

译者的话

R·M·S 译文选 10 《国外软件可靠性资料汇编》中第八篇“评估承包商软件工程能力的一种方法”中给出的方法，经过多年试用和完善，于 1991 年发展成为软件工程过程能力成熟度模型，即 CMM1.0，又经过实践、改进，于 1993 年修改为 CMM1.1。该模型的原理和作用现已得到国际公认，它不仅可用来评估软件开发单位的能力，更重要的是可供软件开发单位自我评估，并找出提高本单位软件开发能力的最有效途径。国际标准化组织正在以该模型思想为基础，组织制定相应的一系列标准。

我国软件产业正处在亟待加速发展的时期，装备系统对软件可靠性提出了很高要求。在这种情况下，如何尽快提高软件开发单位的软件工程能力便成了紧迫的课题。统计表明，随着软件工程能力提高，软件开发的风险将显著降低，软件产品所含缺陷数将明显减少。因此，提高软件工程能力也是保证软件质量与可靠性的根本措施。

基于上述认识，我们感到将 CMM1.1 翻译出来，供广大软件开发单位参考，会有助于提高各自的软件工程能力，从而增强其市场竞争能力，也有利于保证软件质量与可靠性。由于水平所限，难免有谬误之处，恳请读者指正。

译者 1996·06

内容简介

本集包括有关 SEI 提出的 CMM 的叁篇重要文献。第一篇是“软件能力成熟度模型 1.1 版”；第二篇是“能力成熟度模型的关键实践 1.1 版”；第三篇是“软件过程成熟度提问单”。

第一篇包含以下内容：定义为理解 CMM 所需要的概念和简略介绍模型、描述五个成熟度等级及其基础原理、描述 CMM 的可操作定义和它的结构、简介有关组织如何运用 CMM 进行软件过程评估和软件能力评价，提出 CMM 未来的发展方向。

第二篇的开头是概述部分共四章：给出有关 CMM 及这篇文章的概述；简介 CMM 及其构成成分；介绍关键实践的版面安排；描述运用和解释关键实践的方法。概述之后按 CMM 的每个等级（除一级外），分别给出每个关键过程区域的简要描述、目标、及所包含的全部关键实践（按五个共同特点——执行约定、执行能力、执行的活动、测量和分析、验证实施——加以组织）。附录 C 中给出了关键实践的简略版本，它包括关键过程区域的简要描述、关键过程区域的目标，关键过程区域在执行的活动这一共同特点下全部关键实践的上层实践，供读者快速查阅用。第二篇的附录中还包含它所用的术语汇编。

第三篇是基于“能力成熟度模型 1.1 版”的评估工具。

译者 1996 · 06 · 20

致读者

最初软件能力成熟度模型是由一个专门小组研制的,小组人员化费许多时间讨论模型和它的特点,然后以 CMM V1.0 的形式记录下来。该小组由下列人员组成。Mark Paulk, BillCurtis, Mary BethChrissis, Edward Averil, Judy Bamberger, Tim Kasse, MikeKonrad, Jeff Perdue, Charlie Weber, 和 Jim Withey。

本文以 SEI 软件过程计划 (Program) 的首任领导 Watts Humphrey 的版本为基础。曾几易其稿才最后形成本文。Jim Withey, Mark Paulk, 和 Cynthia Wise 于 1990 年编写了最早的草稿。Watts HumPhrey 提供了本文件的第二稿,最后 Mark Paulk 接过该文,作为编写领导直到结束。Mary Beth Chrissis 和 Bill Curtis 帮助 Mark 于 1991 年 8 月完成一份修定稿,即 CMM V1.0 版本。Mark Paulk 编写了又一份修定稿 CMM V1.1 版本,即将技术报告。

一些人员在不同阶段曾对文中所阐述的概念作出贡献。他们包括 JOe Besselman, MarilynBush, Anlta Carleton, Marly Carlson, Batty Delmel, Suzle Garcla, Richard Kauffold, Sieve Masters, Mary Merrill, Jim Over, George Pandellos, Jane Slegel. 和 Charlle Weber。我们感谢 Todd Bowman, Dorothy Josephson, Debble Punjack, Carolyn Tady, Marcla Theoref, Andy Tsounos 和 David White 在行政管理方面的帮助,以及 Mary Beth Chrississ, Suzanne Couturiaux 和 Bill Pollak 在编辑方面的协助。美国研究协会 (American Institute for Research) 的 Renne Dutkowski 对本文件的设计提出了建议。

1986 年 11 月,在 Mitre 公司帮助下,软件工程研究所 (SEI) 开始研制能帮助组织改善其软件过程的成熟度框架。此项工作最初是应联邦政府的要求提供一种评估其软件承包商能力的办法而进行的。1987 年 9 月,SEI 发表了过程成熟度框架的简要描述 [Humntirey 87a] 和成熟度提问单 [HumPhrey 87bSEI] 打算使软件成熟度提问单成为识别组织软件过程需改进区域的简单工具。遗憾的是,人们常把该提问单看成为“模型”,而不是探究过程成熟度问题的载体。

在对软件过程成熟度框架和成熟度提问单的初始版本取得四年经验之后,SEI 已将软件过程成熟度框架进化为软件能力成熟度模型 (CMM) [Paulk, Weber 91]。CMM 充分吸取了从软件过程评估中所获得的知识 and 工业部门及政府所提供的广泛的反馈意见。通过精心推敲成熟度框架,业已形成能给组织在建立过程改进大纲方面提供更有效指导的模型。

在 1991 年和 1992 年期间,软件界评审和使用了 CMM 的最早发布,即版本 1.0。1992 年 4 月举办了 CMM V1.0 的研讨会,约 200 名软件专业人员参加了会议。CMM 的这个 1.1 版本,是吸取该研讨会和软件界反馈意见的结果。

CMM 是系统地构造一组工具的基础,这组工具包括在软件过程改进中很有用的成熟度提问单。需牢记的最重要的一点是,正是模型而不是提问单是改进软件过程的基础。本文打算向读者介绍 CMM V1.1。

本文的目的

本文对软件能力成熟度模型进行技术性综述，反映版本 1.1。具体讲，本文描述五个成熟度等级的过程成熟度框架、组成 CMM 的结构成分、在实践中如何运用 CMM、及 CMM 的未来发展方向。本文是了解 CMM 的最好材料之一，并能澄清与 SEI 所提倡的软件过程成熟度有关的一些误解。

SEI 一直与工业部门和政府一起提炼和扩展该模型，鼓励软件组织关注 CMM 而不是成熟度提问单。为了促进这种关注，SEI 已经研究并正在制定一套过程产品。本文 [Paulk 93a] 和“能力成熟度模型的关键实践 1.1 版” [Paulk 93b] 一起构成 CMM V1.1。后者描述 CMM 每一等级的关键实践。本文叙述软件过程成熟度的基础原理，并打算帮助软件组织运用 CMM V1.1 作为改进其软件过程成熟度的指南。

本文的对象

本文介绍 CMM 及其相关产品。所以，任何打算学习 CMM 的人都应阅读本文。可是本文假定，读者已具有开发和（或）维护软件的某些知识和经验，并对当前软件界所面临的问题有所了解。

本文可有以下几种用途：

- ✍✍ 任何人用来了解开发和维护软件的有效过程中的关键实践，
- ✍✍ 任何人用来识别为达到 CMM 下一个成熟等级所必须的关键实践，
- ✍✍ 组织用来了解和改进其有效开发软件的能力，
- ✍✍ 采购组织或主承包商用来了解让某个具体组织完成合同工作的风险，
- ✍✍ SEI 采用本文作为拟定成熟度提问单中提问的基础，
- ✍✍ 指导者用来指导群组（team）为执行软件过程评估或软件能力评价作好准备。

本文的组织方式

本文共有五章

第一章定义那些为理解 CMM 及其动机和目的所需要的概念。

第二章描述 CMM 五个等级和它们的基础原理。

第三章描述 CMM 如何被构造造成一些关键过程区域，按共同特点进行组织，用关键实践加以刻画。

第四章在高层次上综述 CMM 如何指导软件评估、软件能力评价和过程改进大纲。

第五章作为本文的结束，描述了 CMM 及其相关产品的未来发展方向。

CMM 的其它产品

虽然孤立地阅读本文是可以的，但按照设计，本文是其它产品的出发点。本文及其相关产品能帮助读者理解和使用 CMM。所有这些基于 CMM 的产品都已经或将被系统地从此模型中导出。在写本文时，虽然产品的初始版本都已处在先导性试验和发布等各种阶段，但这些产品中的绝大多数尚未形成最终可用产品。

基于 CMM 的产品集合包括一些诊断工具 ,供软件过程评估组 和软件能力评价组 用来识别组织软件过程的强项、弱项和风险。其中最著名的可能是成熟度提问单。软件过程评估和软件能力评价的方法及培训 1 工作也依靠 CMM。

这些产品的用户形成一个致力于改进其软件过程成熟度的团体。SEI 将继续与此软件团体合作 , 改善此模型及其相关产品。

如何获得更多的信息

为得到进一步的有关 CMM 及其相关产品的信息 , 包括有关 CMM 的培训以及如何进行软件过程评估和软件能力评价 , 请与以下单位接触 :

可直接从国防技术信息中心 (DTIC)、国家技术信息局 (NTIS) 和研究查询股份有限公司 (RAI) 获得 SEI 的技术报告 , 例如本文和 “能力成熟度模型的关键实践 1.1 版”。具体请与以下地址联系 :

<ftp://ftp.sei.cmu.edu>

通过 Internet 网也可得到 SEI 报告。从 Internet 上的 Unix 系统采用不具名的 ftp :

READ ME 文件中包括可得到何种文件的信息。以类似的方式可得到 SEI 其它的出版物。

软件过程评估量由一个经相识的软件专业人员小组所作的一项鉴定 , 旨在规定组织当前软件过程的状态、组织所面对的与软件过程有关的具有高优先级的问题 , 及获用组织对软件过程改进的支持。

软件能力评价是由一个经培训的软件专业人及小组所作的一项鉴定 , 旨在鉴别出合格的软件工作承包商 , 或者监控现有软件工作中所采用的软件过程的状态。

SEI FTP 机器地址是 128.237. 2.179

1 过程成熟度框架

人们关于通过应用新的软件方法和技术能提高生产率和质量的希望，经过了 20 年仍未实现，于是工业和政府组织意识到，他们的基本问题是不能管理其软件过程 [DoD 87]。在无纪律的、混乱的项目状态下，组织不可能从较好的方法和工具中获益。许多组织中、项目常极度迟后、预算翻番 [Siegel 90]。在这些情况下，组织经常不提供帮助项目避免这些问题所需要的基础设施和资助。

可是，即使在无纪律的组织中，个别软件项目仍能生产出优质产品。这些项目的成功一般是通过工作组的杰出努力，而不是通过重复使用具有成熟软件过程的组织的经过验证的方法。在没有全组织范围的软件过程的条件下，重复结果的取得完全取决于能使同样人员去做下一个项目。仅仅建立在可得到特定人员上的成功不能为全组织生产率和质量的长期提高打下基础。只有通过建立有效软件工程和管理实践等过程基础设施方面，坚持不懈的努力，才能取得不断改进。

1.1 不成熟和成熟软件组织的比较

为了设置合理的过程改进目标，必须了解不成熟软件组织和成熟软件组织间的区别。在不成熟软件组织中，软件过程一般由实践者及其管理者在项目进程中临时拼凑而成。即使已规定了软件过程，也不能严格地遵守和贯彻它。不成熟的软件组织是反应式的，通常经理们集中精力于解决即时危机（称为消防）。由于制定进度和预算不是基于现实的估计，因而超出进度和预算已成惯例。当硬性规定限时时，为满足进度要求，常在产品功能和质量上作出让步。

在不成熟组织中，不存在判断产品质量或者解决产品或过程问题的客观基础。因此，产品质量难以预测。当项目进度迟后时，常缩短或取消像评审和测试这些旨在提高质量的活动。可是，一个成熟软件组织具有全组织范围的管理软件开发和维护过程的能力。软件过程被正确无误地通知到现有职员和新雇员，工作活动均按照已规划的过程进行。强制式的过程适于使用 [Humphrey 91b]，而且和实际进行工作的方式相一致。需要时就对这些已定义的过程进行更新，并且通过可控的先导性试验和（或）费效分析使其得到改进。在已定义的过程中，遍及所有项目和在整个组织中，所有的岗位及其职责都是清楚的。

成熟组织中，经理监控软件产品的质量和顾客的满意程度。在判断产品质量和分析产品及过程问题方面有客观的、定量的基础。进度和预算是基于以前的性能数据，因而是现实的；通常都能达到产品的成本、进度、功能和质量的预期结果。一般讲，成熟组织一致地遵循一个有纪律的过程，因为所有的参加者都了解这样做的价值，而且存在支持该过程的必要基础设施。

要从这些有关不成熟和成熟软件组织的观察资料中获益，需要构造一个软件过程成熟度框架。该框架描述一条从无序的、混乱的过程到成熟的、有纪律的软件过程的进化途径。没有此框架改进大纲可能显得无效，因为尚未建立起支持连续改进的必不可少的基础。软件过程、软件过程能力、软件过程性能和软件过程成熟度等概念集为一体，形成了软件过程成熟度框架，在下面几节中将定义所有这些概念。

1.2 构成过程成熟度基础的基本概念

根据韦氏字典，一个过程是“某物生产的操作体系……能导致结束或得到结果的一系列活动、变更、或操作。”IEEE定义过程为“为实现给定目标所执行的一序列的步骤”[IEEE - STD - 610]。所以，一个软件过程可以定义为，人们用以开发和维护软件及其相关产品（例如，项目计划、设计文档、代码、测试用例、用户手册等等）的一组活动、方法、实践和变换。随着一个组织的成熟，其软件过程得到更好的定义，并在整个组织内得到更一致的实施。

软件过程能力描述通过遵循软件过程能够实现预期结果的程度。一个组织的软件过程能力提供一种预测该组织承担下一个软件项目时最可能的预期结果的方法。

软件过程性能表示遵循软件过程所得到的实际结果。所以，软件过程性能关注已得到的结果，而软件过程能力则关注预期结果。由于一个特定项目的属性和执行该项目的环境所限，该项目的实际性能可能并不充分反映组织的整个过程能力，即项目的能力受限于它的环境。例如，项目在应用领域或所采用的技术上的根本改变可能造成其职员正处于学习状态，这使得他们的项目能力和性能远达不到该组织的整个过程能力。

软件过程成熟度是一个特定过程被明确地定义、管理、测量、控制、并且是有效的程度。成熟度意味着能力上的增长潜力，并且表明一个组织软件过程的丰富性和在遍及组织的项目中运用它时的一致性。在成熟组织中，通常通过文档和培训使全组织有关人员对软件过程都能很好的了解，并且使该过程得到其用户不断的监控和改进。一个成熟软件组织的能力是已知的。软件过程成熟度意味着，由于运用组织的软件过程使过程纪律性一致增强，从而其软件过程所导致的生产率和质量能随时间的推移得到改进。

随着软件组织的软件过程成熟度的提高，组织通过方针、标准和组织机构将其软件过程规范化。规范化需要建立一种支持经营方法、实践和规程的基础设施及社团文化，使得在最初定义方法、实践和规程的人员离去后，它们仍能继续下去。

1.3 能力成熟度模型概述

尽管软件工程师和经理通常都非常清楚存在的问题，但他们对哪项改进最为重要可能意见不一致。如果对改进工作没有一项有机的改进策略，要在管理者和专业人员间就首先需进行哪些改进活动形成一致意见是很困难的。为了使过程改进工作能不断取得成效，必须设计一条进化路径，使组织的软件过程成熟度按阶段逐步提高。软件过程成熟度框架 [HumPhrey87a]将这些阶段排序，使每个阶段上的改进能为下一阶段的改进打下基础。所以，从软件过程成熟度框架导出的改进策略，对过程不断改进的历程提供了一份导引图。它指导组织前进并识别出其缺陷，但它并不企图对有问题的项目提供快速修复措施。

软件能力成熟度模型给软件组织提供如何增加对其开发和维护软件过程的控制，如何向软件工程和管理的优秀文化进化等方面的指导。设计CMM是为了指导软件组织，通过确定当前过程的成熟度和通过识别出对软件质量和过程改进至关重要的少数问题，来选择其过程改进策略。通过关注一组有限的活动，并为实现它们而积极工作，组织能稳步地改善其全组织的软件过程，使其软件过程能力持续不断地增长。

CMM 分阶段的结构基于已有 60 多年历史的产品质量原理。Walter Shewart 于 30 年代公布了统计质量控制原理。W·Edward Deming [Deming 86] 和 Joseph Juran [Jura 88, Juran89] 的著作又进一步发展和成功地论证了该原理。SEI 已将这些原理改编成为成熟度框架，该框架为软件过程定量控制建立了项目管理和项目工程的基本原则，这是过程不断改进的基础。

ITT 的 philip Crosby 在其书 “Quality is Free” [Crosby 79] 中首先提出将质量原理改编为成熟度框架的思想。Crosby 的质量管理成熟度网格描述了采用质量实践时的五个进化阶段。该成熟度框架又由 IBM 的 Ron Radice 和他的同事们在 Watts Humphrey 指导下进一步改进以适应软件过程 [Radice 85]。1986 年，Humphrey 将此成熟度框架带到软件工程研究所加上了成熟度等级的概念，研制成当前整个软件产业界所使用的框架的基础。

Humphrey 的成熟度框架早期版本发表在 SEI 技术报告 [Humphrey 87a, Humphrey 87b]、文章 [Humphrey 88] 和书 “Managing the software Process” [Humphrey 89] 中。1987 年发表了初步的成熟度提问单 [Humphrey 87b]，它作为工具给组织提供软件过程特征化的一种方法。1987 年又进一步研制出软件过程评估和软件能力评价两个方法，以便估计软件过程成熟度。自 1990 年以来，在政府和工业部门许多人的帮助下，SEI 基于几年来将框架运用到软件过程改进方面的经验，已经进一步扩展和精炼了该模型。

2 软件过程成熟度的五个等级

过程的不断改进基于许多小的、进化的步骤，而不是革命性的创新 (Imai 86)。CMM 提供了一个框架，将这些进化步骤组织成五个成熟度等级，它为过程不断改进奠定了顺序渐进的基础。这五个成熟度等级定义了一个有序的尺度，用以测量组织软件过程成熟度和评价其软件过程能力。这些等级还能帮助组织对其改进工作排出优先次序。

成熟度等级是妥善定义的在向成熟软件组织前进途中的平台。每一个成熟度等级为过程继续改进提供一个台基。每一等级包含一组过程目标，当目标满足时，能使软件过程的一个重要成分稳定。每达到成熟度框架的一个等级，就建立起软件过程的一个不同的成分，导致组织过程能力的增长。

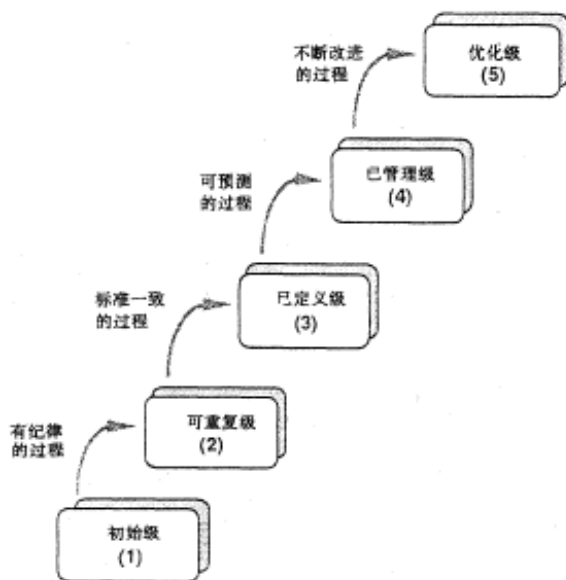


图 2.1 软件过程成熟度的五个等级

将 CMM 组织成如图 2.1 所示的五个等级，对旨在增加软件过程成熟度的改进行动 (action) 按优先级排序。图 2.1 中带有标记的箭头，指示在成熟度框架的每一步骤上，组织应予以规范化的过程能力的类型。

下列的五个成熟度等级的特性突出说明在每个等级上过程的主要变化。

1) 初始级 软件过程的特点是无秩序的, 偶尔甚至是混乱的。几乎没有什么过程是经过定义的, 成功依赖于个人的努力。

2) 可重复级 已建立基本的项目管理过程去跟踪成本、进度和功能性。必要的过程纪律已经就位, 使具有类似应用的项目。能重复以前的成功。

3) 已定义级 管理活动和工程活动两方面的软件过程均已文档化、标准化、并集成到组织的标准软件过程。全部项目均采用供开发和维护软件用的组织标准软件过程的一个经批准的剪裁版本。

4) 已管理级 已采集详细的有关软件过程和产品质量的度量。无论软件过程还是产品均得到定量了解和控制。

5) 优化级 利用来自过程和来自新思想、新技术的先导性试验的定量反馈信息, 使持续过程改进成为可能。

2.1 成熟度等级的行为特征

等级 2 到等级 5 的成熟度能通过以下各项对其特征加以描述: 组织为建立或改进软件过程所进行的活动、对每个项目所进行的活动和所产生的横跨各项目的过程能力。这里之所以列出等级 1 的行为特性, 是为了给较高成熟度等级提供比较过程改进的基础。

2.1.1 等级 1——初始组

在初始级上组织一般不提供开发和维护软件的稳定环境。当组织中缺乏健全的管理实践时, 不适当的规划和反应驱动体系会降低由良好的软件工程实践所带来的效益。

在危机时刻, 项目一般抛弃预定的规程, 回复到仅作编码和测试。成功完全依赖于有一个杰出的经理及一支有经验的、战斗力强的软件队伍。偶尔, 有能力的、坚强的软件经理能经受住要他们在软件过程中走捷径的压力, 但是当他们离开项目后, 他们能使过程稳定的影响也随之消失。甚至一个强的工程过程也不能克服由于缺乏健全的管理实践所造成的不稳定。

等级 1 组织的过程能力是不可预测的, 因为随着工作进展软件过程经常被改变或修定 (即过程是无秩序的 (ad hoc))。进度、预算、功能性和产品质量一般是不可预测的。性能依赖于个人的能力, 且随个人固有的技能、知识和动机的不同而变化。等级 1 组织几乎没有明显的稳定的软件过程, 只能通过个人的能力而不是组织的能力去预测性能。

2.1.2 等级 2——可重复级

在可重复级上, 已建立管理软件项目的方针和实施这些方针的规程。基于在类似项目上的经验对新项目进行规划和管理。达到等级 2 的目的是使软件项目的有效管理过程制度化, 这使得组织能重复在以前项目上所开发的成功实践, 尽管项目所实施的具体过程可能不同。一个有效过程可特征化为实用的、已文档化的、已实施的、已培训的、已测量的和能改进的。

等级 2 组织中的项目已设置基本的软件管理控制。实际可行的项目约定是基于对以前项目的观察结果和当前项目的需求。项目的软件经理跟踪软件成本、进度和功能性; 在满足约定方面的问题, 一旦出现就被识别。对软件需求和为实现需求所开发的工作产品建立基线, 并控制其完整性。软件项目标准均已确定, 并且组织能保证准确地执行这些标准。如果有子

承包商的话，软件项目与他们一起努力建立一种强有力的顾客——供应商关系。

等级 2 组织的过程能力可概括为有纪律的，因为软件项目的规划和跟踪是稳定的，能重复以前的成功。由于遵循基于以前项目性能所制定的切实可行的计划，项目过程处在项目管理系统的有效控制之下。

2.1.3 等级 3——已定义级

在已定义级上，全组织的开发和维护软件的标准过程已文档化，包括软件工程过程 and 软件管理过程，而且这些过程被集成为一有机的整体。在 CMM 中的所有地方，均称此标准过程为组织的标准软件过程。等级 3 上所建立（适当时，经更改）的过程，被用来帮助软件经理和技术人员工作得更有效。组织在使其软件过程标准化时，利用有效的软件工程实践。存在一个负责组织的软件过程活动的组，例如软件工程过程组（SEP）[Fowler 90]。实施全组织的培训计划以保证职员和经理具有履行其职责所必须的知识和技能。

