理人员所接受。充分利用 CASE 工具来推行 SPI 是一个很好的策略。

例如可以引进 Clear&Case 配置管理工具、Project 2000 和 ProjectCentral 计划工具、SQA 测

试工具以及 RequisitePro 需求管理工具等其他的一些工具。开始引入的时候是有些难度的,毕竟是工作方法产生了改变,一旦熟练了,就习惯了。目前它们已经成了大家在工作中必不可少的工具。

对工具的选择与购买需要把握好“够用即可”的原则。软件开发管理工具一般比较昂贵,如果一次性投资购买了比较昂贵的软件,可能软件的 80% 的功能用不上,等组织的管理提高到工具软件可以支持的较高的管理水平,已经是 2 年以后的事情了,而 2 年以后的版本也需要升级更新了。所以,没有必要为用不到的 80% 的功能提前投资。

1.1.4 CMM 与 UML、XP

最近,出现了很多轻量型的软件开发方法,例如水晶模型、适应模型以及极限编程等。其中以极限编程(XP)最为流行,Kent Beck 在 XP 的开篇之作《Extreme Programming Explained——Embrace Change》中首次提出了极限编程这一创新的软件过程方法论。那么 XP 与 CMM 到底有什么样的关系呢?

首先,来看 XP 的四个核心价值:

- 沟通:项目中发现的问题往往是由于开发人员与设计人员、设计人员与客户之间的沟通不畅造成的。因此,在 XP 项目中没有沟通是不可能的。
- 简单:XP 认为应该尽量保持代码的简单,只要它能工作就可以。Kent Beck 指出与其实现一个复杂的系统,不如设计一个能够满足目前需要的、简单的系统,因为用户所考虑的情况可能永远都不会发生。
- 反馈:尽快获得用户的反馈,并且越详细越好,使得开发人员能够保证自己的成果符合用户的需要。
- 勇气:这是最重要的核心价值。因为 XP 强调要“拥抱变化”,因此对于用户的反馈,要勇于对自己的代码进行修改,丢掉坏的代码。

这四个核心价值其实也是 CMM 的基本原则。CMM 重视沟通,因此要求通过评审等多种方法来达到有效沟通;CMM 重视简单,因此 CMM 要求定义过程时抓住本质的需要的信息,强调有用性和可用性;CMM 重视反馈,快速反馈可以认为是 CMM4 级的灵魂;CMM 重视勇气,从 CMM1 级的混沌无序到 CMM2 级的有效控制的转变中就需要有正视现实并不断加以改进的勇气。

其次,从 CMM2 级的 KPA 中来寻找 XP 的最佳实践。在 CMM2 级中,需要管理(RM)同 XP 的现场客户(On-site Customer)本质上是一致的,需求管理要求在确定需求时要取得客户代表的认可。随着需求的改变,软件计划、工作产品和相应活动都要随之改变,这和 XP 的持续集成(Continuous Integration)的思路基本一致。软件项目计划(SPP)、软件项目跟踪和监督(SPTO)和 XP 的计划博弈(Planning Game)、小型发布(Small Release)也有一致性。项目的计划在建立起来以后,需要根据项目的进展来进行调整,一成不变的计划是不现实和不可能的。因此,项目组需要控制风险、预见变化,从而制订有效、可行的项目计划。软件质量保证(SQA)组不太可能成为 XP 文化的一部分,但是 SQA 的作用和 XP 的成对编程(Pair Programming)有一定的相同点,不同的是 XP 的 SQA 压力来自内部,而 CMM 的 SQA 压力来自外部独立的 SQA 组。此外,XP 的代码规范(Code Standards)本身就是 SQA 的责任。配置管理(SCM)和 XP 的代码全体拥有(Collective Ownership)、持续集成(Continuous Integration)、小型