项目通过剪裁组织的标准软件过程去建立他们自己定义的软件过程，它说明项目独有的特征。在 CMM 中，这种剪裁后的过程称作项目定义软件过程。一个已定义软件过程包含一组协调的、集成的、妥善定义的软件工程过程和管理过程。妥善定义的过程可特征化为具有准备就绪判据、输入、标准、进行工作的规模、验证机制（例如同行评审）、输出、以及完成判据。因为软件过程已妥善定义。管理者就能洞察所有项目的技术进展。

等级 3 组织的软件过程能力可概括为标准的和一致的，因为无论软件工程活动还是管理活动，过程都是稳定的和可重复的。在所建立的产品线内，成本、进度和功能性均受控制、对软件质量也进行跟踪。这种过程能力建立在整个组织范围内对已定义过程中的活动、角色和职责的共同理解之上。

2.1.4 等级 4——已管理级

已管理级上，组织对软件产品和过程都设置定量的质量目标。作为组织测量大纲的一部分，对所有的项目都测量其重要的软件过程活动的生产率和质量。利用一个全组织的软件过程数据库收集和分析从项目定义软件过程中得到的数据。等级 4 上的软件过程均已配备有妥善定义的和一致的度量。这些度量为定量地评价项目的软件过程和产品打下基础。

项目通过将其过程性能的变化限制在定量的可接受的范围之内，实现对其产品和过程的控制。可以将过程性能方面有意义的变化与随机变化（噪声）区别开来，特别在所建立的产品线内。开发新应用领域的软件所带来的风险是已知的，并得到精心的管理。

等级 4 组织的软件过程能力可概括为可预测的，因为过程是已测量的并在可测的范围内运行。该等级的过程能力使得组织能在定量限制的范围内预测过程 and 产品质量方面的趋势。当超过限制范围时，采取措施予以纠正。软件产品具有可预测的高质量。

2.1.5 等级 5——优化组

在优化级，整个组织集中精力进行不断的 process 改进。为了预防缺陷出现，组织有办法识别出弱点并预先针对性地加强过程。在对新技术和所建议的组织软件过程的更改进行费效分析时利用有关软件过程有效性的数据。识别出采用最好软件工程实践的技术创新并推广到整个组织。

等级 5 组织的软件项目群组分析缺陷以确定其发生的原因。为了防止已知类型的缺陷再次出现他们认真评价软件过程，同时将经验教训告知其它项目。

等级 5 组织的软件过程能力可特征化为不断改进，因为这些组织为扩大其过程能力的范围进行着不懈的努力，因而不断改善其项目的过程性能。既通过在现有过程中增量式前进的办法，也通过采用新技术、新方法的革新办法，使改进不断出现。

2.2 理解成熟度等级

CMM 描述那些预期能刻画特定成熟度等级组织的本质（或关键）属性，在此意义下，它是一个描述模型。CMM 的详细实践描述组织在政府签约环境中从事大规模项目时预期具有的行为规范类型的特征，在此意义下，它是一个规范模型。这样做的意图是，让 CMM 处在充分高的抽象层次上，使它不会过多限制一个组织如何去实施软件过程：CMM 仅仅描述通常所期待的软件过程的本质属性。

在所有运用 CMM 的环境中，均应该采用对于实践的合理解释。当组织的经营环境明显地不同于大的签约组织的经营环境时，必须运用有根据的专业判断，合理地解释 CMM。

CMM 不是处方，它并不告诉组织如何进行改进。CMM 描述在每个成熟度等级上的组织的特性，并不规定达到成熟度等级的具体方法。从等级 1 到等级 2 可能需用几年的时间，而在其它等级间的升迁通常花费约 2 年时间。

软件过程改进发生在以下的上下文内：组织的战略计划和经营目标、它的组织机构、所使用的技术、它的社会文化和它的管理体系。CMM 集中注意力于全面质量管理工作的过程方面；成功的过程改进意味着还需致力于超出软件过程范围之外的其它方面（例如：与改变组织文化有关的人员问题，组织文化使得待实施和规范化的过程的改进成为可能）。

2.2.1 理解初始级

尽管等级 1 组织常被特征化为具有无秩序的、甚至是混乱的过程，但它们常常开发出能工作的产品，即使它们可能超出预算和进度要求。等级 1 组织的成功依赖于组织中人员的能力和杰出的努力。对所有成熟度等级的组织来说，选择、雇用、培养、和（或）留住有能力的人都是重要议题，但它们多半在 CMM 的考虑范围之外。

2.2.2 理解可重复级和已定义组

随着项目规模和复杂性的增长，注意力逐渐从技术问题转向组织体系和管理问题——过程成熟度的焦点问题 [Siegel 90, DoD 87, GAO - 92 - 48]。通过将最优秀职员所取得的经验教训纳入已文档化的过程、通过培育有效地执行那些过程所必须的技能（通常以培训的方式）、以及通过以向完成任务的人们学习的方式所得到的不断改进，过程能使人们工作得更有效。

为达到等级 2，管理者必须集中注意力于他们自己的过程以便实现一个有纪律的软件过程。等级 2 是等级 3 的基础，因为在处理等级 3 上的技术和组织体系问题之前，关注焦点必须放在管理者改进其过程的行为上。在实现等级 2 的过程中，通过将项目管理过程编制成文档并遵照执行，管理者建立起领导地位。

在等级 2 组织中不同项目的过程可以不同；为了达到等级 2，组织体系上的要求是存在

指导项目建立合适的管理过程的方针。巴文档化的规程为这些一致的过程打下基础，在培训和软件质量保证工作的帮助下，能在全组织范围内使这些过程规范化。

通过对整个软件过程加以定义、集成和文档化，等级 3 建立在项目管理的基础之上。在此情况下，集成意味着一个作业的输出顺利地成为下一个作业的输入。当作业间存在不匹配现象时，在软件过程的规划阶段它们就能得到识别和处理，而不是等到实施过程中遇到它们时才加以解决。等级 3 上的一个挑战是构造这样一些过程，这些过程允许软件人员开展工作 时仍有一定的自由度 (Humphrey 91b)。

2.2.3 理群已管理级和优化组

成熟度等级 4 和等级 5 对软件产业界来说还是一个相对未知的领域。仅存在不多几个等级 4 和等级 5 的软件项目和组织 [Humphrey gla Kitson 92]。由于现有的实例太少，以致不能导出有关等级 4 和等级 5 组织特征的一般结论。通过与其它产业类比和利用软件产业界中能显示这些等级过程能力的少数例子，这些等级的特征已经得到定义。

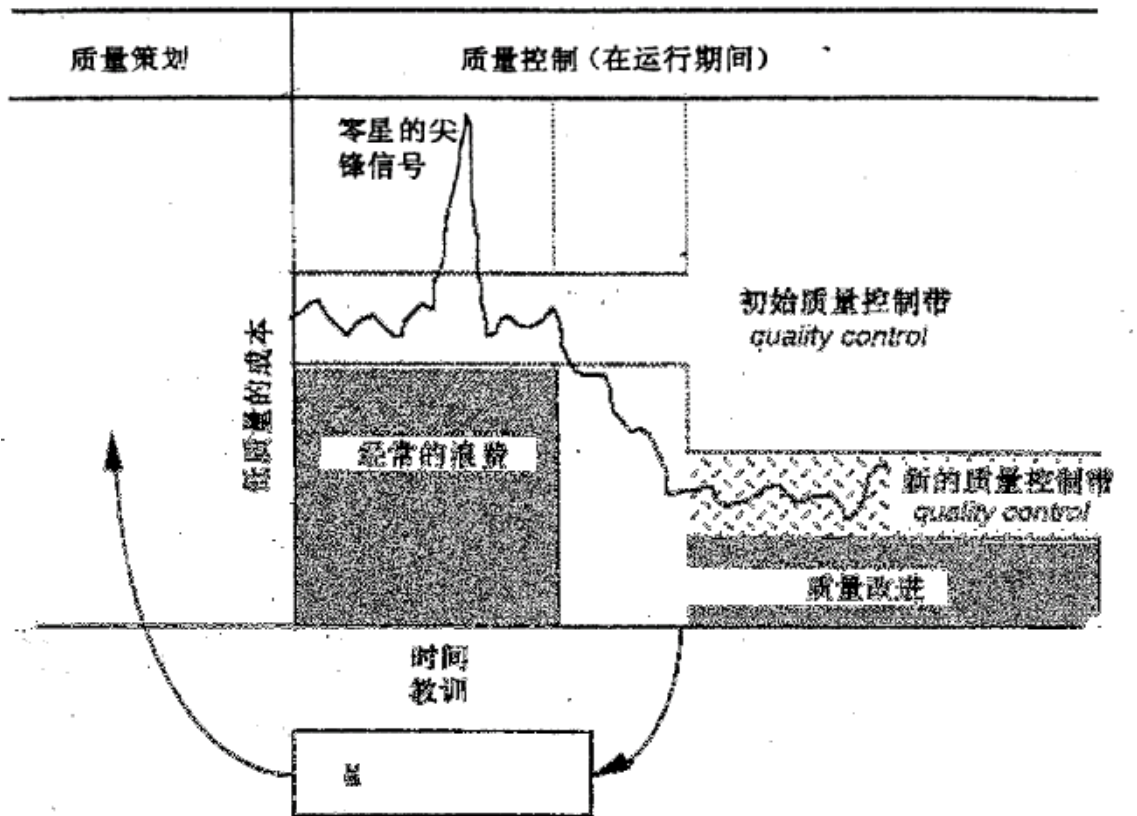


图 2.2 朱兰三部曲 (Juran Trilogy) 图：质量策划、质量控制和质量改进

图 2.2 的例子说明统计过程的基本概念，等级 4 和等级 5 的许多特征建立在这些概念之上。朱兰三部曲 (Juran Trilogy) 图 [Juran 88] 显示过程管理的主要目标。

Juran 将质量管理分成三个基本的管理过程 [Juran 88]。质量策划的目的是向运作人员，即软件生产者提供生产能满足顾客需求的产品的方法。操作人员生产出产品，但由于质量缺陷不得不做某些退工。这种浪费是经常性的 (因为过程被设计成那种样子)；执行质量控制

是为了防止事情恶化。过程中的零星的尖峰，如图 2.2 所示，代表消防活动。经常性的浪费提供了改进的机会，抓住这个机会进行工作就叫做质量改进。

质量改进的第一个职责，也是等级 4 的关注焦点是过程控制。对软件过程进行管理以致它能稳定运行在质量控制带内。虽然必定还有一些经常性的浪费，测量结果中可能还有尖峰信号需要控制，但是系统在整体上是稳定的。正是在这些地方，控制变化的特殊原因这个概念开始起作用。因为过程既稳定又可测，所以当出现某些例外情况时，能够识别出并阐述变化的“特殊原因”。

质量改进的第二个职责，也是等级 5 的关注焦点，是过程的不断改进。为了提高质量，变更软件过程，随后质量控制带移动。建立新的能减少经常性浪费的性能基线。在改进这样一个过程中所得到的经验教训被运用到规划未来的过程中去。正是在这些地方，消除变化的一般原因这个概念开始引人注目。仅仅由于随机变化，在任何系统中均以返工的形式存在经常性的浪费。浪费是不能容忍的；有组织地减少浪费的努力导致更改系统，也就是通过克服造成低效性的“一般原因”来改善过程以防止出现浪费。

我们期望，达到 CMM 最高成熟度等级的组织，具有能在预测的成本和进度的限制下生产出极端可靠的软件的过程。随着对较高成熟度等级理解的加深，将对已有的关键过程区域加以提炼，也可能给模型添加些内容。CMM 是从制造业中所产生的过程思想推导出的，但是软件过程不像制造过程那样以重复问题占主导地位。软件过程以设计问题占主导地位，而且是知识密集性的活动[CrutiS 88]。

2.3 软件过程的可视性

软件工程师们对项目状态有详细而深入的了解，因为他们掌握有关项目状态和性能的第一手材料。可是，对大项目而言，他们的洞察力常来源于在其职责范围内的个人经验。项目以外的不掌握第一手材料的人，例如高级经理，对项目过程缺乏可视性，为得到他们监控进程所需要的信息只得依靠定期的评审。图 2-3 显示在过程成熟度的每个等级上，提供给管理者的对项目状态和性能的可视性等级。每个后继的成熟度等级逐步增加地提供软件过程更好的可视性。

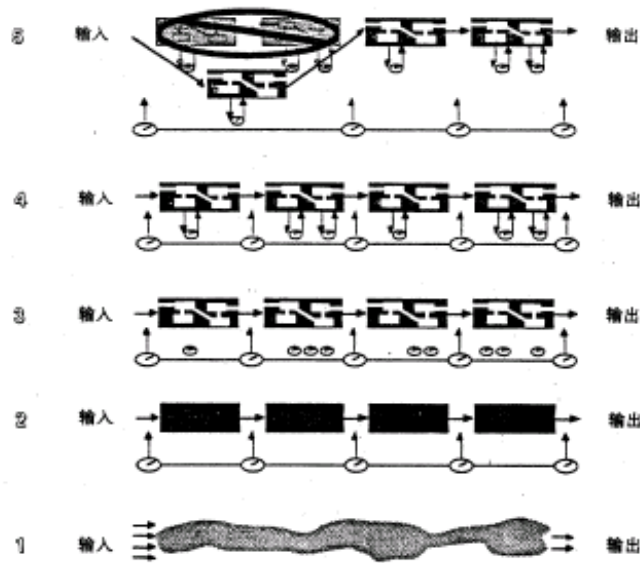


图 2-3 管理者对每个成熟度等级的软件过程的可视性

等级 1 的软件过程是一种无组织的整体——一个黑盒，项目过程的可视性是有限的。由

于没有很好地定义活动阶段的划分，经理们在确定项目进程和活动状态上极为困难

需求以失控的方式进入软件过程，然后生产出产品。软件开发常被看作为不可知的魔术，特别对不熟悉软件的经理而言更是如此。

等级 2 上，顾客需求和工作产品受到控制，已经建立起基本的项目管理实践。这些管理控制使得在规定的场合项目可视。构造软件的过程可看作为一个接一个的黑金子，这使得随着活动在金子间流动，在各过渡点（项目里程碑）上具有管理可视性。尽管管理者可能不知道盒子内发生事情的细节，但是过程的产品和用以证实过程正在适当进行的检验点都是确定且已知的。当问题出现时，管理者能对它们作出反应。

等级 3 上，盒子的内部结构（即项目定义软件过程中的作业）是可视的。内部结构表示组织的标准软件过程用于具体项目的方式。经理们和工程师们都了解他们在过程中的角色和职责，并且知道他们的活动如何在合适的细节层次上相互影响，管理者预先作好应付可能发生风险的准备。由于已定义的过程提供对项目活动很好的可视性，项目外的个人能获得精确的、即时的状态更新信息。

等级 4 上已定义的软件过程被配备上度量，并得到定量地控制。经理们能够度量进程和问题。他们作决策时，有一个客观的定量的基础。随着过程的变异性越来越小，他们预测结果的能力稳定地成为越来越准确。

等级 5 上，为了提高生产率和质量，以受控的方式对构造软件的新的和已改进的方法进行不断的试验。因为无效的或者易出错的活动均得到标识并加以替代或修改，有纪律的更改成为一种生活方式。不仅对现存过程明察秋毫，而且洞察力扩展到能了解过程潜在变化的影响。经理们能定量地估计更改的影响和有效性，然后跟踪它们。

2.4 过程能力和性能预测

一个组织的软件过程成熟度能帮助预测一个项目达到其目标的能力。等级 1 组织中的不同项目在达到成本、进度、功能和质量等指标的能力上差别很大。正如图 2.4 中的例子所示，随着组织软件过程成熟，在满足预定目标方面能观察到三个改进之处。

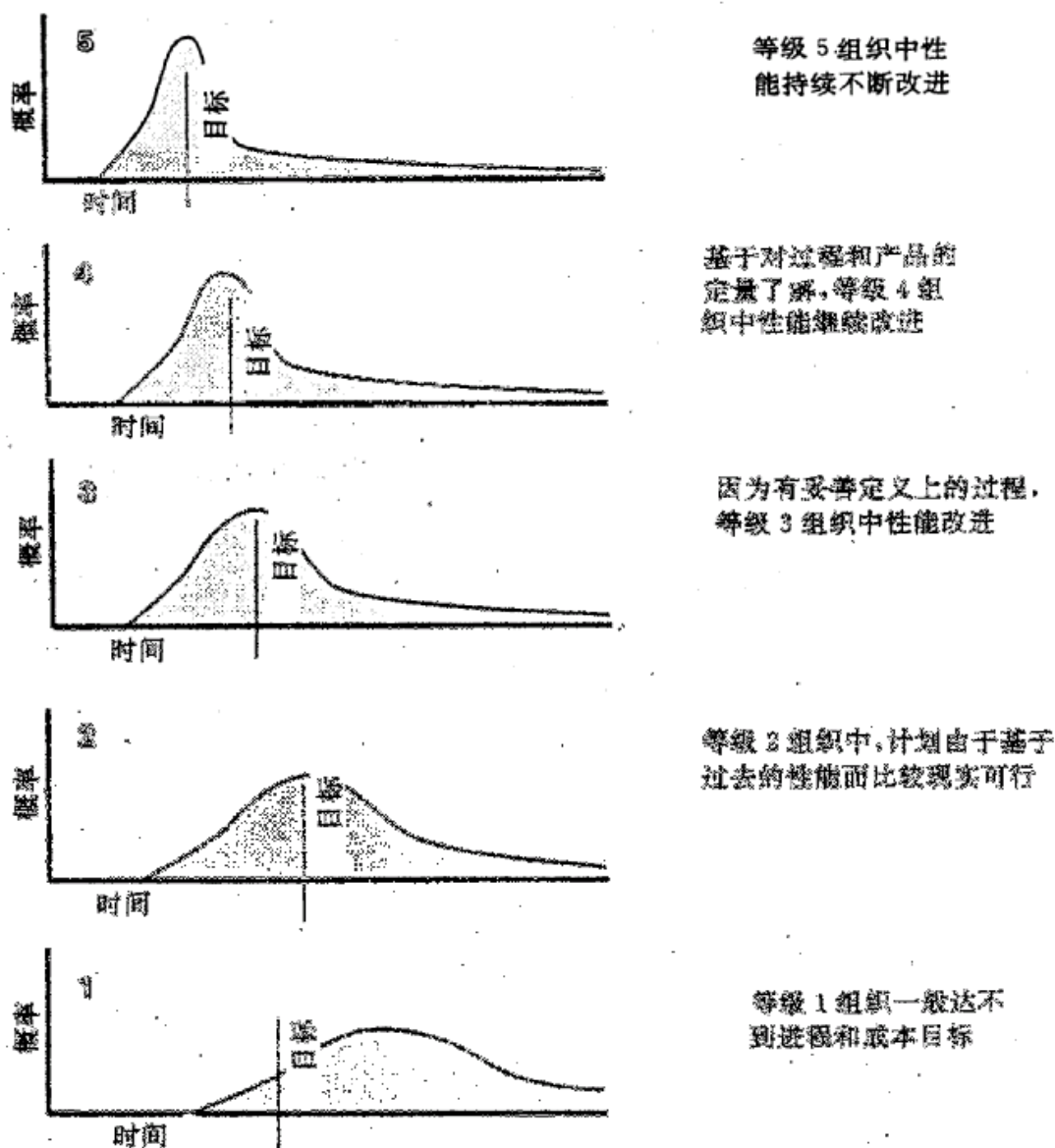


图 2.4 成熟度等级所指示的过程能力

首先, 随着成熟度增长, 所有项目的预定目标结果与实际结果间的差异减小。例如, 如果有十个相同规模的项目预定在 5 月 1 日交付, 那么随着组织的成熟, 它们交付的平均日期会越来越靠近 5 月 1 日。等级 1 组织常常远迟后于其进度表规定的交付日期, 而等级 5 的组织应该能相当精确地满足预定日期要求。这是因为等级 5 上组织采用仔细构造的、在已知参数范围内运行的软件过程, 而且确定预定日期是基于他们所具有的有关其过程的大量数据, 以及运用数据时的性能。(在图 2.4 中, 用预定日期线右边曲线下的面积大小表示这一点。)

其次, 随着成熟度增长, 实际结果相对预定目标结果的偏差范围减小。例如, 等级 1 组织中, 对具有类似规模项目的交付日期是不可预测的, 其变化很大。而等级 5 组织中的类似项目的交付日期在小得多的范围内变化。在最高成熟度等级上变化范围很小的原因是所有的项目实际上均在接近组织的有关成本、进度、功能和质量等过程能力的受控参数的范围内

运行。(图 2.4 中,以集中于预定目标线附近的面积大小说明这点。)

第三,随着组织成熟度的增加,预定目标结果得到改善。这就是说,随着软件组织的成熟,成本降低,开发时间缩短、生产率和质量提高。在等级 1 上的组织,其开发时间可能十分长,因为必须完成大量的用以纠正错误的返工。相反,等级 5 组织采用不断改进过程和缺陷预防技术增加过程有效性和消除费钱的返工,使得开发时间得以缩短。(图 2.4 中,这点反映在预定目标线在指向原点方向上的水平移动。)

图 2-4 之中所表示的在预测项目结果方面的改进基于以下假定,即随着噪声(通常以返工形式出现)从软件过程中消除,软件项目结果更加可以预测。但是,无先例的系统会使情况复杂化,因为新的技术和应用问题增加可变性,从而降低过程能力。即使在无先例系统的情况下,与比较不成熟的组织相比,较成熟的组织管理和工程实践的特征能在开发周期的较早阶段帮助识别和阐述问题。由于较早识别出缺陷,能消除后面阶段的返工,从而提高项目的稳定性和性能。风险管理是成熟过程中项目管理的必不可少部分。在某些情况下,一个成熟过程意味着在软件生命周期的早期识别出“失败”项目,使得在徒劳无功的事情上的投资最小。

2.5 跳越成熟度等级

CMM 中的成熟度等级描述在一个成熟度等级上组织的特征。每一等级均为接连的几个等级奠定基础,为有效地且效率高地实施过程提供支持。可是,在有益时,组织也可以使用比它们所在等级高的那些成熟度等级中所描述的过程。例如,虽然 CMM 中在等级 3 以前不讨论工程过程——诸如需求分析、设计、编码和测试,但是甚至等级 1 组织都必须进行这些活动。在有利可图时,等级 1 或等级 2 组织可以进行同行评审(等级 3 的)、Pareto 分析(等级 4 的)、或者引入新技术(等级 5 的)。在讨论为了从等级 1 提高到等级 2 一个组织应采取何种步骤时,一个经常性的建议是建立软件工程过程小组,而这是等级 3 组织的属性。虽然度量是等级 4 的关注焦点,但它也是较低成熟度等级的必备部分。

可是这些过程的潜力只有在建立适当基础之后才能得到完全的发挥。例如,同行评审不可能完全有效,除非始终一致地实施这种评审,即使对非常紧急的项目也是如此。成熟度等级只描述一个等级上占主导地位的问题。等级 1 组织的占主导地位的问题是管理;策划和管理软件项目的困难掩盖了其它问题。

因为每个等级形成一个必要的基础,从此基础出发才能达到下一个等级,因此,跳越等级是违反生产规律的。CMM 识别几个等级,一个组织必须逐步经历这些等级才能建立优秀的软件工程文化。没有合适基础的过程,在正是最需要它们的时刻——即处于压力下——不起——作用,它们也不提供未来改进的基础。

等级 1 组织,在尚未建立可重复过程(等级 2)之前,试图去实施已定义的过程,通常不会成功,因为项目经理会被进度和成本的压力压垮。这是在关注工程过程之前应首先集中注意力于管理过程的基本原因。看起来定义和实施一个工程过程似乎要比定义和实施管理过程容易(特别在技术人员眼中),但是如果没有管理规定,工程过程会成为进度和成本等压力的牺牲品 [Humphrey 88]。

一个尚无已定义过程作为基础就试图实施已管理过程(等级 4)的组织通常是不成功的,因为没有已定义的过程就没有解释度量的共同基础。虽然对于一个个项目能采集数据,但几乎没有什么度量对本项目之外的其它项目有重大意义,也不能显著地增加组织对软件过程的理解。缺少已定义过程时,由于被测过程的变化,要鉴别出有质量的质量是困难的。

一个尚无已管理过程(等级 4)作为基础就试图实施优化过程(等级 5)的组织,由于缺乏对过程更改所产生的后果的了解,多半会失败。在不能控制过程使它处于统计意义上狭

窄的范围内（即过程度量中仅有小的变化）的情况下，数据中有太多的噪声以致不能客观地确定某项具体的过程改进是否有效。因为几乎没有定量基础用于作出理性的、有信息依据的决策，决策可能退化为宗教式的争斗。

过程改进工作应该把焦点集中在经营环境的上下文中组织的需要。具有实施较高成熟度等级的过程的能力并不表示可以跳越成熟度等级。

3 CMM 的可操作定义

CMM 是一个框架，描述推荐给试图提高其软件过程能力的组织的一条改进路径。为了支持使用 CMM 的多种方法，特设计了 CMM 的操作细节。至少有四种 CMM 的用法受到支持：

- ✍✍ 评估组运用 CMM 去识别组织中的强项和弱项。
- ✍✍ 评价组运用 CMM 去确定选择承包商授予经营权的风险，并运用 CMM 监控合同。
- ✍✍ 经理和技术人员运用 CMM 去理解那些对于规划和实施其软件过程改进大纲来说必不可少的活动。
- ✍✍ 过程改进组，例如 SEPG，运用 CMM 作为指南，帮助他们定义和改进其组织的软件过程。

因为 CMM 有多种用法，所以必须将它分解得足够细，使得能从成熟度等级的结构中导出有关实际过程的建议。这种分解也指出那些能对软件过程成熟度和软件过程能力进行特征描述的过程及其结构。

3.1 成熟度等级的内部结构

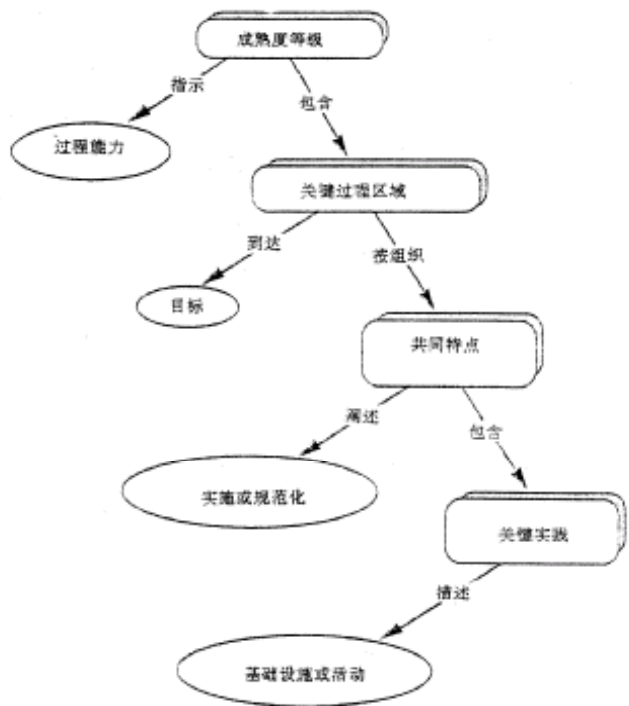


图 3.1 CMM 结构

已将每个成熟度等级分解为多个组成部分。除第一级外，每个成熟度等级的分解均从每个等级的抽象概要向下直至它们在关键实践方面的可操作定义，如图 3.1 所示。每个成熟度等级由几个关键过程区域组成。每个关键过程区域又按五个称为共同特点 的部分加以组织。

共同特点规定关键实践，当这些关键实践均得到实施时，就能实现关键过程区域的目标。

3.2 成熟度等级

一个成熟度等级是一个妥善定义的、朝着实现成熟软件过程目标的进化途中的平台。正如图 2.1 所示，每个成熟度等级指示过程能力的一个等级。例如，在等级 2 上，通过建立健全的项目管理控制，组织的过程能力已经从无序的提高到有纪律的。

3.3 关键过程区域

除等级 1 外，每个成熟度等级被分解成几个关键过程区域，指明为了改进其软件过程组织应关注的区域。关键过程区域识别出为了达到某个成熟度等级所必须着手解决的问题。

每个关键过程区域识别出一单相关活动，当这些活动全部完成时，能达到一组对增强过程能力至关重要的目标。如图 3.2 所示，每个关键过程区域按定义存在在单个成熟度等级

(译注)：共同特点是指所有关键过程区域均有的特点，故称为共同特点。

上。达到关键过程区域目标的途径可能因项目而异，这是因为在应用领域或环境上有差异。不过，为了使得组织实现某个关键过程区域，必须达到该关键过程区域的全部目标。当在连续的基础上，对所有项目均已达到一个关键过程区域的目标时，可以说该组织已使以此关键过程区域为特征的过程能力规范化了。

形容词“关键”的使用蕴含着存在对于实现一个成熟度等级来说并不关键的过程区域（和若干过程）。CMM 并不详细描述所有与开发和维护软件有关的过程区域。已经鉴别出某些过程为过程能力的关键的决定因素；CMM 中描述的就是这些因素。

尽管其它问题也影响过程性能，但我们只鉴别出关键过程区域，这是因为它们在改进组织软件过程能力上最有效。可以认为它们是达到一成熟度等级的必要条件。图 3.2 显示每一成熟度等级的关键过程区域。为了达到一个成熟度等级，必须实现该等级上的全部关键过程区域。为了实现一个关键过程区域必须达到该关键区域的每一个目标概括一个关键过程区域的关键实践，可用来确定是否一个组织或一个项目已有效地实现该关键过程区域。目标表明每个关键过程区域的范围、边界和意图。

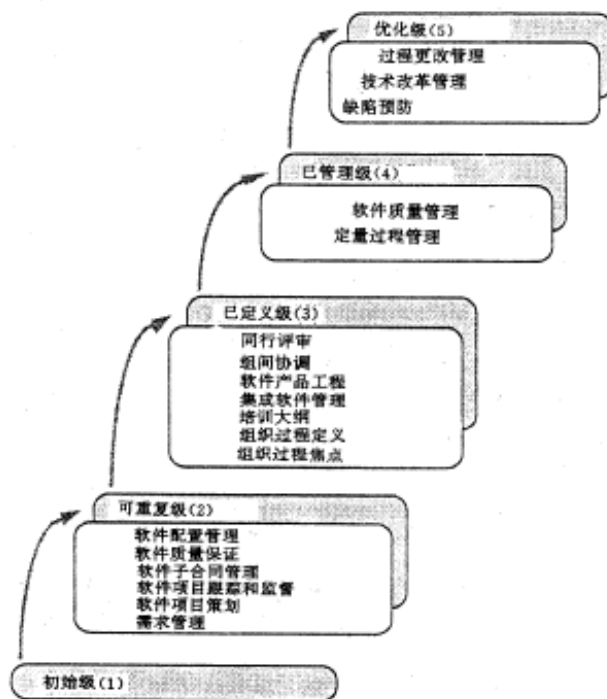


图 3.2 按成熟度等级排列的关键过程区域

随着组织晋升到过程成熟度的更高等级，在每个关键过程区域上应进行的具体实践在内容上将有所发展。例如，等级 2 上软件项目策划关键过程区域所描述的项目估计能力中的许多项必须进化以便能处理在等级 3、4、5 上可得到的、附加的项目数据。当采用已定义软件过程来管理项目时，等级 2 的软件项目策划及软件项目跟踪和监督进化为等级 3 上的集成软件管理。

CMM 的关键过程区域表示一种描述组织如何成熟的方法。确定这些关键过程区域是基于多年来在软件工程和软件管理方面的经验和五年多在软件过程评估和软件能力评价方面的经验。

等级 2 上的关键过程区域集中关注软件项目所关心的、与建立基本项目管理控制有关的事情。下面列出对等级 2 上每个关键过程区域的描述：

需求管理目的是在顾客和软件项目之间建立对顾客需求的共同理解，顾客需求将由

软件项目处理。与顾客的协议是策划（正如在软件项目策划中所描述的）和管理（正如在软件项目跟踪和监督中所描述的）软件项目的基础。对与顾客关系的控制依靠遵循有效的更改控制过程（正如在配置管理中所描述的）。

✂✂ 软件项目策划的目的是制定进行软件工程和管理软件项目的合理的计划。这些计划是管理软件项目的必要基础（正如在软件项目跟踪和监督中所描述的）。没有切合实际的计划不可能实施有效的项目管理。

✂✂ 软件项目跟踪和监督的目的是建立适当的对实际进展的可视性,使管理者在软件项目性能显著偏离软件计划时能采取有效的措施。

✂✂ 软件子合同管理的目的是选择合格的软件子承包商,并有效地管理它们。它把用于基本管理控制的需求管理、软件项目策划、以及软件项目跟踪和监督等关键过程区域所关注的事情与软件质量保证和软件配置管理等过程区域中的必不可少的协调结合在一起,并且当合适时对于承包商实施这项管理。

✂✂ 软件质量保证的目的是给管理者提供对于软件项目正采用的过程和正在构造的产品的恰当的可视性。软件质量保证是绝大多数软件工程过程和管理过程的不可缺少的部分。

✂✂ 软件配置管理的目的是在项目的整个软件生存周期中建立和维护软件产品的完整性。软件配置管理是绝大多数软件工程过程和管理过程的不可缺少的部分。

等级 3 的关键过程区域既阐述项目的问题,又阐述组织的问题,这是因为组织建立起便对所有项目有效的软件工程过程和管理过程规范化的基础设施。下面列出对等级 3 上每个关键过程区域的描述:

✂✂ 组织过程焦点的目的是规定组织在改进其整体软件过程能力的软件过程活动方面的职责。组织过程焦点活动的主要结果是一组软件过程财富(它们在组织过程定义中加以描述。正如集成软件管理中所描述的,这些财富供软件项目使用。

✂✂ 组织过程定义的目的是开发和保持一组便于使用的软件过程财富,它们能改进横跨项目的过程性能,并且为组织能获得积累性的、长期得益奠定基础、这些财富提供一组稳定的基本原则,通过诸如培训等机制就能使其成为制度。培训在培训大纲中加以描述。

✂✂ 培训大纲的目的是培育个人的技能和知识,使得他们能有效地和效率高地执行其任务。尽管培训是组织的责任,但是软件项目应该识别出他们所需要的技能、当项目需求独特时,该项目应提供所需要的培训。

✂✂ 集成软件管理的目的是将软件工程活动和管理活动集成为一个协调的、已定义的软件过程,该过程是剪裁组织的标准软件过程和组织过程定义中所描述的相关的过程财富而得到的。剪裁基于项目的经营环境和技术需要,正如在软件产品工程中所描述的那样。集成软件管理是从等级 2 的软件项目策划和软件项目跟踪和监督进化而得到的。

✂✂ 软件产品工程的目的是—致地执行一个妥善定义的工程过程,为了能有效地和效率高地生产正确的、—致的软件产品,该工程过程集成全部软件工程活动。软件产品工程描述项目的技术活动,例如,需求分析、设计、编码和测试。

✂✂ 组间协调的目的是为软件工程组积极参与其它工程组工作制定一种方法,使得项目更能有效地和效率高地满足顾客的需求。组间协调是集成软件管理的一个涉及多学科方面,它延伸到软件工程之外;不仅应该集成软件过程而且软件工程组和其它组之间的相互作用也必须加以协调和控制。

✂✂ 同行评审的目的是及早和高效地除去软件工作产品中的缺陷、一个重要的必然结果是增强对软件工作产品和可预防的缺陷的了解、同行评审是一种重要而又有效的工

程方法，在软件产品工程中调用此方法，可通过法根式审查（Fagan-style 审查）[Fagan86]、结构化走查、或者一些其它的学院式的评审方法下[Freedman 90]加以实施。

等级 4 上的关键过程区域的关注焦点是建立起对软件过程和正在构造的软件工作产品的定量了解。正如下所述，该等级上的两个关键过程区域——定量过程管理和软件质量管理——是互相紧密依赖的：

- ✍ 定量过程管理的目的是定量地控制软件项目的过程性能。软件过程性能表示遵循一个软件过程所得到的实际结果。焦点是在一个可测的稳定的过程范围内鉴别出变化的特殊原因，并且适当时改正那些促使瞬时变化出现的环境。定量过程管理给组织过程定义、集成软件管理、组间协调、和同行评审的实践附加一个内容丰富的测量计划。
- ✍ 软件质量管理的目的是建立对项目的软件产品质量的定量了解和实现特定的质量目标。软件质量管理对软件产品工程中所描述的软件工作产品实施内容丰富的测量计划。

等级 5 上的关键过程区域包括那些为了实施连续不断的和可测的软件过程改进。组织和项目都必须解决的问题。下面列出等级 5 的每个关键过程区域的描述。

- ✍ 缺陷预防的目的是鉴别缺陷的原因并防止它们再次出现。正如在集成软件管理中所描述的，软件项目分析缺陷、鉴别其原因并更改项目定义软件过程。正如在过程复政管理中所描述的，应将具有普遍价值的过程更改通知给其它软件项目。
- ✍ 技术改革管理的目的是识别出能获利的新技术（即工具、方法和过程），并以有序的方式将它引进到组织中去，正如在过程更改管理中所描述的那样。技术改革管理的关注焦点是在不断变化的环境里高效率地进行创新。
- ✍ 过程更改管理的目的是出于改进软件质量、提高生产率和缩短产品开发周期的目的持续不断地改进组织中所采用的软件过程。过程更改管理既采用缺陷预防的增量式改进，又采用技术改革管理的创新式改进，并使得整个组织可以享用这些改进。

3.4 共同特点

为方便起见，关键过程区域按共同特点加以组织。共同特点是表明一个关键过程区域的实施和规范化是否有效、可重复且持久的一些属性。五个共同特点列举如下：

执行约定 执行约定描述组织保证过程得以建立和继续起作用所必须采取的行动。执行约定一般包含制定组织的方针和规定高级管理者的支持。

执行能力 执行能力描述为了能实施软件过程，项目或组织中必须存在的先决条件。执行能力一般包括资源、组织结构和培训。

执行的活动 执行的活动描述为实现一个关键过程区域所必须的角色和规程。执行的活动一般包括制定计划和规程，进行工作，跟踪它，并在需要时采取纠正措施。

测量和分析 测量和分析描述对过程进行测量和对测量结果进行分析的需要。测量和分析一般包括一些为了确定所执行活动的状态和有效性所能采用的测量的例子。

验证实施 验证实施描述那些能保证遵照已建立的过程进行活动的措施。验证一般包括管理者和软件质量保证部门所作的评审和审计。

在执行的活动这一共同特点中的实践描述为了建立过程能力必须作些什么。而其它的实践作为一个整体形成了使组织能将执行的活动中所描述的实践规范化的基础。

3.5 关键实践

每个关键过程区域用一些能对实现其目标作贡献的关键实践加以描述。关键实践描述对关键过程区域的有效实施和规范化贡献最大的基础设施和活动。

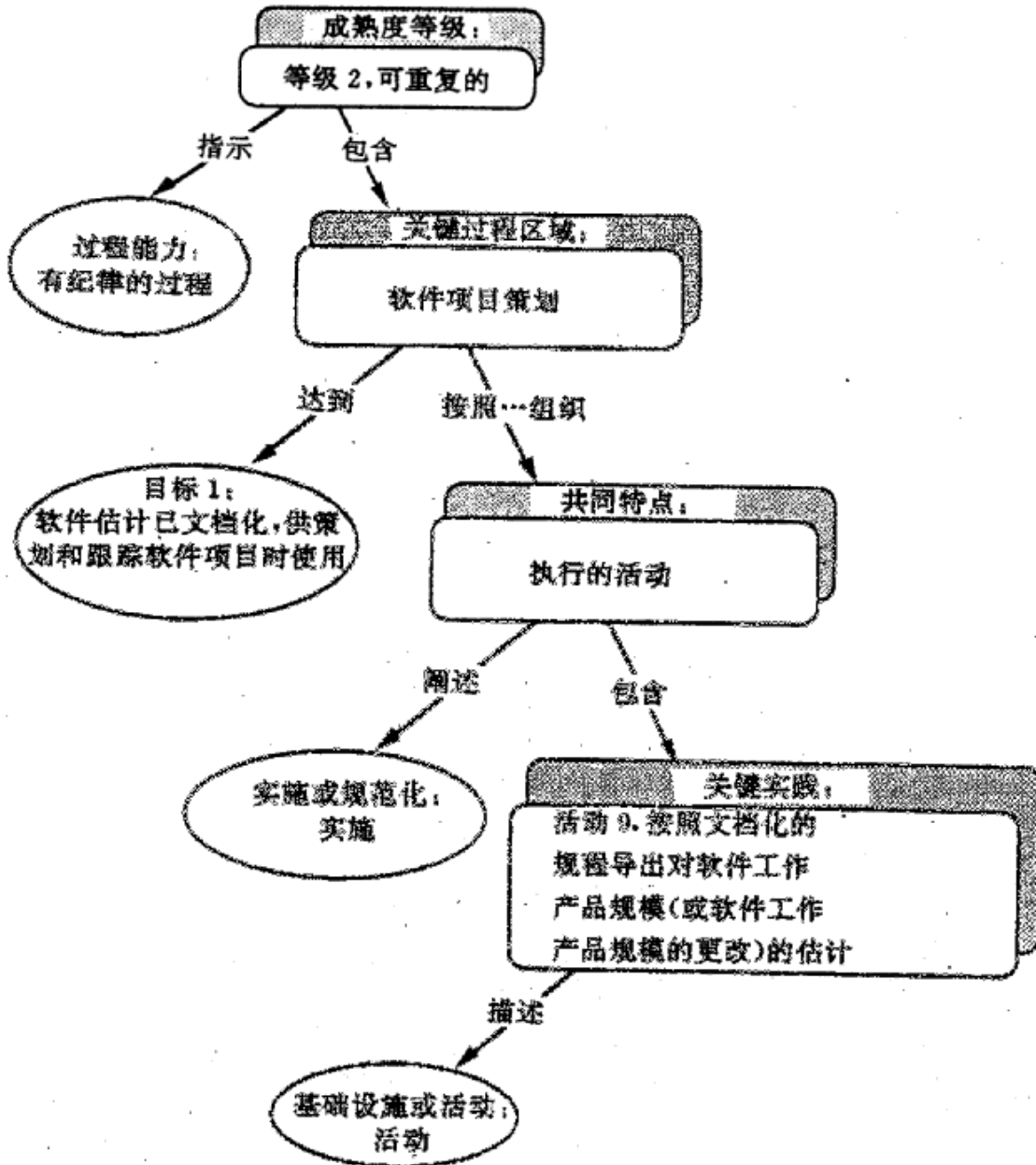


图 3.3 构造 CMM 的结构：一个关键实践的例子

每个关键实践的组成是：一个单个句子，常常紧跟着较为详细的描述，后者可能包括例子和详细细节。前者，又称为顶层关键实践，说明关键过程区域的基本方针、规程和活动。进行详细描述的分量常称作实践。图 3.3 给出软件项目策划关键过程区域中一个关键实践的内部结构的例子。

正如图 3.3 所示，为了保证一致地实现提供用于策划和跟踪项目的计划文件这一目标，

组织必须建立一个文档化的规程用以导出对软件规模的估计。由于从来不明确阐述测定规模时假定上的许多差别，如果不是用文档化的规程导出这些估计，那么估计值可以在很大范围内变化。在这样的规程中所期待的详细描述包括使用以前的规模数据、对假定提供文件说明、和对信号进行评审。这些准则对是否遵循一个合理的规模估计规程作出判断提供指导。

关键实践描述应做“什么”，而不应解释为强制要求应该“如何”实现目标。替代的实践也可能实现该关键过程区域的目标。应该合理地解释关键实践以便判断关键过程区域的目标是否已被有效地实现，尽管实现目标的方式可能不同。“能力成熟度模型的关键实践 1.1 版”〔Pank 93b〕中收纳了关键实践，以及有关他们的解释的指南。

4 运用 CMM

CMM 建立一组公用、有效的描述成熟软件组织特征的准则。组织能运用这些准则去改进其开发和维护软件的过程，政府或商业组织能用它们去评价与一特定公司签订软件项目合同时的风险。

本章集中精力于两个 SEI 研制的用以评估组织实施软件过程的成熟度的方法：软件过程评估和软件能力评价。

✍ 软件过程评估 用于确定一个组织的当前软件过程的状态，确定组织所面临的具有高优先级的与软件过程有关的问题，和获得组织对软件过程改进的支持。

✍ 软件能力评价 用于识别合格的能完成软件工作的承包商或者监控现有软件工作中所运用的软件过程的状态。

仅本概述本身文才于读者进行评估或评价均是不充分的。任何希望通过这两个方法去运用 CMM 的人应该请求得到有关评估和评价培训的进一步的信息。

CMM 是软件过程评估和软件能力评价的共同基础。可是这两种方法的目的十分不同，因此在所用的具体方法上存在重大差异。但是两者又都基于 CMM 模型及从它导出的产品。

CMM 和基于 CMM 的产品一起，使得可靠地、一致地使用这些方法成为可能。

4.1 软件过程评估和软件能力评价方法

软件过程评估集中关注在组织自身的软件过程中识别出改进的优先组。评估组采用 CMM 去指导他们对调查发现进行鉴别和优先级排序。这些调查发现与 CMM 中的关键实践所提供的指导一起被（例如，软件工程过程小组）用来规划组织的改进策略。

软件能力评价集中关注识别与一项在进度要求和预算内构造出高质量软件的特定项目或合同相联系的风险。在采购过程中可以对投标者进行软件能力评价。评价的发现，正如 CMM 所设计的那样，可以用于确定在挑选特定承包商方面的风险，也可对现有的合同进行评价以便监控它们的过程性能，其目的是在承包商的软件过程中识别出潜在的可改进之处。

CMM 为进行软件过程评估和软件能力评价建立一个共用的参考框架。虽然此两个方法的目的不同，但均采用 CMM 作为评估软件过程成熟度的根据。图 4.1 概要描述评估和评价中的共同步骤。

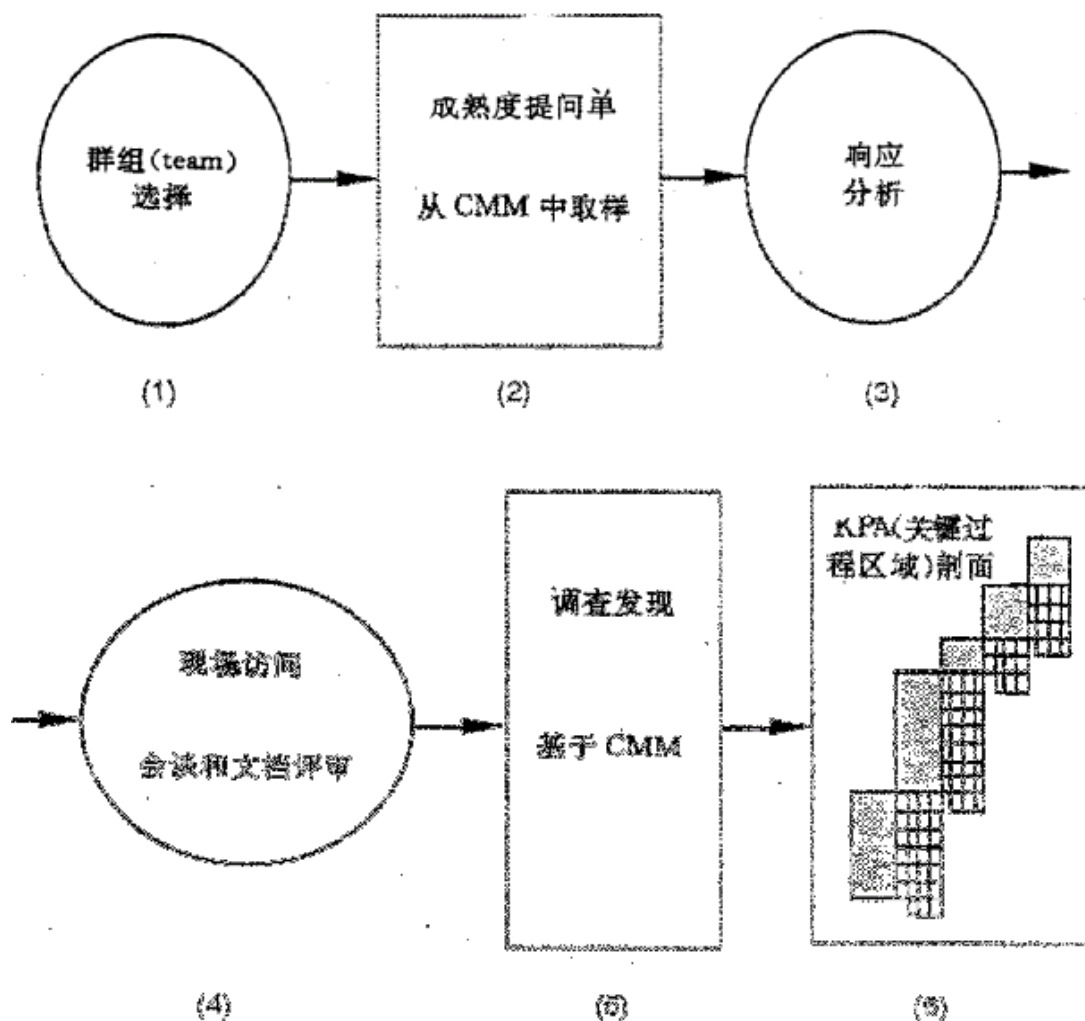


图 4.1 软件过程评估和软件能力评价的共同步骤

其第一步是选择一个群组。该群组应该接受在 CMM 基本概念和评估或评价方法的细节方面的培训。该群组的成员应是具有丰富软件工程和管理方面知识的专业人员。

第二步是让来自待评估或评价单位的代表完成成熟度提问单的填写和其它诊断工具的要求。一旦完成此项活动，评估或评价组就对提问单响应进行分析（第三步），即对提问响应进行统计，并识别出必须作进一步探查的区域。待探查的区域与 CMM 的关键过程区域相对应。

现在该群组已准备好去访问被评估或评价单位的现场（第四步）。从响应分析的结果着手，该群组进行会谈和评审文档以便了解该现场所遵循的软件过程。CMM 中的关键过程区域和关键实践对该群组在提问、倾听、评审和综合得自会见和文档中的信息等方面提供指导。在确定现场的关键过程区域的实施是否满足相关的关键区域的目标方面，该群组运用专业性的判断。当 CMM 的关键实践与现场的实践间存有明显差异时，该群组必须用文件记下用于判断此关键过程区域的理论依据。

在现场工作阶段结束时，该群组生成一个调查发现清单（第五步），标识出该组织软件

当涉及到公司专利或保密问题时，可能不得不在缺乏完全信息的情况下，作出这些判断。

过程的强项和弱项。在软件过程评价中，该调查发现成为提出过程改进建议的基础；在软件能力评价中调查发现成为采购单位所作的风险分析的一部分。

最后，该群组作出一份关键过程区域剖面图（第六步），指出该组织已满足和尚未满足关键过程区域目标的那些区域。一个关键过程区域可能是已满足的，但仍有一些相关的调查发现，假定这些调查发现没有识别出能阻止实现该关键过程区域的任何一个目标的主要问题。

总之，软件过程评估和软件能力评价方法两者均：

- ✍ 采用成熟度提问单作为现场访问的出发点；
- ✍ 采用 CMM 作为指导现场调查研究的路线图；
- ✍ 利用 CMM 中的关键过程区域生成标识软件过程强项和弱项的调查发现；
- ✍ 在对关键过程区域目标满足情况进行分析的基础上，推导出一个剖面；
- ✍ 根据调查发现和关键过程区域剖面，向合适的对象提出结论意见。

4.2 软件过程评估和软件能力评价之间的差异

尽管有这些相似之处，但软件过程评估或软件能力评价的结果可能不同，甚至在相继运用相同的方法接连进行评估（或评价）的情况下也是如此。

一个原因是评估或评价的范围可能变化。首先，必须确定受调查的组织情况。对一个大公司而言，对“组织”有几种不同的定义是完全可能的。范围可以基于有共同的高级管理者、共同的地理位置、指明为一个盈亏中心、共同的应用领域或者其它考虑。其次，甚至在似乎是相同的组织中，所选项目的样本也可能影响范围。评估可能对公司中的一个部门进行，那么评估组基于全部部门的范围得出其调查发现。后来，可能对该部门中的一条产品线进行评价，此时评价组所得出的调查发现基于窄得多的范围。没有了解上下文就在这些结果间作比较是成问题的。

软件过程评估和软件能力评价在动机、目的、输出和结果的所有权上均不同。这些因素导致在会谈目的、询问的范围、所采集的信息和结果的表示方式上有本质的不同。如果考察所采用的详细规模，那么评估和评价的方法大不相同。评估培训并不使得该组作好完成评价工作的准备，反之亦然。

软件过程评估都是在开放、合作的环境中进行的，评估的成功取决于管理者和专业人员两方面对改进组织的支持。评估目的在于暴露问题和帮助经理和工程师改进他们的组织。尽管提问单在使评估组的注意力集中于成熟度等级问题上是有价值的，但是重点仍应放在作为了解组织软件过程的工具的结构化和非结构化的会谈上。除了识别组织所面临的软件过程问题外，评估的最有价值的结果还有为改进而作的买进，全组织范围关注过程，以及执行改进行动计划上的动力和热情。

另一方面，软件能力评价在更为面向审计的环境中进行。评价目的与金钱密切相关，因为评价群组的推荐将帮助挑选承包商或设置奖金。重点放在巴文档化的审计记录上，它们能揭示组织实际执行的软件过程。

可是这并不意味着软件过程评估和软件能力评价一定不能比较。由于两者均基于 CMM，类似点和不同点应该是明显的且可以解释的。

4.3 CMM 在过程改进方面的其它用法

对软件工程过程组（SEPG）或试图改进其软件过程的其它组来说，CMM 在改进措施

的策划上，措施计划的实施上、和过程定义上有着特殊的价值。在策划改进措施期间，具有有关其软件过程问题和经营环境的知识的软件工程过程组的成员，可将 CMM 中关键过程区域的目标与其当前的实践相比较。应该考察与公司目标、管理优先级、实践运行的层次、实施每次实践对组织的价值、以及该组织在其文化背景下实施一个实践的能力等方面有关的关键实践。

接下来软件工程过程组必须确定需要作哪些过程改进，如何实现更改、以及如何获得所需要的买进。CMM 通过给有关过程改进的讨论提供讨论的出发点，并且帮助揭示与通用软件工程实践所采用的那些假定完全不同的假定，从而对这项活动提供帮助。在实施行动计划时，过程组可用 CMM 和关键实践来构造部分可操作的行动计划和定义软件过程。

5 CMM 的未来发展方向

实现软件过程成熟度的较高等级是增量式进化过程，要求对不断改进过程有长期的支持。软件组织可能用十年或更长的时间去建立持续过程改进的基础和文化。虽然对绝大多数美国公司来说，长达十年的过程改进计划是陌生的，但是为了成为成熟软件组织，这种层次的努力是必须的。这个时间框架与其它工业部门（例如美国的汽车制造业）的经验相一致，它们在过程成熟度上已取得重大的进展 [Gabor 90]。

5.1 CMM 不包括什么

CMM 不是无所不包的 [Brooks 87]，它并不阐述所有的对成功项目来说是重要的问题。例如，当前 CMM 并未谈及特定应用领域内的专门知识、鼓吹具体的软件技术、或者就如何选择、雇用、激励和留住有能力的人，提出建议。虽然这些问题对项目的成功至关重要，其中一些已在其它上下文中，进行了分析 [Curtis 90]，可是尚未将这些问题综合到 CMM 中。CMM 是为了提供一个有秩序的、有纪律的框架，以便在此框架内阐述软件管理过程和工程过程的问题而专门研制的。

5.2 近期活动

在遍及美国的主要会议和讨论班上经常举办短训班及讲课，以保证软件产业界对 CMM 及其相关联的产品有足够的了解。正在研制基于 CMM 的工具（例如，成熟度提问单），软件过程评估培训和软件能力评价培训，和（或）正在修改它们以便纳入到 CMM 中去。

CMM 开发活动的近期关注焦点是那些经剪裁的 CMM 版本，例如适用于小项目和（或）小规模组织的 CMM。CMM V1.1 是用适宜于大的政府签约组织的标准实践表达的，这些实践必须加以剪裁才能适合不同于这个样板组织的需要。

5.3 长期活动

在以后的几年内，通过在软件过程评估和软件能力评价中不断使用 CMM，CMM 将继续经受广泛的检验。适当时，将研制和修定基于 CMM 的产品和培训资料。CMM 是一个将得到改进的、正在使用的文件，但是预期至少直到 1996 年 CMM V1.1 将仍然是基线。这样就在稳定性和连续不断改进的需要之间提供合适的和切合实际的平衡。

对于 CMM 的下一个版本，版本 2.0，SEI 将其注意力转向改进整个模型。虽然模型的所有等级都可能被修改，但重点放在等级 4 和等级 5 上。当前已非常完全地定义了等级 2 和等级 3 的关键过程区域。由于几乎没有什么组织经评估在等级 4 或等级 5 上 [Humphrey 91a, Kitson 92]，对这类组织的特征知之甚少。SEI 正与力求了解和达到等级 4 和等级 5 的组织密切合作，将对这两个等级的实践进行修订。CMM 可能成为多维的，以便阐述技术和人类资源的问题。

SEI 还正在和国际标准化组织（ISO）一起工作，制定软件过程评估、改进和能力评价的国际标准。SEI 的有关过程方面的工作将影响 ISO 的活动，与此同时，ISO 标准的制定也将影响 CMM 的 2.0 版本。

5.4 结论

正如连续改进运用到软件过程一样，连续改进也运用到成熟度模型和实践。虽然，应该认真考虑 CMM 的更改对软件界的潜在影响，但是 CMM、成熟度提问单、和软件过程评估及软件能力评价等方法将随着有关软件过程改进方面经验的不断积累而不断地进化。SEI 打算在进行此项进化工作上继续与工业部门、政府和学术界紧密合作。

CMM 提供了一种以有纪律的和一致的方式改进软件产品的管理和开发的概念结构。但是它并不能保证软件产品将成功地构造出来，或者保证恰当地解决全部软件工程中的问题。虽然 CMM 能标识一个成熟软件过程的实践和提供某些当前最高实践水平的（并且在某种情况下，是当前最高技术水平）的例子，但是这并不意味着它是详尽无漏的或是可以强制执行的。CMM 只标识出有效软件过程的特征，但是成熟组织必须致力于对成功项目来说是必不可少全部问题，包括人、技术和过程。

6.参考文献

- | | |
|-------------|---|
| Brooks87 | F.P.Brooks "No Silver Bullet: Essence and Accidents of Software Engineering" IEEE Computer, Vol.20, No.4, April 1987, pp.10-19 |
| Crosby79 | P.B.Crosby, Quality is Free, McGraw-Hill, New York, NY, 1979. |
| Curtis90 | B.Curtis, "Managing the Real Leverage in software Productivity and Quality," American Programmer, Vol.3, NO.7, July 1990, PP. 4 - 14. |
| Deming86 | W. Edwards Deming, Out of the Crisis, MIT Center for Advanced Engineering Study, Cambridge, MA, 1986. |
| DoD87 | Report of the Defense Science Board Task Force on Military Software, Office of the Under Secretary of Defense for Acquisition, Washington, D.C, September 1987. |
| Fagan86 | M. E. Fagan, 'Advances in software Inspections,' IEEE Transactions on Software Engineering, Vol. 12, No. 7, July, 1986, PP. 744-751. |
| Fowler90 | P. Fowler and S. Rifkin, Software Engineering Process Group Guide, Software Engineering Institute, CMU / SEI-90-TR-24, ADA235784, September, 1990. |
| Freedman 90 | D.P.Freedman and G.M.Welnsberg, Handbook of Walkthroughs, Inspections, and Technical Reviews, Third Edition, Dorset House, New York, NY, 1990. |
| Gabor90 | A.Gabor, The Man who Discovered Quality, Random House, New York, NY, |

- 1990
- GAO-92-48 Embedded Computer Systems : significant software Problem'On C- 17 Must Be Addressed , GAO / IMTEC-92-48 , May 1992 .
- Humphrey87a W . S . Humphrey , Characterizing the Software Process : A Maturity Framework , Software Engineering Institute , CMU / SEI - 87 - TR - 11 , ADA182895 , June 1987 .
- Humphrey87b W . S . Humphrey and W . L . Sweet , A Method for Assessing the Software Engineering Capability of Contractors , Software Engineering Institute , CMU / SEI - 87 - TR - 23 , ADA187320 , September 1987 .
- Humphrey88 W . S . Humphrey , characterizing the Software Process , IEEE Software , Vol . 5 , No . 2 , March , 1988 , pp . 73 - 79 .
- Humphrey89 W . S . Humphrey , Managing the Software Process , Addison - Wesley , Reading , MA , 1989 .
- Humphrey91a W . S . Humphrey , D . H . Kitson , and J . Gale , " A Comparison of U.S . and Japanese Software Process Maturity , " Proceedings of the 13th International Conference on Software Engineering Austin , TX , 13-17 May 1991 , PP38-49 .
- Humphrey91b W . S . Humphrey , ' Process Fitness and Fidelity , " Proceedings of the Seventh International Software Process Workshop , 16 - 18 October 1991
- IEEE-STD-610 ANSI / IEEE Std 610- 12 - 1990 , ' ' IEEE Standard Glossary of Software Engineering Terminology , ' February 1991 .
- Imai86 M . Imai , Kaizen :The Key to Japan 's Competitive Success , McGraw-Hill , New York , NY , 1986
- Juran88 J.M.Juran , Juran on Planning for quality, Macmillan , New York, NY , 1988 .
- Juran89 M.Juran Juran on Leadership for Quality,The Free Press,New York, NY , 1989 .
- Kitson92 D . H . Kitson and S . Masters , An Analysis of SEI Software Process Assessment Results : 1987-1991 . Software Engineering Institute , CMU / SEI - 92 - TR - 24 , July 1992 .
- Paulk91 M . C . Paulk , B . Curtis , M . B . Chrissis , et al , Capability Maturity Model for Software , Software Engineering Institute , CMU / SEI-91-TD-24 , ADA240603 , August 1991 .
- Paulk93a M . C . Paulk , B . Curtis , M . B . Chrissis , and C . V . Weber , Capability Maturity Model for Software , Version 1. 1 , Software Engineering Institute , CMU / SEI93-TR-24 , February 1993 .
- Paulk93b M . C . Paulk , C . V . Weber , S . Garcia , M . B . Chrissis , and M . Bush , Key Practices of the Capability Maturity Model , Version 1. 1 , Software Engineering Institute , CMU/SEI - 93-TR-25 February 1993.
- Radice85 R . A . Radice , J . T . Harding , P . E . Munnis , and R . W . Phillips , ' A Programming Process Study , ' IBM systems ; journal , Vol . 24 , no . 2 , 1985 .
- Siegel90 J.A.I.Siegel , et al , National software Capacity : Near-Term Study , Software Engineering Institute CMU / SEI - 90 - TR - 12 , ADA226694 , May 1990.
- Weber91 V . Weber , M . C . Paulk , C . J . Wise , and J . V . Withey , Key Practices of the Capability Maturity model , Software Engineering Institute , CMU / SEI - 91-TR-25 , ADA244644 August 1991 .

第二篇 能力成熟度模型的关键实践 1.1 版

1 介绍关键实践文件

1.1 致读者

对许多软件组织来说，要在预算内按时开发出可靠和可用的软件是一项困难的工作。软件产品延期交付、或超出预算、或不能如预期的那样工作也给软件组织的顾客带来麻烦。随着项目规模和重要性不断增长，这些难题日益严重。但是通过在建立有效软件工程实践和管理实践的过程基础设施方面集中而又持续不断的努力，这些困难能得以克服。

为了构造这个过程基础设施，生产软件的组织需要有办法评价他们成功地执行其软件过程的能力。他们也需要有指南帮助他们去改进其过程能力。诸如国防部（DOD）这样的顾客需要有方法更有效地评价一个组织成功地执行软件工程合同的能力。主承包商需要有手段去评价潜在子承包商的能力。

为了帮助软件组织、像 DoD 那样的顾客、以及主承包商，软件工程研究所（SEI）已经开发出软件能力成熟度模型（CMM），它描述一个成熟的、有能力的软件过程的特征。此模型还按成熟度等级描述了从一个不成熟、不可重复的软件过程到成熟的、已妥善管理的软件过程的进展状态。

CMM 可用于：

- ✍ 软件过程改进，为此组织策划、设计和实施对其软件过程的更改；
- ✍ 软件过程评估，为此一个经培训的软件专业人员组确定组织当前软件过程的状态、确定组织所面临的具有高优先级的软件过程一有关问题，并获得组织上对软件过程改进的支持；
- ✍ 软件能力评价，为此一个经培训的软件专业人员组鉴别合格的能完成软件工作的承包商或监控现有软件工作中所用软件过程的状态。

本文件描述与 CMM 中每个成熟度等级相对应的关键实践。它详尽阐述在 CMM 的每一等级上成熟度的含义，并且是一份可用于软件过程改进、软件过程评估、和软件能力评价的指南。

CMM 的关键实践是用那些进行大的、政府合同工作的组织应该具有的标准实践来描述的。因此，在任何一种运用 CMM 的上下文中，必须采用有关如何运用这些实践的合理解释。

本文件的第四章中包含有关解释 CMM 的指南。当组织的经营环境明显不同于大的签约组织的经营环境时，必须恰当地解释 CMM。必须承认在有根据地使用 CMM 方面专业判断的作用。

本文件可有以下几种用途：

- ✍ 任何人用来了解开发和维护软件的有效过程中的关键实践，

- ✍✍ 任何人用来识别出为达到 CMM 下一个成熟度等级所必须的关键实践，
- ✍✍ 组织用来了解和改进其有效地开发软件的能力，
- ✍✍ 采购组织或主承包商用来了解让某个具体组织完成合同工作的风险，
- ✍✍ SEI 采用本文作为开发过程产品成熟度提问单的基础，
- ✍✍ 指导者用来指导评估组作执行软件过程评估或软件能力评价的准备。

1.2 本文件与其它文件的关系

提供 CMM 的初步基础的两个文档是：

- ✍✍ 文章 “characterizing the software process (刻画软件过程)” [Humphrey 88]，
 - ✍✍ 书 Managing the software Process (管理软件过程) (Humphrey 89)。
- 1991 年 8 月在以下两篇技术报告中公布了 CMM 的 1.0 版本：
- ✍✍ “Capability Maturity Model for Software (软件的能力成熟度模型)” (Paulk 91)，
 - ✍✍ “Key Practices of the Capability Maturity Model (能力成熟度模型的关键实践)” [Weber 91]。

在 1992 年对 CMM 的初始版本进行了修订。为了理解和使用 CMM 的当前版本，以下两文件是必不可少的：

- ✍✍ “Capability Maturity Model for Software, Version 1.1 (软件能力成熟度模型 1.1 版)” (Paulk 93)
- ✍✍ 本文件，“Key practices of the Capability Maturity Model, Version 1.1 (能力成熟度模型的关键实践 1.1 版)” (Paulk 93b)。

“软件能力成熟度模型 1.1 版”包含以下内容：模型的介绍、五个成熟度等级的描述、CMM 的可操作定义和它的结构、关于组织如何使用成熟度模型的讨论，和有关 CMM 未来发展方向的几点意见。

“能力成熟度模型的关键实践”包含与 CMM 的每个成熟度等级上的关键过程区域相对应的关键实践以及帮助解释关键实践的信息。

成熟度提问单和其它过程产品是从能力成熟度模型的关键实践导出的。支持软件过程改进、软件过程评估和软件能力评价的其它 SEI 过程产品包括培训课程、手册和现场调查指南。

1.3 本文的结构

概述的第一章给出 CMM 及本文的简介。后三章是：

- ✍✍ 概述 CMM 及其构成部分，
- ✍✍ 描述关键实践格式的用法，
- ✍✍ 描述运用和解释关键实践的方法。

在概述之后，描述 CMM 的关键过程区域的关键实践。为了那些想快速了解关键实践而无需运用关键实践时所必须的严格性的人的方便，附件 C 提供了关键实践的摘要。

附件中有：本文所引用的参考文献清单、本文所用的术语汇编、关键实践的摘要、本文的更改过程和本文术语的索引。

1.4 本文件预计的使用方法

如果你不熟悉 CMM，你应首先阅读“软件能力成熟度模型 1.1 版”CPaulk 93a3，并且在试图使用关键实践之前，阅读本文概述中的全部共四章内容。

如果你已经熟悉 CMM 及它的结构，你可能愿意直接进入第四章以便得到如何解释关键实践的建议。

2 能力成熟度模型概述

2.1 介绍能力成熟度模型

软件能力成熟度模型是一种描述有效软件过程的关键元素的框架。CMM 描述一条从无序的不成熟的过程到成熟的、有纪律的过程的进化的改进途径。

CMM 包括对软件开发和维护进行策划、工程化和管理的实践。遵循这些关键实践，就能改进组织在实现有关成本、进度、功能和产品质量等目标上的能力。

CMM 建立起一个标准，对照这个标准就能以可重复的方式判断组织软件过程的成熟度，并能将过程成熟度与工业的实践状态作比较 [Kitson 92]。组织也能采用 CMM 去规划它的软件过程改进。

2.2 CMM 的起源

软件工程研究所（SEI）应政府的要求在 MITRE 公司的帮助下开发出成熟度模型和成熟度提问单的一个初始版本。在模型和提问单的开发期间，SEI 始终重视参与开发和改进软件过程的实际工作人员的建议。我们的目标一直是提供这样一个模型，它：

- ✍✍ 基于现实的实践；
- ✍✍ 反映最好的实践状态；
- ✍✍ 反映从事软件过程改进、软件过程评估或软件能力评价的个人的需要；
- ✍✍ 是巴文档化的；
- ✍✍ 是公开能得到的。

自从成熟度模型的早期版本发行以来，人们已经获得了有关软件过程成熟度的补充知识和透彻理解。

这种理解是通过以下方式得到的：

- ✍✍ 研究非一软件组织；
- ✍✍ 执行和观察软件过程评估和软件能力评价；
- ✍✍ 征求和分析对模型的更改要求；
- ✍✍ 与工业界和政府的代表一起参加会议共同研讨；
- ✍✍ 征求来自工业部门和政府的评审人员的反馈意见。

利用这些补充知识，业已修订了能力成熟度模型及其实践，产生了 CMMI.1 版本。

2.3 CMM 的结构

CMM 由五个成熟度等级组成。除了等级 1 外，每个成熟度等级由几个关键过程区域组成。每个关键过程区域又划分为五个称作共同特点的部分。共同特点规定一些关键实践，当这些关键实践群体得到认真执行，能实现关键过程区域的目标。图 2.1 中显示了 CMM 的这个结构。

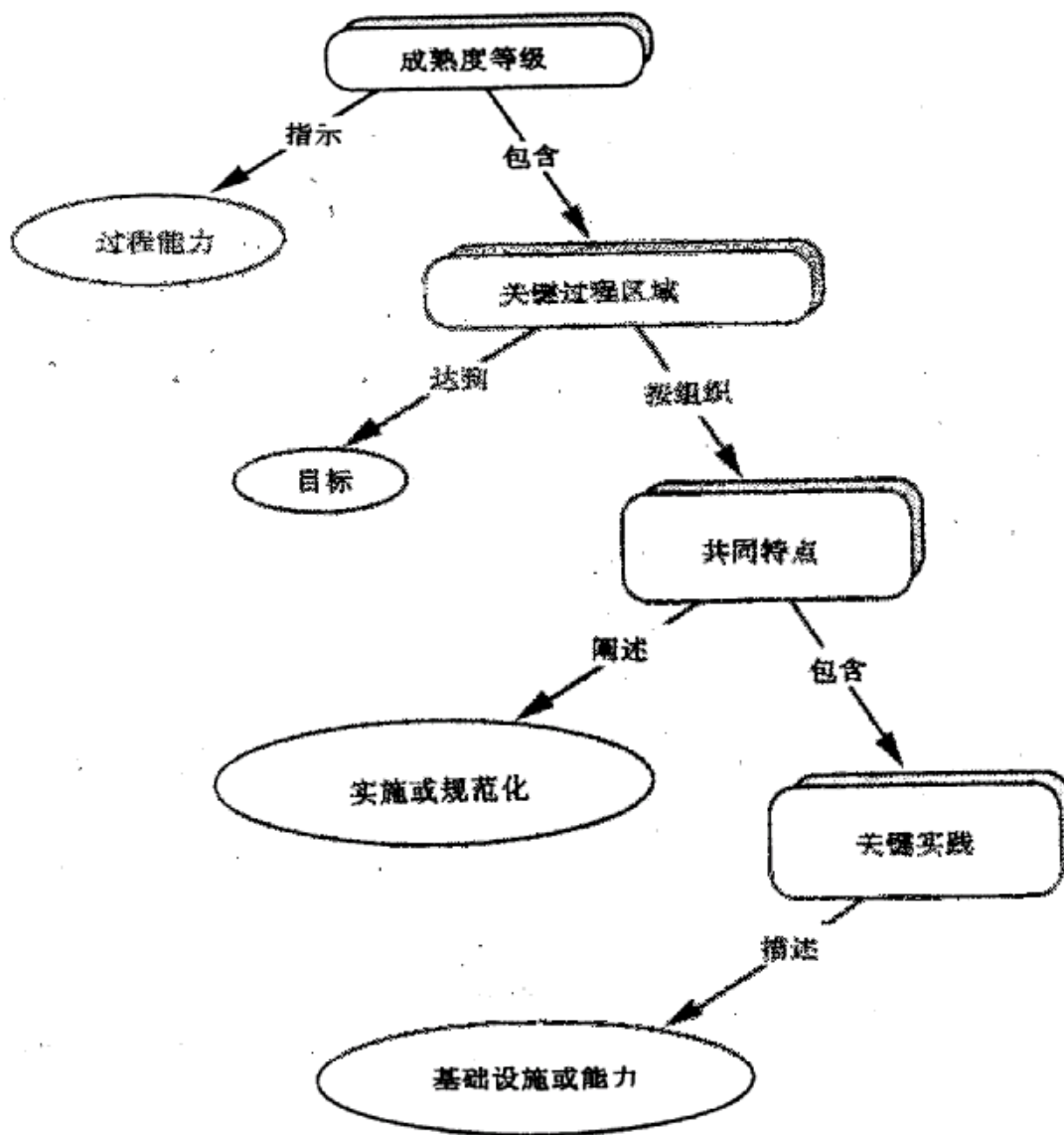


图 2.1 能力成熟度模型的结构

CMM 的成分包括：

成熟度等级
(Maturity levels)

一个成熟度等级是在朝着实现成熟软件过程进化选中的一个妥善定义的平台。五个成熟度等级提供 CMM 的顶层结构。

过程能力
(Process Capability)

软件过程能力描述通过遵循软件过程能实现预期结果的程度。一个组织的软件过程能力提供一种预测组织承担下一个软件项目时预

关键过程区域 (KeyProcessareas)	<p>期的最可能结果的方法。</p> <p>每个成熟度等级由若干关键过程区域组成。每个关键过程区域标识出一串相关的活动，当它们作为群体完成时，就达到一组目标，此组目标对建立该过程成熟度等级是至关重要的。关键过程区域是分别定义在各个成熟度等级并与之相连在一起的。例如，等级 2 的一个关键过程区域是软件项目策划。</p>
目标 (Goals)	<p>目标概括一个关键过程区域中的关键实践，并可用于确定一个组织或项目是否已有效地实施该关键过程区域。目标表示每个关键过程区域的范围、边界和意图。例如，软件项目策划关键过程区域的一个目标是“软件估计已文档化，供策划和跟踪软件项目使用。”参见“软件能力成熟度模型，1.1 版”[Paulk93a]和本文的 4.5 节（即运用专业判断），能得到更多的有关解释目标的信息。</p>
共同特点 (CommonFeatures)	<p>将关键实践分别归入下列五个共同特点中：执行约定、执行能力、执行的活动、测量和分析、及验证实施。共同特点是一种属性，它能指示一个关键过程区域的实施和规范化是否是有效的、可重复的和持久的。执行的这个共同特点描述实施活动。其余四个共同特点描述规范化因素，它们使得过程成为组织文化的一部分。</p>
关键实践 (KeyPractices)	<p>每个关键过程区域用若干关键实践加以描述，当实施这些关键实践时，能帮助实现该关键过程区域的目标。关键实践描述对关键过程区域的有效实施和规范化贡献最大的基础设施和活动。例如，软件项目策划这个关键过程区域的一个关键实践是“按照已文档化的规程制定项目的软件开发计划”。</p>

2.4 CMM 成熟度等级定义

组织建立和改进他们用以开发和维护其软件下作产品的软件过程时，他们沿着成熟度等级逐步前进。图 2.2 显示 CMM 的五个成熟度等级。

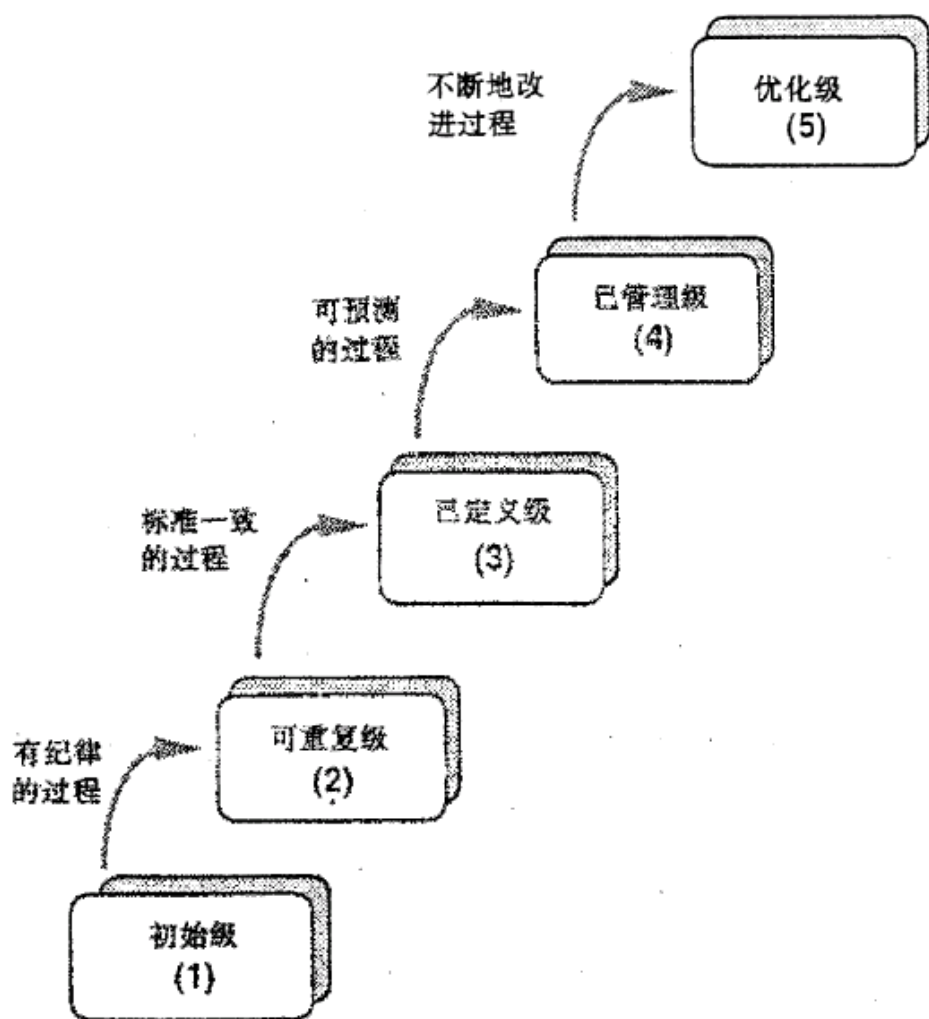


图 2.2 软件过程成熟度的五个等级

每个成熟度等级为连续过程改进提供一层基础。每个关键过程区域包含一组目标，当这组目标实现时，就使得软件过程的一个重要成分稳定。达到成熟度模型中的每一个等级，都使得软件过程的一个不同的成分规范化，导致组织过程能力的整体增长。

2.4.1 等级 1——初始级

在初始级上组织一般不能提供开发和维护软件的稳定环境。当组织中缺乏健全的管理实践时，无效的策划和反应驱动的保证体系会削弱好的软件工程实践所带来的受益。

在危机时刻，项目一股抛弃预定的规程，回复到仅作编码和测试。成功完全依赖于有一个杰出的经理及一支有经验的、战斗力强的软件队伍。偶尔，有能力的、坚强的软件经理能经受住要他们在软件过程中走捷径的压力，但是当他们离开项目后，他们能使过程稳定的影响也随之消失。甚至一个强的工程过程也不能克服由于缺乏健全的管理实践所造成的不稳定。

等级 1 组织的过程能力是不可预测的，因为随着工作进展软件过程经常被改变或修定（即过程是无秩序的（ad hoc））。进度、预算、功能性和产品质量一般是不可预测的。性能

依赖于个人的能力，且随个人固有的技能、知识和动机的不同而变化。等级 1 组织几乎没有明显的稳定的软件过程，只能通过个体的能力而不是组织的能力去预测性能。

2.4.2 等级 2——可重复级

在可重复级上，已建立管理软件项目的方针和实施这些方针的规程。基于在类似项目上的经验对新项目进行策划和管理。达到等级 2 的目的是使软件项目的有效管理过程制度化，这使得组织能重复在以前项目上所开发的成功实践，尽管项目所实施的具体过程可能不同。一个有效过程可特征化为实用的、已文档化的、已实施的、已培训的、已测量的和能改进的。

等级 2 组织中的项目已设置基本的软件管理控制。实际可行的项目约定是基于对以前项目的观察结果和当前项目的需求。项目的软件经理跟踪软件成本、进度和功能；识别在满足约定方面出现的问题。对软件需求和为实现需求所开发的工作产品建立基线，并控制其完整性。软件项目标准已确定，并且组织能保证准确地执行它们。如果有子承包商的话，软件项目与他们一起努力建立一种强有力的顾客—供应商关系。

等级 2 组织的过程能力可概括为有纪律的，因为软件项目的规划和跟踪是稳定的，能重复以前的成功。由于遵循实际可行的基于以前项目性能的计划，项目过程处在项目管理系统的有效控制之下。

2.4.3 等级 3——已定义级

在已定义级上，全组织的开发和维护软件的标准过程已文档化，包括软件工程过程和软件管理过程，而且这些过程被集成为一有机的整体。在 CMM 中的所有地方，均称此标准过程为组织标准软件过程。运用等级 3 上所建立（适当时，经更改）的过程，以帮助软件经理和技术人员工作得更有效。组织在使其软件过程标准化时，利用有效的软件工程实践。存在一个负责组织软件过程活动的组，例如软件工程过程组（SEPG）[Fowle90]。实施全组织的培训计划以保证职员和经理具有履行其职责所必须的知识和技能。

项目通过剪裁组织标准软件过程去建立他们自己定义的软件过程，它说明项目独有的特征。在 CMM 中，这种剪裁后的过程称作项目定义软件过程。一个已定义软件过程包含一组协调的、集成的、妥善定义的软件工程过程和管理过程。妥善定义的过程可特征化为具有准备就绪判据、输入、标准、进行工作的规程、验证机制（例如同行评审）、输出以及完成判据。因为软件过程已妥善定义，管理者就能洞察所有项目的技术进展。

等级 3 组织的软件过程能力可概括为标准的和一致的，因为无论软件工程活动还是管理活动，过程都是稳定的和可重复的。在所建立的各种产品线（Productlines）内，成本、进度和功能性均受控，对软件质量也进行跟踪。这种过程能力建立在整个组织范围内对已定义过程中的活动、角色和职责的共同理解之上。

2.4.4 等级 4——已管理级

在已管理级上，组织对软件产品和过程都设置定量的质量目标、作为组织测量大纲的一部分，对所有项目都测量其重要的软件过程活动的生产率和质量。利用一个全组织的软件过程数据库收集和分析从项目定义的软件过程中得到的数据。等级 4 上的软件过程已经配备有妥善定义的和一致的度量。这些度量为定量地评价项目的软件过程和产品打下基础。

项目通过将其过程性能的变化限制在定量的可接受的范围之内,实现对其产品和过程的控制。可以将过程性能方面有意义的变化,与随机变化(噪声)区别开来,特别在所建立的各种产品线内。开发新应用领域的软件所带来的风险是已知的并得到精心的管理。

等级 4 组织的软件过程能力可概括为可预测的,因为过程是已测量的并在可测的限制范围内运行。该等级的过程能力使得组织能在定量限制的范围内预测过程和产品质量方面的趋势。当超过限制范围时,采取措施予以纠正。软件产品具有可预测的高质量。

2.4.5 等级 5——优化级

在优化级,整个组织集中精力进行不断地过程改进。为了预防缺陷出现,组织有办法识别出弱点并预先针对性地加强过程。在对新技术和所建议的组织软件过程的更改进行费效分析时利用有关软件过程有效性的数据。识别出采用最佳软件工程实践的技术创新并推广到整个组织。

等级 5 组织的软件项目群组分析缺陷以确定其发生的原因。为了防止已知类型的缺陷再次出现他们认真评价软件过程,同时将经验教训告知其它项目。

等级 5 组织的软件过程能力可特征化为不断改进,因为这些组织为扩大其过程能力的范围进行不懈的努力,因而不断改善其项目的过程性能。既通过现有过程中增量式前进的办法,也通过采用新技术、新方法的革新办法,使改进不断出现。

2.5 CMM 的关键过程区域

图 2.3 中列出 CMM 中每个成熟度等级的关键过程区域。每个关键过程区域识别出一单相关活动,当这些活动全部完成时,能达到一组对增强过程能力至关重要的目标。每个关键过程区域定义在单个成熟度等级上。关键过程区域是一些结构单元,它们指明组织为改进其软件过程应关注的区域。关键过程区域鉴别出为达到某一成熟度等级必须解决的问题。

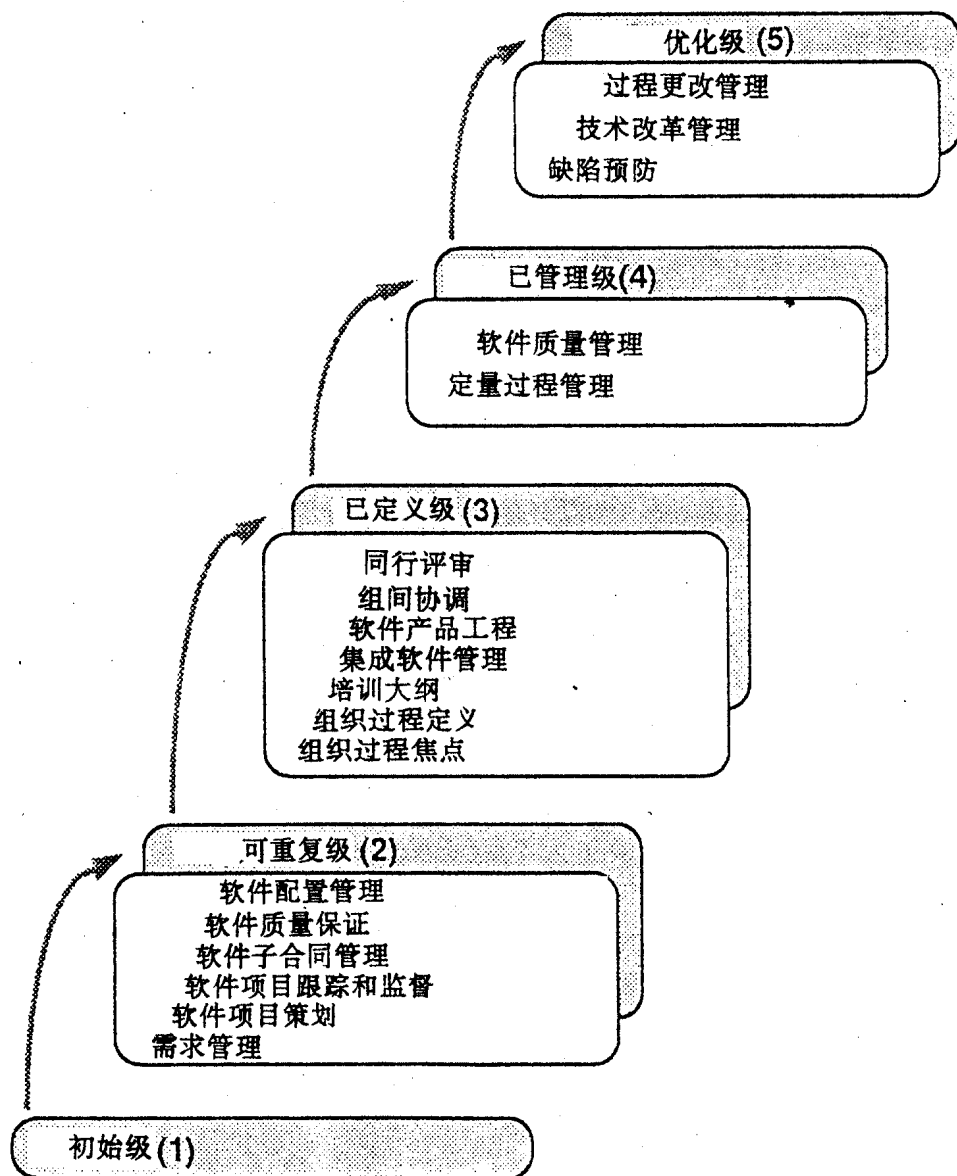


图 2.3 按成熟度等级排列的关键过程区域

等级 2 上的关键过程区域集中关注软件项目所关心的、与建立基本项目管理控制有关的事情。下面列出对等级 2 上每个关键过程区域的描述；

需求管理的目的是在顾客和软件项目之间建立对顾客需求的共同理解，顾客需求将由软件项目处理。与顾客的协议是策划（正如在软件项目策划中所描述的）和管理（正如在软件项目跟踪和监督中所描述的）软件项目的基础。对与顾客关系的控制依靠遵循有效的更改控制过程（正如在配置管理中所描述的）。

项目策划的目的是制定进行软件工程的和管理软件项目的合理的计划。这些计划是管理软件项目的必要基础（正如在软件项目跟踪和监督中所描述的）。没有切合实际的计划就不可能实施有效的项目管理。

项目跟踪和监督的目的是建立适当的对实际进展的可视性，使管理者在软件项目性能显著偏离软件计划时能采取有效的措施。

子合同管理的目的是选择合格的软件子承包商，并有效地管理它们。它把用于基本管理控制的需求管理、项目策划、以及软件项目跟踪和监督等关键过程区域所关注的事情与软件质量保证和软件配置管理等关键过程区域中必不可少的协调结合在一起，并且当合适时对于承包商实施这项管理。

软件质量保证的目的是给管理者提供对于软件项目正采用的过程和正在构造的产品的恰当的可视性。软件质量保证是绝大多数软件工程过程和管理过程的不可缺少的部分。

软件配置管理的目的是在项目的整个软件生存周期中建立和维护软件产品的完整性。软件配置管理是绝大多数软件工程过程和管理过程的不可缺少的部分。

等级 3 的关键过程区域既阐述项目问题，又阐述组织的问题，这是因为组织建立起使涉及所有项目的、有效的软件工程和管理过程规范化的基础设施。下面列出对等级 3 上每个关键过程区域的描述：

组织过程焦点的目的是规定组织在改进其整体软件过程能力的软件过程活动方面的职责。组织过程焦点活动的主要结果是一组软件过程财富，它们在组织过程定义中加以描述。正如集成软件管理中所描述的，这些财富供软件项目使用。

组织过程定义的目的是开发和保持一组便于使用的软件过程财富，它们能改进横跨项目的过程性能，并且为组织获得积累性的、长期得益奠定基础。这些财富提供一组稳定的基本原则，通过诸如培训等机制就能使其成为制度，培训在培训大纲中加以描述。

培训大纲的目的是培育个人的技能和知识使得他们能有效地和效率高地执行其任务。尽管培训是组织的责任，但是软件项目应该识别出他们所需要的技能，当项目的需求独特时，该项目应提供所需要的培训。

集成软件管理的目的是将软件工程活动和管理活动集成为一个协调的、已定义的软件过程，该过程是剪裁组织标准软件过程和组织过程定义中所描述的相关的过程财富而得到的。剪裁基于项目的经营环境和技术需要，正如在软件产品工程中所描述的那样，集成软件管理是从等级 2 的软件项目策划和软件项目跟踪和监督进化而得到的。

软件产品工程的目的是—致地执行一个妥善定义的工程过程，为了能有效地和效率高地生产正确的、一致的软件产品，该工程过程集成全部软件工程活动。软件产品工程描述项目的技术活动，例如，需求分析、设计、编码和测试。

组间协调的目的是为软件工程组积极参与其它工程组工作建立一种手段，使得项目更能有效地和效率高满足顾客的需求。组间协调是集成软件管理的一个涉及多学科的方面，它延伸到软件工程之外；不仅应该集成软件过程，而且软件工程组和其它组之间的相互作用也必须加以协调和控制。

同行评审的目的是及早和高效地除去软件工作产品中的缺陷。一个重要的必然结果是增强对软件工作产品和可预防的缺陷的了解。同行评审是一种重要而又有效的工程方法，在软件产品工程中调用此方法，可通过法根式审查（Fagan - style 审查）[Fagan86] 结构化走查、或者一些其它的学院式的评审方法[Freedman90]加以实施。

等级 4 上的关键过程区域的关注焦点是建立起对软件过程和正在构造的软件工作产品的定量了解。正如下所述，该等级上的两个关键过程区域—定量过程管理和软件质量普理——是互相紧密依赖的。

定量过程管理的目的是定量地控制软件项目的过程性能。软件过程性能表示遵循一个软件过程所得到的实际结果。焦点是在一个可测的稳定的过程范围内鉴别出变化的特殊原因，并且适当时改正那些促使瞬时变化出现的环境。定量过程管理给组织过程定义、集成软件管理、组间协调和同行评审的实践附加一个内容丰富的测量计划。

软件质量管理的目的是建立对项目的软件产品质量的定量了解和实现特定的质量目标。软件质量管理对软件产品工程中所描述的软件工作产品实施一个内容丰富的测量计划。

等级 5 上的关键过程区域包括那些为了实施连续不断的和可测的软件过程改进，组织和项目都必须解决的问题。下面列出等级 5 的每个关键过程区域的描述：

缺陷预防的目的是鉴别缺陷的原因并防止它们再次出现。正如在集成软件管理中所描述的，软件项目分析缺陷，鉴别其原因，和更改项目定义软件过程。正如在过程复政管理中

所描述的，应将具有普遍价值的过程更改通知到其它软件项目。

技术改革管理的目的是识别出能获利的新技术（即工具、方法和过程），并以有序的方式将它引进到组织中去，正如在过程更改管理中所描述的那样。技术改革管理的关注焦点是在不断变化的环境里高效率地进行创新。

过程更改管理的目的是本着改进软件质量、提高生产率和缩短产品开发周期的目的持续不断地改进组织中所采用的软件过程。过程更改管理既采用缺陷预防的增量式改进，又采用技术改革管理的创新式改进，并使得整个组织可以享用这些改进。

按定义，每个关键过程区域仅与单个成熟度等级有关。可是在关键过程区域之间存在着一定的关系，而且某些特定的管理或技术区域方面的改进并不必限定在单个关键过程区域。图 2.4 显示这些关系。组织在它们达到某个较低成熟度等级之前可能工作在较高等级的关键过程区域，并且甚至在已达到较高成熟度等级的关键过程区域时，还必须继续关注较低等级的关键过程区域。

在图 2.4 中关键过程区域被归入三大类：管理过程、组织方面的过程和工程过程。管理过程类包括项目管理活动，它们从等级 2 的策划和跟踪进化到等级 3 上按已定义的软件过程进行管理，到等级 4 上的定量管理，到等级 5 上在经常变化环境中的创新管理。组织方面的过程类包括随着组织逐渐成熟而变化的跨项目的职责，开始在等级 3 上是对过程问题的关注，

继而在等级 4 上是对过程的定量理解，最后在等级 5 上是在不断的过程改进环境中对更改的管理。工程过程类包含技术活动，诸如：需求分析、设计、编码和测试等，在所有等级上都进行这些活动，但是它们又是不断进化的，朝着等级 3 的工程学科、等级 4 的统计质量控制和等级 5 的已测量的不断改进进化。

注意在等级 4 和等级 5 上有横跨这些过程类的关键过程区域。随着对等级 4 和等级 5 的了解加深，这能帮助为 CMM2.0 版识别出新的潜在的关键过程区域。

2.6 关键实践

每个关键过程区域用能对实现其目标作出贡献的关键实践加以描述。关键实践描述对关键过程区域的有效实施和规范化贡献最大的基础设施和活动。

每个关键实践的组成是：一个单个句子，常常紧跟着较为详细的描述，后者可能包括例子和详细细节。前者，又称为顶层关键实践，说明关键过程区域的基本方针、规模和活动。进行详细描述的分量常称作于实践。

关键实践描述应作什么，但是不应解释为强制如何实现目标。替代的实践也可能实现该关键过程区域的目标。应该合理地解释关键实践以便判断关键过程区域的目标是否已有效地实现，尽管实现目标的方式可能不同。

2.7 目标

目标概括一个关键过程区域中的关键实践，并能用于确定一个组织或项目是否已有效地实施关键过程区域。目标表示每个关键过程区域的范围、边界和意图。在调整一个关键过程区域的关键实践使其适应于一个具体项目时，目标可用于确定该项调整是否对原实践作出合理的描述。类似地，当评估或评价实施关键过程区域的替代方法时，目标可用于确定替代方法是否满足关键过程区域的意图。请参考“软件能力或成熟度模型，1.1 版”（FPaulk 93a）和本文的节 4.5（运用专业判断），以便得到更多的有关解释一个组织的目标的信息。

2.8 共同特点

每个关键过程区域中的关键实践按一组共同特点加以组织。共同特点是表明一个关键过程区域的实施和规范化是否有效、可重复且持久的一些属性。共同特点也将关键实践分组和排成序列，方便组织使用它们。五个共同特点列举如下：

执行约定执行约定描述组织为保证过程得以建立和继续起作用所必须采取的行动。执行约定一般包括制定组织的方针和建立高级管理者的支持。

执行能力执行能力描述为了能实施软件过程，项目或组织中必须存在的先决条件。执行能力一般包括资源、组织机构和培训。

执行的活动执行的活动描述为实现一个关键过程区域所必须的岗位和规程。执行的活动一般包括制定计划和规程、进行工作、跟踪它、并在需要时采取纠正措施。

测量和分析测量和分析描述对过程进行测量和对测量结果进行分析的需要。测量和分析一般包括为了确定所执行活动的状态和有效性所能采用的测量的例子。

验证实施验证实施描述那些能保证遵照已建立的过程进行活动的措施。验证一般包括管理者和软件质量保证部门所作的评审和审计。

在执行的活动这一共同特点中的实践描述为建立过程能力必须作些什么。而其它的实践作为一个整体形成了一个基础，在此基础上组织能规范化执行的活动中描述的实践。由项目或组织所执行的活动构成关键实践中最大的一类，因为它们描述关键过程区域的实际实施。在其它共同特点下的关键实践也是同等重要的，因为它们阐述为了支持和规范化关键过程区域而必须做的事情。

3 关键实践的版面介绍

关键实践按成熟度等级分组。而等级间又用一个标记页分开。标记页上包括成熟度等级的描述、那个等级的关键过程区域的清单，以及每个关键过程区域的起始页码。

每个关键过程区域包括：

- ▮ 关键过程区域的简要描述，
- ▮ 关键过程区域的目标，
- ▮ 关键实践。

关键实践本身又被分组到五个共同特点（即执行约定、执行能力、执行的活动、测量和分析、及验证实施）的名下，并且以层次式的格式表示，正如在图3.1中所显示的一个关键实践的版面例子。

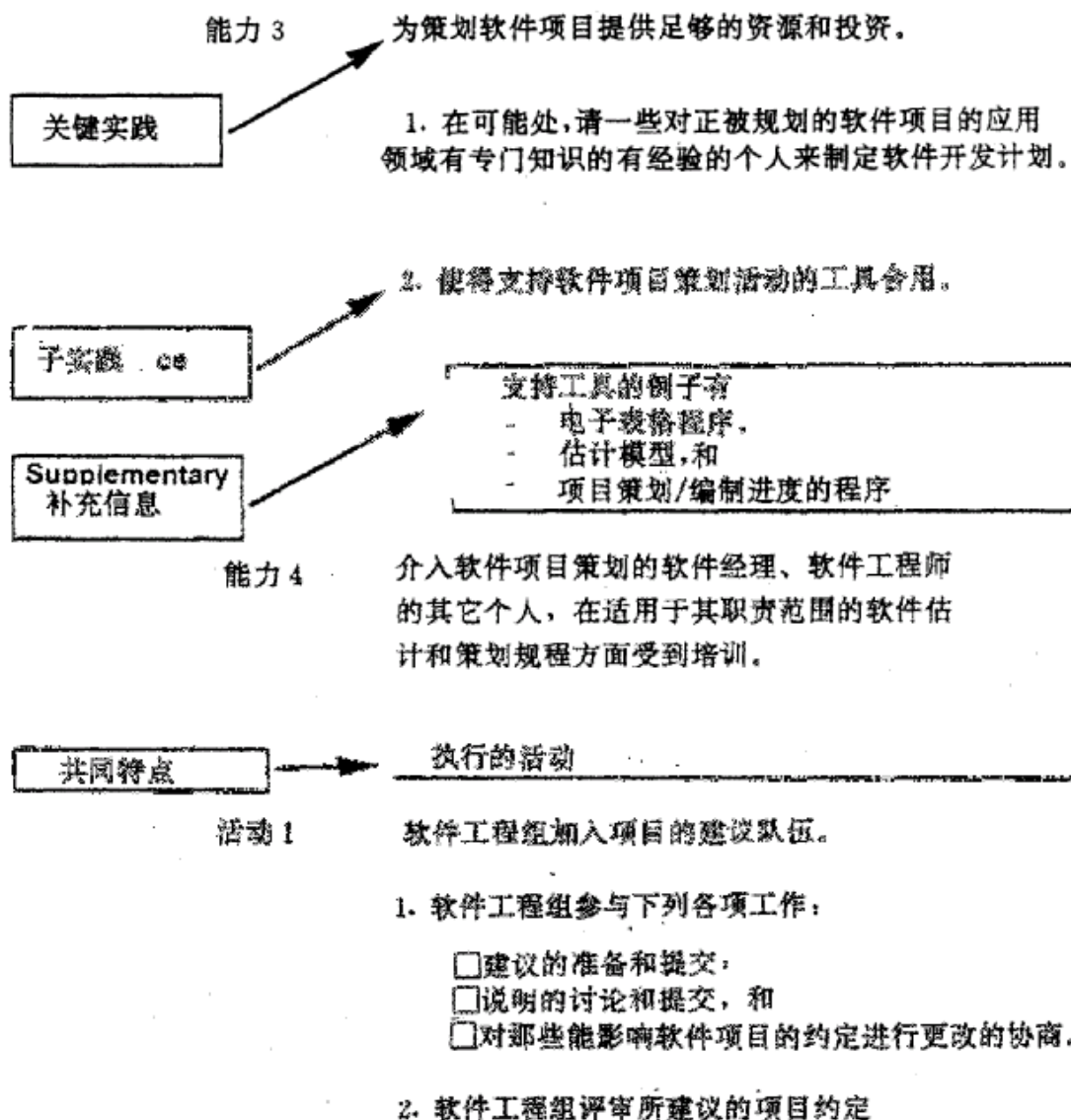


图 3.1 关键实践陈述的例子

关键实践包括：

- | | |
|--|---|
| 关键实践
(key practices) | 关键实践，又称为顶层关键实践，阐述关键过程区域的基本方针、规程和活动。他们以粗体字表示并在每个共同特点内加以编号。例如在执行的活动这一共同特点中的第一个关键实践被标志为活动 1。 |
| 子实践
(subpractice) | 子实践，又称下级关键实践，被列在顶层关键实践的下面，它描述人们所预期的，为完成顶层实践而应该做的事。于实践可用于帮助确定关键实践是否得到满意的实施。 |
| 补充信息
(Supplementary
Information) | 在关键实践之后用方括号括起补充信息。补充信息包括例子、详细解释，以及对其它关键过程区域的索引。 |

当一个关键实践下的子实践或补充信息扩展到另一页时 ,在新页开头处的园括号内标出该关键实践以指示这页的信息是前页关键实践信息的继续。

4 解释 CMM

4.1 解释关键实践

我们制定关键实践的意图不是要求或者主张某个具体的软件生存周期模型,某种具体的组织机构、某种具体的职责划分或某种具体的用于开发的管理和技术方法,而是给出对有效软件过程的基本元素的描述。

打算用关键实践来传达那些基本原理,它们能适用于种类广泛的项目和组织,对许多典型的软件应用都正确,且随着时间的推移仍然有效。所以我们的办法是描述原理,而将其具体实施留给每个组织,按其文化和职员及经理的经验去做。

虽然关键实践不依靠任何具体的实施,但在叙述关键实践时必须一致地使用特定的术语和例子以提高文字的清晰度。本节描述 CMM 中对角色、职责、关系、产品和活动所用的约定术语。使用关键实践的组织应该了解这些约定术语,并在自己的组织、项目和经营环境中恰当地说明它们。

附件 B 中的术语汇编收纳了术语的定义,包括在本节和其它地方已描述过的术语。

4.2.1 解释共同特点

在关键实践的每个共同特点中,为了在关键过程区域之间提供连续性和一致性,采用某些短语和约定术语。下面描述组织机构方面的主要的约定术语,按共同特点排列。

4.2.1 执行约定

方针陈述 (Policy statement)	<p>当采用方针陈述时,它们一般指的是项目遵循书面的、用于那个关键过程区域的实践的组织上的方针。这是为了强调在组织上的约定和实际正进行工作的项目之间的连结。</p> <p>方针陈述的子实践通常概述以后将被包括在关键过程区域中并特别适合于通过书面方针加以规范化的活动。</p> <p>在某些关键过程区域(例如组织过程焦点)中,关键过程区域的活动的焦点是组织,而不是项目。这种情况下方针陈述要改写,并且指的是组织遵循书面方针。</p>
领导 (leadership)	<p>在某些关键过程区域,执行约定包含一个阐述任命一个领导角色(例如项目软件经理)的语句,或者一个描述特别资助活动的语句,这些是成功地规范化该关键过程区域所必须的。</p>

4.2.2 执行能力

资源和投资	大多数关键过程区域包含一个反映关键过程区域的活动对适当
-------	-----------------------------

(Resources and funding)	<p>的资源 and 投资要求的关键实践。由子实践描述的资源及投资一般分成三类；能利用的特别的技能、足够的投资和能利用的工具。作为例子，列出在执行关键过程区域活动中可能利用的工具。</p> <p>采用“投资（funding）”这个词而不用“预算（budget）”这个词是为了强调，对于实际过程而言，已交付的和已使用的东西要比许诺的东西更有实际价值。</p>
培训 (training)	<p>对于术语培训，CMM 中所指的范围比通常使用此术语时所考虑的要宽。提供培训是为了使得个人精通专门的细则和实践。该培训可以包括正式的和非正式的在组织中将技能和知识传授给个人的办法。虽然课堂培训是许多组织所采用的培育雇员技能的办法，但 CMM 还提供其它的手段，例如方便的录相教学、计算机辅助指导、或者正规的师徒计划。培训大纲这个关键过程区域描述与这些培训办法有关的具体实践。在 CMM 的所有地方一般能发现描述培训的两个样板短语。在等级 2 使用短语“接受培训”，在等级 3 以上使用短语“接受所要求的培训”。使用这些不同样板短语的意图是，区别出等级 2 上的培训多半尚未在全组织范围内规范化。在等级 3 和更高等级，预料培训大纲这个关键过程区域的关键实践能控制组织的培训活动。在所有的关键过程区域，潜在可能的培训专题用例子块表示，承认不同的组织情况多半有不同的具体培训需要。</p>
走向培训 (orientation)	<p>在某些关键过程区域，能找到描述定向培训的关键实践。广泛使用术语定向培训去指明所传授的技能和知识的深度要比由培训传授的浅。定向培训是对一个专题所作的概述或介绍，其对象是那些对在该专题范围内负责进行工作的个人加以监督和打交道的人。</p>
先决条款 (Prerequisite Items)	<p>某些关键过程区域包括描述要求先决条款的关键实践；例如，软件开发计划是软件项目跟踪和监督的先决条款。在某些情况下，这些先决条款可能为另一个关键过程区域活动的输出。在另一些情况下，它们是预期能从软件项目范围之外获得的条款（例如，分配给软件的系统需求是需求管理的先决条款）。</p> <p>与 CMM 突出“关键”实践的哲理相一致，对每个关键过程区域并不列出全部先决条款。在 CMM 中仅仅引用那些已被发现对关键过程区域的实施起特别关键作用的先决条款。</p>

4.2.3 执行的活动

在产所有的共同特点中，执行的活动显示出最多的结构可变性，因为关键过程区域的实施活动在细节层次上、组织方面的焦点（例如，项目或组织）上，和对策划及文档化的要求上各不相同。下面突出阐述某些一般的概念。

计划的类型 (Types Of plans)	<p>在关键实践中所描述的两个主要的计划类型是：正式计划（例如：软件开发计划、软件质量保证计划和软件配置管理计划）及非正式计划（例如同行评审计划、风险管理计划和技术管理计划）。</p> <p>非正式计划一般编写为正式计划的一部分（例如，同行评审计划</p>
---------------------------	--

	<p>可以作为软件开发计划的一部分) 或者作为正式计划的附件(例如, 同行评审进度表可以是软件开发计划中的一节)。正式计划要求有管理者的高度支持, 无论从制定这些计划的立场上看还是从保证它们得到遵循的立场上看均是如此。在签约环境中这些计划一般交付给作为合同甲方的顾客。</p>
正式计划 (Formal Plans)	<p>在提到正式计划时, 通常有两个关键实践专门描述策划活动: 一个关键实践要求按照已文档化的规程制定和修改计划, 另一个要求关键过程区域的活动基于计划。涉及已文档化的规程的子实践的内容包括: 计划的输入所必须有的内容和为了获得计划所要求的约定和支持预计需采取的步骤。这些干实践标识出该计划的典型评审者, 还突出说明预期的批准等级。</p> <p>有些计划是活动的基础, 与这些计划有关的干实践描述这些计划所预计有的内容。在描述计划内容方面可以提供不同层次的细节, 它取决于计划的类型和在包括哪些一般的计划专题方面有关组织灵活性的要求。</p>
非正式计划 (Informal plans)	<p>通常非正式计划由单个关键实践描述。其子实践包括有关计划内容的信息和用于制定或修定计划的规程。</p>
按照文档化的规程 (According to a documented procedure)	<p>一般需要一个文档化的规程, 这样, 负责一个作业或活动的个人能以重复的方式执行它, 而且其它具有该领域一般知识的人员也能学会并以同样的方式执行该作业和活动。这是使过程规范化的一个方面。文档化规程的形式和细节层次可能变化很大, 从手写的个人的桌面规程到正式的组织的标准操作规程。形式和细节层次依赖于谁将执行该作业或活动(例如, 个人或群组), 执行频率, 结果的重要性和所预计的用途, 以及结果的预计接收者。</p>
分配给软件的系统需求 (System requirements allocated to Software)	<p>在 CMM 中分配给软件的系统需求通常称为“分配需求”, 它是系统需求中将用系统的软件成分实现的子集。分配需求是软件开发计划的主要输入。软件需求分析详细阐述和精炼分配需求, 结果产生文档化的软件需求。</p> <p>顾客需求涉及整个的系统, 不只是软件。CMM 中, 对顾客需求的讨论集中于那些将用软件实现的顾客需求。将系统需求分配给硬件、软件等等是整个系统设计的一部分, 一般由系统工程组完成。分配给软件项目的系统需求在 CMM 中一般称作“分配需求”, 既包括技术需求(功能性、性能等等), 也包括非技术需求(交付日期、成本等等)。</p>
顾客-供应商关系 (Customer supplier relationship)	<p>顾客可以是组织内部的, 也可以是组织外部的。内部顾客的一个例子是营销小组; 外部顾客的例子是国防部。用户也可能不同于顾客, 国防部签约环境中情况正是如此。CMM 是用外部顾客加以表述的, 他们是采购带有关键软件成分的系的人员或组织在需要处必须恰当地解释(正如在 CMM 中所述的)组间的边界, 例如, 在仅仅采购软件的情况下, 顾客和软件工程组之间可以没有系统工程组。此时, 顾客需求、系统需求和分配需求可能是同义语, 系统工程组的职责将由顾客及软件工程组分担。</p>
按管理要求进行跟踪和采取纠正措施 (Tracking and	<p>等级 2 在软件项目跟踪和监督关键过程区域中的许多关键实践采用短语“跟踪..., 必要时采取纠正措施。”在等级 3 的集成软件管理中的许多类似的关键实践采用短语“进行管理”。这个用字上的差异反映出等</p>

Taking corrective action versus managing)	级 2 上的项目缺乏一个完全定义的软件过程。管理措施很可能是对实际问题的反应。在等级 3 上项目有一个完全定义的软件过程，不同软件工作产品、作业和活动之间的关系也是妥善定义的。管理者更能预见会出现的问题并能预先采取措施防止它们发生。在要求进行干预时，干预对整个软件过程的效果是清楚的，因而能更有效地定义和运用这些干预。
经过评审和经过同行评审的比较 (Reviewed versus under goes peer reviews)	在评审中，将一个软件工作产品或一组工作产品提交给经理、顾客、最终用户、或其它有兴趣的个人，以得到他们的评论或批准。评审一般出现在作业结束时。在同行评审中，将一个软件工作产品或一组工作产品提交给生产者的同事们以识别出缺陷。经理、顾客或终端用户一般不出席同行评审。同行评审是作业过程中不可缺少的部分。进行同行评审使得缺陷能及早排除，导致较高的生产率和高质量的产品。某些软件工作产品将经过评审；某些将经过同行评审；还有一些将既经受评审又经受同行评审。
置于配置管理之下 与之进行管理和控制相比较 (Placed under configuration management versus managed and Controlled)	<p>某些软件工作产品，例如软件设计和代码，应在预先确定的点上建立基线。这些基线经过正式评审和认同，作为进一步开发工作的基础。对已基线化的配置项施加严格的更改控制过程。在与顾客打交道时，这些基线提供控制和稳定性。有时称此为基线配置管理。短语“置于配置管理之下”用于此类软件工作产品。当配置控制由开发者实施时，通常称之为开发配置管理。在预先确定的开发中的某些点上，开发配置管理下的某些项可置于基线配置管理之下。虽然对短语“置于配置管理之下”的解释可扩展至开发配置管理，但是正确的基本解释是仅要求实施基线配置管理。</p> <p>某些软件工作产品，例如估计或软件开发计划，可以不必置于配置管理之下，但仍然需要“进行管理和控制”。这些软件工作产品不是基线的一部分，因此不用置于配置管理之下，但它们必须受控以使得项目按一种有纪律的方式进行工作，短语“进行管理和控制”用于对标识和定义这种软件工作产品的过程进行特征描述。“进行管理和控制”意味着在给定时间（过去或现在）使用的工作产品的版本是已知的（即版本控制），并且更改以一种受控的方式引入（即更改控制）。</p>

4.2.4 测量和分析

测量和分析这个共同特点中的关键实践描述基本的测量实践，对确定与关键实践的执行的这一共同特点有关的状态来说，这些测量实践是必不可少的。有些测量是关键过程区域中活动的固有部分，这些测量则被列在执行的这一共同特点之下。

所建议的测量例子都表示为补充信息，因为项目环境的可变性可以导致不同的测量需要和方法。

4.2.5 验证实施

验证实施这个共同特点一般包含与高级管理者和项目管理者的监督有关的关键实践，以及某些专门的验证活动，后者预期由质量保证组成其它组执行，以证实关键实践正在恰当地

进行。

高级管理者的定期监督 Senior management oversight on a periodic basis	<p>高级管理者作定期评审的主要目的是在合适的抽象层次上和以及时的方式了解和洞察软件过程活动。评审间隔应满足组织的需要，只要已存在有报告例外情况的合适机制，间隔可以长。</p> <p>高级管理者评审的范围和内容很大程度上依赖于评审中所涉及的是哪些高级经理。在评审的时间表和所阐述的问题方面，预期负责组织的全部软件活动的高级经理作的评审与整个组织的高级行政负责人作的评审不同。预期高级管理者的评审还会包含与项目管理者监督评审不同的专题，或者包含类似的专题但在一个更高的抽象层次上。</p>
项目管理者既定期地也事件驱动地监督 (Project management oversight on both a periodic and event driver basis)	<p>为了强调项目在不同的阶段和依赖于项目的不同特征而需要不同类型的评审，在这些关键实践的描述中采用样板短语“既定期地也事件驱动地”。项目管理者应保持对软件工作状态的不断了解，并在软件项目出现重大事件时收到报告。例子包括项目管理者参加诸如关键设计评审那样的正式评审，和围绕过程问题的评审——例如，有关过程改进规划的状态和过程不符合问题的解决等的评审。预计在项目层上，项目管理者监督要比高级管理者的监督详细，反映出项目管理者更为积极地卷入项目的运行方面。</p>
软件质量保证活动 (Software quality assurance activities)	<p>用一个关键实践描述那些据信适宜于软件质量保证（SQA）组作评审和（或）审计的特定活动。但存在一些特殊情况，在那里 SQA 验证活动未被描述，例如在培训大纲和组间协调关键过程区域中。这些是处在软件项目和组织间的边界上的关键过程区域，预计在这里 SQA 组不会有权威。</p>

4.3 解释软件过程定义

为达到较高的成熟度等级，软件过程定义是非常根本的。本节仅讨论软件过程定义的一些方面，这些方面对于使用与过程定义有关的关键实践是有帮助的，从等级 3 的组织过程定义开始讨论。

CMM 中过程定义的一个基本概念是组织标准软件过程。组织的基本过程指导建立一个近历组织中所有软件项目的共用的软件过程，而组织标准软件过程就是此基本过程的可操作的定义。组织标准软件过程描述基本的软件过程元素，它们是预计每个软件项目都会引入到自己的项目定义软件过程中去的过程元素。它也描述这些软件过程元素间的关系（例如排序和界面）。它在全组织中建立起开展软件活动的一致方式，这对长期的稳定性和长期改进是十分重要的。

在组织层上，组织标准软件过程需要以正式的方式加以描述、管理、控制和改进。在项目层上，强调的是项目定义软件过程的适用性，和它给项目带来的益处。一个项目定义软件过程是该项目所用软件过程的可操作定义。项目定义软件过程是很好地特征化了的和已理解的软件过程，用软件标准、规程、工具和方法加以描述。通过剪裁组织标准软件过程使其适合项目的具体特征来制定项目定义软件过程。

有一种过程定义的方法既能支持定义的稳定性又能支持定义的灵活性，在组织过程定义的关键实践的描述中就采用能反映这种方法的术语。图 4.1 中描述了这种方法，在随后几节中描述它的关键元素。

4.3.1 过程定义的概念

支持 SEI 在其过程定义工作中所采用方法的基本概念是能够以类似于开发和维护产品的方式来制定和维护过程。因此它必须有：

- ✍✍ 需求，它定义待描述的是什么过程；
- ✍✍ 体系结构和设计，它提供将如何定义过程的信息；
- ✍✍ 在项目或组织的情况下，过程设计的实现；
- ✍✍ 藉助于测量，对该过程描述的确认；和
- ✍✍ 在原定使用该过程的组织或项目内，该过程的推广运用。

采用类比于产品开发的方法，用于软件过程开发和维护的框架已经逐渐形成，它将这些概念转化成更为专门用于过程开发学科的概念(类似于将开发管理信息系统下所用术语转化为开发实时嵌入式系统所用术语的专门性)。图 4.1 显示该框架的关键元素，并在下面加以简要描述。

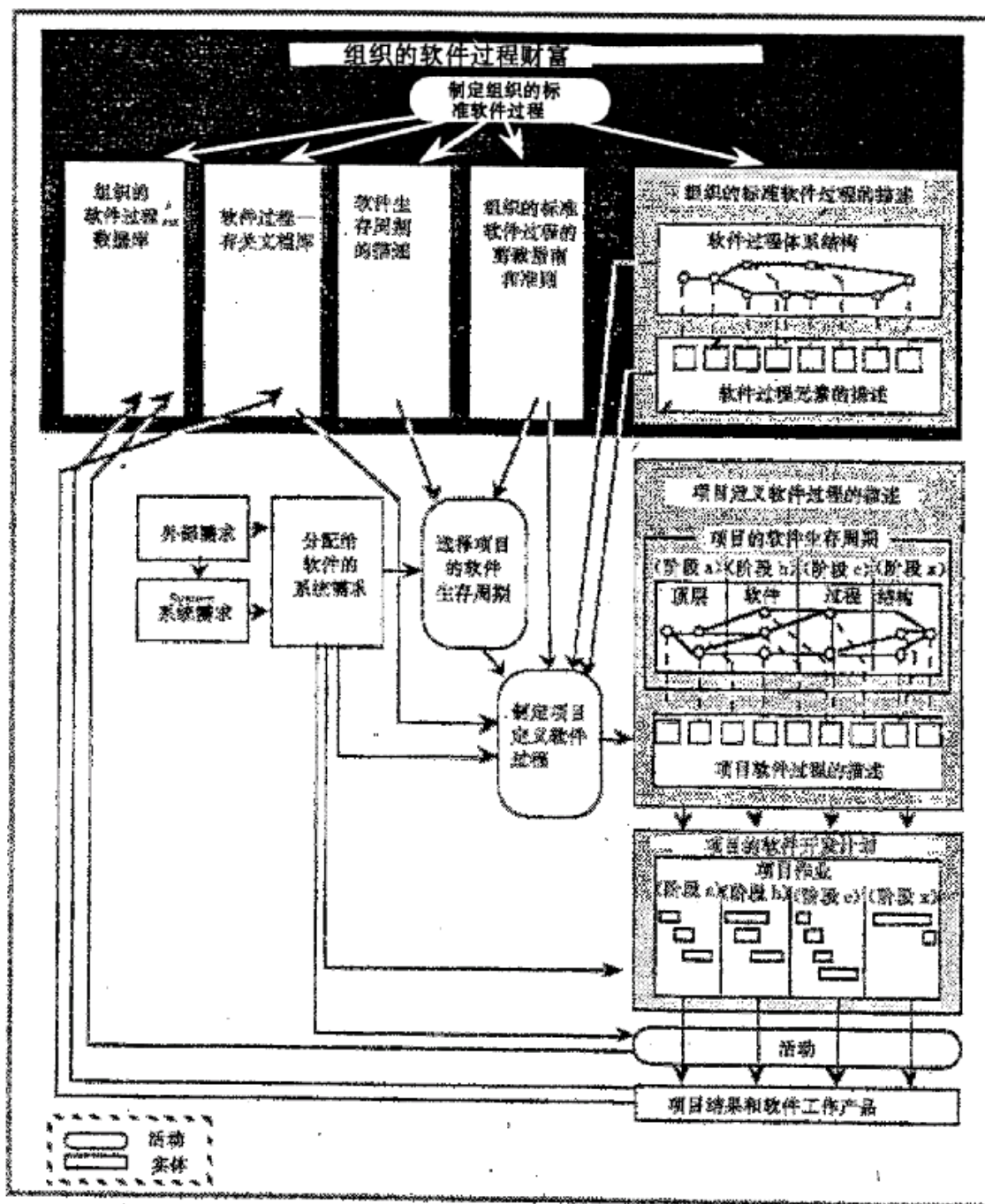


图 4.1 用在 CMM 中的概念化的软件过程框架

关于过程工程界内正在开发的过程定义的概念，进一步阅读可参考文章“Software Process Development and Enactment：Concepts and Definitions”“软件过程开发和规定：概念和定义”[Feiler 92]。

4.3.2 与组织的软件过程财富有关的概念

组织的软件过程财富 (Organizations software	组织建立和维护一组如图 4.1 所示的软件过程财富。这些软件过程财富包括：
--------------------------------------	---------------------------------------

process assets	<p>组织标准软件过程包括软件过程体系结构和软件过程元素），</p> <p>对批准使用的软件生存周期的描述，</p> <p>用手剪裁组织标准软件过程的指南和准则，</p> <p>组织的软件过程数据库，和</p> <p>软件过程有关文档库。</p> <p>这些软件过程财富可以供项目在制定、维护和实施其自己定义的软件过程时使用。组织能以多种方式组织软件过程财富，这取决于组织建立其标准软件过程的方法。例如，软件生存周期的描述可以是组织标准软件过程的一个主要部分。另一个例子是软件过程有关文档库的一些部分可以存在组织的软件过程数据库中。</p>
组织标准软件过程 Organizations standard Software process	<p>组织标准软件过程是基本过程的可操作的定义，基本过程指导在组织中建立一个针对所有软件项目的共用的软件过程。标准软件过程描述基本的软件过程元素，即预计每个项目都会纳入到其自己定义的软件过程的过程元素。它也描述这些软件过程元素间的关系（例如排序和界面）。它指导建立组织中所有软件开发和维护项目共用的软件过程。</p> <p>软件过程元素间的关系有时称为“软件过程体系结构”。</p> <p>组织标准软件过程是项目定义软件过程的基础。它保证组织过程活动的连续性，且是组织所用软件过程的测量和长期改进的依据。</p>
Software process architecture	<p>软件过程体系结构是对组织标准软件过程的高层次（即概括的）描述。它描述组织标准软件过程中软件过程元素的排序、界面、相互依赖关系及其它关系。它也描述对于其它外部过程（例如，系统工程、硬件工程和合同管理）的界面、依赖关系和它的关系。</p>
软件过程元素 Software process element	<p>软件过程元素是一个软件过程描述的构成元素。每个过程元素包括一组妥善定义了的、有限制的、紧密相关的作业（例如，软件估计元素、软件设计元素、编码元素及同行评审元素）。过程元素的描述可以是待填充的样板、待完成的片段、待精炼的抽象、或待修改的完整描述，或用过的无须修改的完整描述。</p>
对批准使用的软件生存周期的描述 Organizations software process assets	<p>软件生存周期是一段时期，从设想软件产品的时间开始，到软件不再能使用的时间为止。软件生存周期一般包括。概念阶段、需求阶段、设计阶段、实现阶段、测试阶段、安装和调整阶段、运行和维护阶段、以及有时还有退役阶段 [IEEE—STD - 610]</p>
Description of software life cycles approved for use	<p>因为一个组织可能为各种多样签约的和（或）商业的顾客和用户生产软件，一种软件生存周期也许不能适合所有的情况、因此组织可以规定多于一种的软件生存周期供项目使用。这些软件生存周期一般从软件工程文献中获得，并可加以修改，使之适于组织的情况。在制定项目定义软件过程时，这些软件生存周期可以和组织标准软件过程结合在一起使用。</p>
用于剪裁的指南和准则 Guidelines and criteria for tailoring	<p>组织标准软件过程是在通用的层次上予以描述的，因此项目可能无法直接使用它。制定指南是为了在以下方面指导软件项目：(1) 从那些批准使用的软件生存周期中挑选出一个，(2) 剪裁和细化组织标准软件过程和所选择的软件生存周期，使之适合项目的具体特</p>

	征。这些指南和准则帮助保证所有软件项目在策划、实施、测量、分析和改进项目定义软件过程时有一个共同基础。
组织的软件过程数据库 (Organizations software process L database)	组织的软件过程数据库是为了收集有关软件过程和它所生成的软件工作产品的数据并使其可供使用而建立的数据库，特别当这些数据与组织标准软件过程有关时。数据库包含或引用实际测量数据和那些为了理解测量数据及评估其合理性、可用性所必须的信息。过程数据和工作产品数据的例子有：对软件规模、工作量和成本的估计值；有关软件规模、工作量和成本的实际数据；生产率数据；同行评审范围和效率；软件代码中所发现的缺陷数及严重性。
软件过程一有关文档 Library of software process-related documentation	建立软件过程一有关文档库是为了：(1) 存储对其它当前和未来的项目可能有用的过程文档，特别当它们和组织标准软件过程有关时，(2) 使它们在全组织范围内能被共享。该库包含一些实例文档和文档片段，预计它们对未来项目在剪裁组织标准软件过程时有用。实例可能含盖以下主题：诸如项目定义软件过程、标准、规程、软件开发计划、测量计划和过程培训材料。由于提供成功项目的例子作为起步点，该库可以帮助减小起动一个新项目所要求的工作量，因此这个库是一项重要资源。

4.3.3 与项目定义软件过程有关的概念

项目定义软件过程的描述 Description of project's defined software process	项目定义软件过程的描述是项目所用软件过程的可操作的定义。项目定义软件过程是一个妥善特征化的和已理解的软件过程，用软件标准、规程、工具和方法加以描述。通过剪裁组织标准软件过程使其适合项目具体特征的方法来制定它。 剪裁包括从组织批准使用的软件生存周期中挑选出一种生存周期和修改组织标准的软件过程以适合项目的具体特征。 对于进行项目的作业和活动的经理和技术人员来说，项目定义软件过程为策划、执行和改进他们的活动奠定了基础。有可能一个项目有多于一个已定义软件过程（例如用于运行软件和用于测试支持软件）或者两个或多个类似项目共用一个已定义软件过程。
阶段 (Stage)	一个阶段是软件工作的一种划分，它有可管理的规模，描述项目所执行的一组有意义的、可测的相关作业、通常认为阶段是软件生存周期的 ~ 种细分，它常以下阶段开始之前的一个正式评审作为结束。
作业 (Tasks)	把要进行的工作分解为作业。一个作业是软件过程模中一个妥善定义的工作单元，它给管理部门提供一个对项目状态可视的检验点。作业有准备就绪准则(先决条件)和完成准则（后置条件）。 在过程定义的上下文中，一个作业是已定义过程的一个妥善定义的成分。所有的作业都可认为是活动，但并不是所有的活动都已完善地定义以致可认为是作业（虽然活动可以包括作业）。正因为如此，在等级 2 上避免使用“作业”一词，而采用不太严格的术语“活动”。
活动 (Activities)	一个活动是为了实现某个目的所采取的任何步骤或所完成的任何功能。既可以是脑力的也可以是体力的。活动包括经理和技术人员

	为了完成项目和组织的作业所作的全部工作。
软件工作产品（项目结果） Software work Products (project results)	活动和作业的结果主要由软件工作产品组成。软件工作产品是作为定义、维护或使用一个软件过程的一部分所生成的任何人工制品，包括过程描述、计划、规程、计算机程序及其相连的文档。可以打算也可以不打算将它们交付给顾客或最终用户。 工作产品成为过程中下一个步骤的输入，或者提供有关软件项目的档案信息供未来项目使用。 软件工作产品的例子有：计划、估计、有关实际工作的数据、纠正措施的文档和需求文档。其中可交付给顾客或最终用户的软件工作产品的子集称作软件产品。
软件产品 Software products	软件产品是指定交付给顾客或最终用户的一套完整的计算机程序、规程、以及相连的文档和数据，或者其中的任何独立的项、[IEEE—STD—610] 所有的软件产品均是软件工作产品。可是不交付给顾客或最终用户的软件工作产品不是软件产品。

4.3.4 项目定义软件过程和软件开发计划间的关系

一般项目定义软件过程的描述仍不够明确以致不能直接执行。虽然它一般标识出诸如角色（即谁去执行一个作业）和执行作业所需的软件工作产品的类型等事项，但它并不规定承担该角色的个人、将建立的具体的软件工作产品、也不规定执行作业和活动的进度。

项目软件开发计划在项目定义软件过程（将做什么及将如何做）和项目具体完成方式（例如哪一个人将按照什么样的进度生产哪种软件工作产品）之间建立联系，该开发计划可以是单个文件，也可以是一些计划的集合，总起来称为软件开发计划。项目定义软件过程和其软件开发计划一起使得实际执行该过程成为可能。

4.3.5 生在周期和 CMM

关键实践并不意味着限制软件生存周期的选择。已多次使用某一个特定软件生存周期的人可能会在关键实践的组织和结构中察觉到该生存周期的元素。可是，并不存在鼓励或阻碍使用任何特定软件生存周期的企图。

术语“阶段”用来指示软件项目工作的一种已定义的划分，但它并不与任何特定的软件生存周期相联系。正如在关键实践中所用的那样，“阶段”可以指严格的序列阶段，也可以指部分重叠的 overlapping 和累接（iterative）的阶段。

4.3.6 技术和 CMM

关键实践既不要求也不反对特定的软件技术——诸如建立原型，面向对象设计，重用软件需求、设计、代码、或其它成分。

4.3.7 文档和 CMM

关键实践描述若干过程一有关文档，每一个复盖特定的内容范围。关键实践并不要求在关键实践中命名的文档与组织或项目的实际工作产品间建立一一对应关系。也不打算对 DOD 所规定的文档或诸如 DOD—STD - 2167A 或 IEEE 软件标准等标准建立一一对应关系。关键实践仅仅要求这些文档的适当的内容是组织或项目的书面工作产品的一部分。

从文档结构方面来看 ,关键实践中所提及的某个文档的内容可能是一个更大的文档的一部分。例如，一个组织可以有一个包括软件风险管理计划要点的软件开发计划。

另一方面，关键实践中所提及的某个文档的内容可能分散在一组文档中，而此组文档不同于该关键实践中所命名的那组文档。例如，一个项目可以编制三个文档：软件开发计划、软件管理计划和项目工作分解结构，以便满足有关一个软件项目的软件风险管理、软件质量保证及软件开发计划等多个关键实践。

4.3.8 过程数据的收集和分析

有关过程数据的收集和分析的关键实践经过各成熟度等级逐渐进化。

在等级 2，数据主要与项目工作产品的规模、工作量和进度有关，并对每个项目分别加以定义、收集和存储。数据经过非正式机制在项目间共享。

在等级 3，每个项目有一个已定义的软件过程，它是组织标准软件过程经剪裁后的版本。与每个项目定义软件过程有关的数据被采集和存储在组织的软件过程数据库中。虽然所采集和存储的数据可能因项目而异，但在组织的软件过程数据库中，它们是很好定义了的。

在等级 4，基于组织标准软件过程，组织确定一组标准的测量。所有项目均采集这组标准的测量数据，同时还采集其它的项目一专有的数据，并将它们存储在组织的软件过程数据库中。项目使用这些数据去定量地了解项目定义软件过程的过程性能，并用这些数据使过程性能稳定。组织也使用这些数据去建立组织标准软件过程的过程能力基线。

在等级 5，数据用于选择技术和过程的改进区域，规划这些改进，并评价这些改进对组织过程能力的影响。

4.4 组织机构和角色

虽然 CMM 力图不依赖任何特定的组织机构和模型 ,但是在阐述 CMM 的实践时必须一致地使用与某种组织机构和角色有关的术语 ,而它们可能不同于任何具体组织所用的组织机构和角色。以下几节描述在解释 CMM 关键实践时所必需的与组织、项目及角色有关的各种概念。

4.4.1 组织的角色

一个角色是一项已定义的职责，它可能安排给一个或多个个人。在关键实践中常用以下角色的描述：

经理（manager）	经理履行的任务包括：对在经理职责范围内进行作业和活动的个人
-------------	-------------------------------

	提供技术和管理方面的指导并进行控制。经理的传统职能包括在其职责范围内进行策划、资源分配、组织、指导和控制。
高级经理 (senior manager)	<p>级经理在组织的足够高的层次上履行管理任务，主要关注组织的长期生存力，而不是短期的项目和合同关心的事项及压力。一般讲，一个高级工程经理负责多个项目。高级经理也提供和保护用于软件过程长期改进的资源（例如，软件工程过程组）。</p> <p>高级管理者，当用在 CMM 中时，能表示满足上述描述的任何经理，直到且包括整个组织的头。当用在关键实践中，术语高级管理者应该按关键过程区域及考虑中的项目和组织的上下文加以解释。为实现关键过程区域的目标必须完成一些十分重要的领导和监督任务；这里的意图就是要特别包括那些履行这些十分重要的任务所必须的高级经理。</p>
项目经理 (Project manager)	<p>项目经理履行的任务是对整个项目的总体业务负责；项目经理是指导、控制、管理和调整项目进行构造软件或硬件 / 软件系统工作的个人，项目经理是最终向顾客负责的个人。</p> <p>在面向项目的组织机构中，工作在同一个项目的大多数人应向项目经理报告，尽管某些科目可能有一个矩阵式的报告关系。而在矩阵式的组织机构中，可能仅业务职员只向项目经理报告，而工程组负有一个矩阵式的报告关系。</p>
项目软件经理 project soft - ware manager	<p>项目软件经理履行的任务是对项目的全部软件活动总负责。项目软件经理是项目经理按软件约定与其打交道的个人，且是控制一个项目的全部软件资源的个人。</p> <p>一个项目的软件工程组应向项目软件经理报告，尽管某些活动——例如工具开发——可以有一个矩阵式的报告关系。</p> <p>在大型项目中，项目软件经理多半是二线、三线或四线经理。在小型项目或只有单个项目的部门中，项目软件经理可能是 ~ 线软件经理或可能在更高层次</p>
一线软件经理 first-line software manaae	<p>一线软件经理履行的任务是对由软件工程师和其它有关人员组成的一个机构单位（例如，一个部门或者项目组）的人员配置和活动负直接管理责任（包括提供技术方向及对人员和薪金进行管理的功能）。</p>
软件作业领导 software taSK leadere	<p>一个软件作业领导履行的任务是充当特定作业的技术组的领导者，负有技术责任并向工作在该作业的职员提供技术方向。</p> <p>软件作业领导通常和其它工作在该作业的人员一样向同一个一统软件经理报告。</p>
职员、软件工程职员、 个人 / staff,sOftware \ \ Staff,individuals /	<p>在 CMM 的不同的关键实践中描述了不同的技术角色，在 CMM 中采用几个术语去指示履行这些不同技术任务的个人。职员是包括作业领导的一些个人，他们负责完成一项指派的功能，例如软件开发或软件配置管理，但他们不是经理。</p> <p>软件工程职员是软件技术人员（例如分析员、程序员和工程师），包括软件作业领导，他们进行项目的软件开发和维护活动，但他们不是经理。术语“个人”，当用在关键实践中时，由该词出现处的上下文来定性和限定（例如参与管理软件子合同的个人）</p>

对于其它的工程组，例如系统工程组或系统测试组，也能对其角色作出类似的分解。

在一个具体的项目或组织中，在这些角色和个人之间不必一一对应。一个人能完成多个角色，或者一个角色可由若干个人承担。

例如，对于一个小的纯软件项目，一个人可以承担多到六个角色：系统工程的一线经理，项目的系统工程经理，软件的一线经理、项目的软件经理、项目经理和软件配置管理经理。对于略大一点的项目，一个人可以担当：系统工程的一线经理、项目的系统工程经理和项目经理，而另一个人可以既是一线软件经理又是项目的软件经理。这两位经理可以在同一个二线机构中，也可以在不同的二线机构中。

对于大项目，许多岗位，特别是管理岗位，多半由不同的个人承担。

4.4.2 组织机构

必须充分理解有关组织、项目和组的基本概念，以便恰当地解释能力成熟度模型的关键实践。以下段落定义这些概念在 CMM 中的用法：

组织 (Organization)	一个组织是一个公司或其它实体（例如政府机构或军种）内的一个单位，在其内部将许多项目作为一个整体加以管理。在一个组织中的所有项目共有个相同的顶层经理并遵守共同的方针
项目 (Project)	项目是一项要求共同努力的任务，其目标是开发和（或）维护一个具体的产品。产品可以包括硬件、软件和其它成分。一般项目有它自己的投资、成本统计和交付时间表。
组 (Group)	组是负责一组作业或活动的部门、经理和个人的集会。组的规模可以变化：从 ~ 个受指派的习日全日制的单个人，或几个从不同部门指派来的非全日制的个人，直到几个全日制的个人。

下面描述在 CMM 中通常提及的组：

软件工程组 Software engineering group	软件工程组是负责一个项目的软件开发和维护活动（即需求分析、设计、编码和测试）的个人（既有经理又有技术人员）的集团。进行软件一有关工作的组，例如软件质量保证组、软件配置管理组和软件工程过程组，不在软件工程组之列。这些组都是“其它软件一有关组”。
软件一有关组 Software-related groups	软件一有关组是代表一种软件工程科目的个人（既有经理又有技术人员）的集团，它支持但不直接负责软件开发和（或）维护。软件工程科目的例子包括软件质量保证和软件配置管理。
软件工程过程组 Software engineering process group	软件工程过程组是由专家组成的组，他们推进组织所采用的软件过程的定义、维护和改进工作。在关键实践中，这个组通常指“负责组织的软件过程活动的组”。
系统工程组 System engineering group	系统工程组是负责下列工作的个人（既有经理又有技术人员）的集团：规定系统需求；将系统需求分配给硬件、软件和其它成分；规定硬件、软件和其它成分之间的界面；以及监控这些成分的设计和开发以保证它们符合其规格说明。
系统测试组	系统测试组是一些负责策划和完成独立的软件系统测试的个人（既

(System test group)	有经理又有技术人员) 的集团，测试的目的是为了确定软件产品是否满足对它的要求。
软件质量保证组 Software quality Assurance group	软件质量保证组是一些计划和实施项目的质量保证活动的个人 (既有经理又有技术人员) 的集团，其工作的目的是保证软件过程的步骤和标准得到遵守。有关软件质量保证的机构问题，在 4.4.3 节中讨论。
软件配置管理组 Software Configuration management group	软件配置管理组是一些负责策划、协调和实施软件项目的正式配置管理活动的个人 (既有经理又有技术人员) 的集团。
培训组 (Training group)	培训组是一些负责协调和安排组织的培训活动的个人 (既有经理又有技术人员) 的集团。通常这个组准备和讲授大多数的培训课程并且协调其它培训方式的使用。

4.4.3 组织机构的独立性问题

为使那些要求独立性的关键实践得到恰当的解释和遵循，组织必须十分小心。特别对小的项目都小的组织更是如此。当技术上或机构上的偏差可能影响有关项目的质量或风险时，关键实践要求独立性。例如，两个涉及独立性的实践是；

- ✍✍ SQA 组有 ~ 个独立于项目经理、项目软件工程组和其它软件一有关组的向高级管理者报告的渠道 (软件质量保证中的约定 1 . 2)。
- ✍✍ (统和验证) 测试用例和测试规程由独立于软件开发者的测试组进行规划和准备 (软件产品工程中活动 7 . 3)。

需要系统测试和验证测试的独立性是基于技术上的考虑。该独立性保证测试者不会受到软件开发者或维护者所作的设计决策和实施决策的不恰当的影响。

软件质量保证组的独立性是必须的，这样才能使其成员的工作能不受进度和成本压力的影响。没有机构上的独立性要保证有效的操作上的独立性是困难的。例如，即使存在严重的不符合问题，一个要向项目经理报告的雇员也可能不愿意停止测试活动。

组织必须确定这样的组织机构，它将在组织的战略经营目标和经营环境的上下文中支持要求独立性的活动，例如 SQA。

独立性应该：

- ✍✍ 给执行 SQA 任务的个人提供组织上的自由度，使其能成为高级管理者了解项目的“耳目”。
- ✍✍ 执行 SQA 任务的个人，使其免受被监督项目管理者所作的性能评价的影响，
- ✍✍ 使高级管理者相信正在报告的有关项目过程和产品的信息是客观的。

由于关键实践允许对独立性准则作解释，在确定关键过程区域的目标是否已达到时组织必须运用专业判断。

4.5 运用专业判断

使得所提供的一组完整的正确的原则能运用到广泛的情况，对某些关键实践有意阐述得允许有灵活性。在关键实践中广泛采用了不确定的短语如“ 受影响的组 ”、“ 当合适时 ”和“ 当需要时 ”。在关键实践中通常已经努力做到最少使用这种不确定的术语，而且在许多情况下提供了例子，至少在第一次使用该术语时是如此。但是，对于两个不同的组织，对于同一组

组织中的两个项目，或者一个项目在其生存周期的不同点上，这些短语可能有不同的含义。因此，每个项目或组织必须就其具体情况澄清这些短语。

澄清这些短语要求组织考虑使用它们时的整体的上下文。一个贴切的提问是对这样一个短语的具体解释是否满足该关键过程区域的目标。在确定目标是否已满足时必须采用专业判断。附录 B 中的术语汇编在解释关键实践中的这些短语或其它短语时能提供指导。

当解释关键实践和它们如何对关键过程区域的目标作贡献时也必须使用专业判断。一般讲，关键过程区域描述所有软件组织均应该具有的 ~ 组基本的行大，而不管他们的规模或他们的产品如何。可是必须根据项目的或组织的经营环境或具体情况解释 CMM 中的关键实践。

这种解释应该既基于对 CMM 的深刻了解，也基于对组织和其项目的丰富知识。关键过程区域的目标提供了构造这个解释的一个方法。如果一个组织关键过程区域的实施方法能满足该区域的目标，但它们明显地不同于一般意义下的关键实践，那么解释的理论基础必须用文档记载。已文档化的理论基础将帮助评估或评价组理解为什么某些实践以他们现在的方式实施。

运用专业判断导致软件过程的“化度 { goodness.}”问题。尽管在许多软件环境中 CMM 确实建立了用于判断一个“合理”过程的最低限度的准则，但它并不对软件过程提出“优度”要求。过程管理的目的是建立能使用的过程，并且该过程能作为基于组织的经营需求进行系统改进的基础。

用于判断一个“合理”软件过程的准则是什么呢？一个合理的软件过程是这样的过程，它在建立组织的能力方面是有效的，并且满足已定义过程的大多数要求。具体地讲，它是实用的、已文档化的、已实施的、经培训的、已测量的和能改进的。

如果一个组织建立一个用于估计的软件过程，而该估计由掷骰子组成，那么它是一个合理的过程吗？可以肯定它能文档化并得到一致遵循。某些人甚至说它和许多其它估计技术一样实际。可是大多数软件专业人员不会认为“掷骰子”是一个合理的估计过程。因为它仅仅遵从概率定律，它不可能被改进。

“掷骰子”和另一个过程“去问乔治”之间差别有多大？后者可能是一个非常好的估计方法。只要乔治在周围，估计甚至是一致的和可重复的。可是它也不能满足我们的准则，因为其它的个人不可能受到此方法的培训。这正是一个以人为中心的过程，没有乔治过程就不可能重复。它不能建立一种前进的组织能力。

通常认为采用 Delphi 方法（此方法是有关某专题的专家评审考虑中的议题，并且就有关此议题的建议形成一致意见）的某种变形去作估计是合理的软件过程。基于 DelPhi 方法的规模估计方法满足用以判断一个合理、有效过程的准则，尽管 Delnhi 方法也是一个以人为中心的过程。组织的能力可以建立在诸如 Detohi 方法等结构化技术之上。

在根本意义上说，为了作出这样的区分，专业判断是必须的。困难在于分辨出符合性和优度。目标概括关键实践，而关键实践又描述合理的软件过程。可是符合一个合理过程并不意味着该过程在实现其目的上是有效的。可能有许多因素影响组织和项目的成功。例如某些成功的项目生产出的产品无人购买，这在商界是个失败。

“优度”属性只能在经营环境和具体的项目和组织情况下加以解释。只有组织能作出这样的“优度”判断，这种判断是组织连续过程改进周期的一部分。因为永远不会实现完美元缺，因此连续过程改进决不会终止。

等级 2 关键实践

需求管理

等级 2（可重复的）的一个关键过程区域

需求管理的目的是在顾客和将处理顾客需求的软件项目之间建立对顾客需求的共同理解。

需求管理包括和顾客一起建立和维护有关软件项目需求的协议，该协议称作“分配给软件的系统需求”。“顾客”可解释为系统工程组、销售组、另一个内部组织、或者一个外部顾客。协议既包括技术需求、又包括非技术需求（例如交付日期）。该协议形成估计、策划和跟踪整个软件生存周期内软件项目活动的基础。

将系统需求分配给软件、硬件和其它系统成分的工作可能由软件工程组之外的组（如系统工程组）完成，软件工程组可能对此分配无直接控制。在项目约束范围内，软件工程组采取恰当步骤以保证对分配给软件的需求建档、并加以控制，该组负责处理分配给软件的系统需求。

为实现此控制，软件工程组评审初始的和经修改的分配给软件的系统需求，以便在它们被纳入软件项目之前使问题得以解决。每当改变分配给软件的系统需求时，都要调整受到影响的软件计划，工作产品和活动，使其与更新后的需求保持一致。

目标

目标 1 分配给软件的系统需求是受控的，建立供软件工程和管理使用的基线。

目标 2 软件计划、产品和活动与分配给软件的系统需求保持一致。

执行约定

约定 1 项目遵循一书面的、组织上的方针去管理分配给软件的系统需求。

在这些实践中，分配给软件的系统需求称为“分配需求”（allocated requirements）。

分配需求是系统需求的子集，它将用系统的软件成分来实现。分配需求是软件开发计划的主要输入。软件需求分析详细描述和提炼分配需求，并生成文档化的软件需求。

该方针一般规定：

1. 对分配需求建立文档。
2. 由下列人员评审分配需求：

✍ 软件经理。

✍ 其它受到影响的组。

受到影响的组的例子有：

- 系统测试组，
- 软件工程组（包括所有小组，例如软件设计小组），
- 系统工程组，
- 软件质量保证组，

- 配置管理组，和
- 文档支持组。

3·更改软件计划、工作产品和活动，以便和分配需求的改变保持一致。

执行能力

能力 1

对每个项目，建立分析系统需求并将其分配到硬件、软件和其它系统成分的职责。
分析和分配系统需求不是软件工程组的职责，而是他们工作的先决条件。

该职责包括：

- 1、在项目整个生存期内，管理系统需求和它们的分配，并对其建立文档。
- 2、实现对系统需求及其分配的更改。

能力 2

对分配需求建立文档。

分配需求包括：

- 1、影响和确定软件项目活动的非技术性需求（即：协议、条件、和（或）合同条款）。
协议、条件和合同条款的例子包括：

- 要交付的产品，
- 交付日期，和
- 里程碑。

- 2、对软件的技术需求。

技术需求的例子有；

- 最终用户、操作员、支持、或集成功能；
- 性能要求；
- 设计约束；
- 编程语言；和
- 界面需求。

- 3、用于确认软件产品满足分配需求的验收准则。

能力 3

提供足够的用以管理分配需求的资源和投资。

- 1．指派在应用领域和软件工程方面有经验和技能的个人去管理分配需求。
- 2．使得支持管理需求活动的工具是可用的。

支持工具的例子有：

- 电子表格程序，
- 配置管理工具，
- 跟踪工具，和
- 测试管理工具。

能力 4

软件工程组和其它软件一有关组的成员受到培训以便完成他们的需求管理活动。

培训的例子包括：

- 项目所使用的方法、标准、规程
- 应用领域。

执行的活动

活动 1

在分配需求被纳入软件项目之前，软件工程组评审它们。

1、鉴别出不完整的和遗漏的分配需求。

2、评审分配需求，确定它们是否：

✍✍ 用软件来实现是可行的和恰当的，

✍✍ 被清晰和正确地阐述，

✍✍ 是相互一致的，和

✍✍ 是可测试的。

3、负责分析和分配系统需求的组评审任何被识别出是有潜在问题的分配需求，并作出必要的更改。

4、和受到影响的组协商由分配需求引起的约定。

受到影响的组的例子包括：

——软件工程组（包括所有的小组，例如软件设计小组），

——软件估计组，

——系统工程组，

——系统测试组，

——软件质量保证组，

——软件配置管理组，

——合同管理组，和

——文档支持组。

参考软件项目策划关键过程区域的活动 6，以便找到包括协商约定的实践。

活动 2

软件工程组采用分配需求作为软件计划、工作产品和活动的基础。分配需求：

1、进行管理和控制。

“进行管理和控制”意味着在给定时间（过去或现在）使用的工作产品的版本是已知的（即版本控制），而且以受控的方式引进更改（即更改控制）。

如果希望有比“进行管理和控制”所蕴含的更加正规的手续，则工作产品可置于配置管理的完备的纪律之下，正如在软件配置管理关键过程区域中所描述的。

2、软件开发计划的基础。

3、制定软件需求的基础。

活动 3

审时分配需求的更改，将其纳入软件项目。

1、估它对现有约定的影响，合适时协商更改。

✍✍ 对组织外部的个人和组所作约定的更改由高级管理者参与评审。

参考软件项目策划关键过程区域活动4 和软件项目跟踪和监督关键过程区域活动3 以便找到包括对组织外部所作约定的实践。

✍✍ 和受到影响的组协商组织内部约定的更改。

参考软件项目跟踪和监督关键过程区域的活动 5、6、7 和 8，以便找到包括对约定的更改进行协商的实践。

2、由于分配需求的更改所造成的对软件计划、工作产品和活动必须作的更改要加以：

✍✍ 识别，

✍✍ 评价，

✍✍ 风险评估，

✍✍ 文档化，

✍✍ 规划，

✍✍ 传达到受到影响的组和个人，和

✍✍ 跟踪直到结束。

测量和分析

测量 1 进行测量并将测量结果用以确定对分配需求的管理活动状态。

78

测量的例子包括；

一每个分配需求的状态；

一关于分配需求的更改活动；和

一对分配需求更改的累积数，包括所建议的、未解决的、已批准的和已并入系统基线的总数。

验证实施

验证 1

高级管理者参与定期评审那些管理分配需求的活动。

高级管理者定期评审的主要目的是保证在合适的抽象层次上并以及时的方式了解和洞察软件过程。评审间隔应该满足组织的需要，只要存在获得例外情况报告的合适机制，间隔可以长。

参考软件项目跟踪和监督关键过程区域的验证 1，以便找到包括高级管理者监督评审的典型内容的实践。

验证 2

项目经理既定期地又事件一驱动地参与评审那些管理分配需求的活动。

参考软件项目跟踪和监督关键过程区域的验证 2 以便找到包括项目管理者监督评审的典型实践。

验证 3

软件质量保证组评审和（或）审计管理分配需求的活动和工作产品，并报告其结果。

参考软件质量保证关键过程区域。

至少，这些评审和（或）审计要验证：

- 1．分配需求是已评审的，且在软件工程组承担它们之前，问题已经解决。
- 2．当分配需求更改时，软件计划、工作产品和活动已经合适地修改。
- 3．由分配需求的更改所导致的对约定的更改已与受影响的组协商过。

软件项目策划

等级 2（可重复的）的一个关键过程区域

软件项目策划的目的是为完成软件工程和管理软件项目制定合理的计划。

软件项目策划包含估计待完成的工作，建立必要的约定和确定进行该工作的计划。

软件策划首先作出有关待完成的工作和其它定义及界定软件项目的约束和目标（由用来管理关键过程区域的实践所建立的）的陈述。软件策划过程包括以下步骤：估计软件工作产品规模及所需的资源，制定时间表，鉴别和评估软件风险和协商约定。为了制定软件计划（即软件开发计划），可能需要重复地通过这些步骤。

该计划提供完成和管理软件项目活动的基础，并按照软件项目的资源、约束和能力，阐述对软件项目的顾客作的约定。

目标

目标 1

对供策划和跟踪软件项目用的软件估计已建立文档。

目标 2

软件项目的活动和约定是有计划的并已建立文档。

目标 3

受影响的组和个人同意他们的关于软件项目的约定。

执行约定

约定 1 指定项目软件经理负责协商约定和制定项目软件开发计划。

约定 2 项目遵循书面的组织的用于策划软件项目的方针。

该方针一般规定：

1、配给软件的需求用作为策划软件项目的基础。

参考需求管理关键过程区域的活动 2。

2、下列人员之间协商软件项目的约定：

项目经理，

项目软件经理，和

其它的软件经理。

3、其它的工程组协商他们介入该软件活动的事宜，并记入文档。

其它工程组的例子包括：

—系统工程组，

—硬件工程组，和

—系统测试组。

4、影响的组评审软件项目的；

- ✍✍ 软件规模估计，
- ✍✍ 工作量和成本估计，
- ✍✍ 进度，和
- ✍✍ 其它约定。

受影响的组的例子有；

- 一软件工程组（包括所有的小组，例如软件设计小组），
- 一软件估计组，
- 一系统工程组，
- 一系统测试组，
- 一软件质量保证组，
- 一软件配置管理组，
- 一合同管理组，和
- 一文档支持组。

5、级管理者评审所有的对组织外部的个人和组所作的软件项目约定。

6、项目的软件开发计划进行管理和控制。

在所有这些实践中术语“软件开发计划”习惯上指的是管理软件项目的全面的计划。采用“开发”术语并不是有意排除软件维护项目或软件支持项目，应在每个项目的上下文中恰当地解释该术语。

“进行管理和控制”意味着在给定时间（过去或现在）使用的工作产品的版本是已知的（即版本控制），而且以受控的方式引进更改（即更改控制）。

如果希望有比“进行管理和控制”所蕴含的更高程度的控制，则工作产品可置于配置管理的完备的纪律之下，正如在软件配置管理关键过程区域中所描述的。

执行能力

能力 1

对软件项目存在文档化的经批准的工作陈述。

1、作陈述包括；

- ✍✍ 工作的范围，
 - ✍✍ 技术目标和对象，
 - ✍✍ 对顾客和最终用户的识别，
- 这些实践中所指的最终用户是顾客指定的最终用户或最终用户的代表。
- ✍✍ 要求实施的标准，
 - ✍✍ 安排的职责，
 - ✍✍ 成本和进度的约束及目标，
 - ✍✍ 软件项目和其它组织间的依赖关系。

其它组织的例子有；

- 一顾客，
 - 一子承包商。和
 - 一合资伙伴。
- ✍✍ 资源限制和目标，和
 - ✍✍ 对开发和（或）维护的其它的约束和目标。

2、作陈述由下列人员评审；

- ✍✍ 项目经理，
- ✍✍ 项目软件经理，
- ✍✍ 其它软件经理，和
- ✍✍ 其它受影响的组。

3、工作陈述进行管理和控制。

能力 2

✍✍ 安排制定软件开发计划的职责。

1、目软件经理，直接工作或通过委托代表，协调项目的软件策划。

2、可追踪，可说明的方式分解对软件产品和活动的职责，并将其安排给软件经理。

软件工作产品的例子包括；

- 一适当时，交付给外部在客或最终用户的工作产品；
- 一供其它工程组使用的工作产品；和
- 一供软件工程组内部使用的主要工作产品。

能力 3

为策划软件项目提供足够的资源和投资。

1、可能处，可以使用对正在策划的软件项目的应用领域有专门知识的有经验的个人来制定软件开发计划。

2、得支持软件项目策划的工具合用。

支持工具的例子有：

- 一电子表格程序，
- 一估计模型，和
- 一项目策划和调度程序。

能力 4 介入软件策划的软件经理、软件工程师和其它个人，在适用于其职责范围的软件估计和策划规程方面受到培训。

执行的活动

活动 1

软件工程组参加项目建议群组。

1、件工程组与下列各项工作有关：

- ✍✍ 建议的准备和提交，
- ✍✍ D 说明的讨论和提交，和
- ✍✍ 对影响软件项目的约定作更改而进行的协商。

2、件工程组评审所建议的项目约定。

项目约定的例子包括：

- 一项目的技术目标和对象；
- 一系统和软件的技术解；
- 一软件预算、进度和资源；和
- 一软件标准和规程。

活动 2

✍✍ 在整个项目策划的早期阶段起动软件项目策划，此两项策划平行进行。

活动 3

✍✍ 在项目的整个生存期内，软件工程组和其它受影响的组一起参加整个项目的策划。

1、件工程组评审项目层的计划。

活动 4

高级管理者参加按照已文档化的规程评审对组织外部的个人和组所作的软件项目约定。

活动 5

识别或确定具有可管理规模的预先规定阶段的软件生存周期。

软件生存周期的例子有：

- 瀑布型，
- 重叠瀑布型，
- 螺旋型，
- 单型构造，和
- 单个原型 / 重叠瀑布型。

活动 6

按照已文档化的规程制定项目的软件开发计划。

该规程一般规定：

1、件开发计划基于且遵守；

顾客的标准，当合适时；

项目的标准；

经批准的工作陈述；和

分配需求。

2、软件一有关组和其它工程组协商他们介入软件工程组活动的计划，把该项支持工作编入预算，并对协议建立文档。

软件一有关组的例子有：

- 软件质量保证组，
- 软件配置管理组，和
- 文档支持组。

其它工程组的例子有：

- 系统工程组，
- 硬件工程组，和
- 系统测试组。

3、其它软件一有关组和其它工程组协商软件工程组介入其活动的计划，把该支持工作编入预算，并对协议建立文档。

4、下列人员评审软件开发计划：

- 项目经理，
- 项目软件经理，
- 其它软件经理，和
- 其它受影响的组。

5、软件开发计划进行管理和控制。

活动 7

对有关软件项目的计划建立文档。

在关键实践中，这个计划或计划的集合称为软件开发计划。

参考软件项目跟踪和监督关键过程区域的活动 1 以便找到关于项目使用软件开发计划的实践。

软件开发计划包括：

- 1、件项目的目的、范围、目标、和对象。
- 2、件生存周期的选择。
- 3、选的供开发和（或）维护软件用的规程、方法和标准的确定。

软件标准和规程的例子有；

- 一软件开发策划，
- 一软件配置管理，
- 一软件质量保证，
- 一软件设计，
- 一问题跟踪和解决，和
- 一软件测量。

4、开发软件工作产品的确定。

5、软件工作产品的规模估计和对软件工作产品的更改。

6、软件项目的工作量和成本的估计。

7、键计算机资源的预计使用情况。

8、件项目的进度，包括里程碑和评审的确定。

9、目软件风险的识别和评估。

10、于项目软件工程设施和支持工具的计划。

活动 8

识别为建立和保持对软件项目的控制所必须的工作产品。

参考软件配置管理关键过程区的活动 4。

活动 9

按照巴文档化的规程导出对软件工作产品规模(或对软件工作产品规模的更改)的估计。

该规程一般规定：

1、所有主要的软件工作产品和活动作出规模估计。

对软件规模的量度的例子有：

- 一功能点 (functionpoints)，
- 一特征点 (featurepoints)，
- 一代码行，
- 一需求数，和
- 一页数。

要作规模估计的工作产品和活动的类型的例子有；

- 一运行软件和支持软件，
- 一可交付的和不交付的工作产品，
- 一软件和非软件工作产品（例如文档），和
- 一开发、验证和确认工作产品的活动。

2、软件工作产品分解到能满足估计对象所需要的粒度。

3、可得到历史数据的地方使用历史数据。

4、规模估计的假定记入文档。

5、规模估计建立文档，进行评审、并使之得到承认。

评审和认可规模估计的组和个人的例子包括：

- 一项目经理，
- 一项目软件经理，和
- 一其它软件经理。

活动 10

按照巴文档化的规程导出对软件项目的工作量及成本的估计。

该规程一般规定：

1、软件项目的工作量及成本的估计与对软件工作产品的规模（或更改的规模）的

估计有关。

2、生产率数据（历史的和（或）当前的）可利用时，将其用于估计；将这些数据的来源及合理的解释记入文档。

✍✍ 当可能时，生产率和成本数据来自该组织的项目。

✍✍ 生产率和成本数据应考虑用于制作软件工作产品的工作量及重要的成本。

用于制作软件工作产品的重要成本的例子有：

—直接的劳务费，

—管理费，

—差旅费，和

—计算机使用成本。

3、工作量、人员配置和成本的估计基于过去的经验。

✍✍ 当可能时，应利用类似项目的经验。

✍✍ 导出各种活动的时间分段（timephasing）。

✍✍ 作出工作量，人员配置和成本估计在软件生存周期上的分布。

4、计和导出估计值时所作的假定记入文档，进行评审、并使之得到认可。

活动 11 按照巴文档化的规程导出对项目的关键计算机资源的估计。

关键计算机资源可以在宿主环境中，在集成与测试的环境中，在目标环境中，或在以上这些环境的任何组合中。

该规程一般规定：

1、别项目的关键计算机资源。

关键计算机资源的例子包括：

—计算机存储能力，

—计算机处理器运用能力，和

—通信通迢容量。

2、关键计算机资源的估计与对下列项的估计有关：

✍✍ 软件工作产品的规模，

✍✍ 运行处理的负载（operationalprocessingload），和

✍✍ 通信量。

3、关键计算机资源的估计建立文档、进行评审、并使之得到认可。

活动 12 按照已文档化的规程导出项目的软件进度表。

该规程一般规定：

1、件进度与以下各项有关：

✍✍ 对软件工作产品的规模（或规模更改）的估计，和

✍✍ 软件工作量和成本。

2、件进度表基于过去的经验。

✍✍ 可能时，利用类似项目的经验。

3、件进度表适应所规定的里程碑日期、关键的相关性日期及其它限制。

4、件进度表中的活动有合适的间隔，且里程碑是以适当的时间长度分开的，以支持在进程测量上的精度。

5、导出进度表时的假定记入文档。

6、软件进度表建立文档、进行评审、并使之得到认可。

活动 13

对与项目成本资源、进度和技术方面相联系的软件风险进行鉴别、评估和建立文档。

1、于风险对项目的潜在影响，对风险进行分析和优先级排序。

2、别风险的偶发事件。

偶发事件的例子包括：

- 进度受阻，
- 更换人员配置计划，和
- 关于附加计算装置的更换计划。

活动 14

制定关于项目软件工程设施和支持工具的计划。

1、这些设施和支持工具的能力需求的估计建立在对于软件工作产品的规模估计和其它特征的基础上。

软件开发设施和支持工具的例子包括：

- 软件开发用的计算机和外设，
- 软件测试用的计算机和外设，
- 目标计算机环境软件，和
- 其它支持软件。

2、购买或研制这些设施和支持工具问题，分配职责和商谈约定。

3、有受到影响的组评审该计划。

活动 15

记录软件策划数据。

1、记录的信息包括估计及为了重构该估计和评估它们的合理性所需要的相关信息。

2、软件策划数据进行管理和控制。

测量和分析

测量 1

进行测量，并将测量结果用于确定软件策划活动的状态。

测量的例子有：

- 与计划相比较，软件项目策划活动的里程碑的完成情况，和
- 与计划相比较，在软件项目策划活动中所完成的工作，所用的工作量和所消耗的资金。

验证实施

验证 1

高级管理者定期参与评审软件项目策划的活动。

高级管理者定期评审的主要目的是在合适的抽象层次上并以及时的方式了解和洞察软件过程活动。评审间隔应满足组织的需要，只要已存在报告例外情况的合适机制，间隔可以长。

- 1、审技术、成本、人员配置和进度等性能。
- 2、析在较低层次上未解决的矛盾和问题。
- 3、析软件项目风险。
- 4、排和评审措施条款并跟踪到结束。
- 5、备每次会议的摘要报告，并将其散发给受影响的组和个人。

验证 2

项目经理既定期地也事件驱动地参与评审软件项目策划的活动。

- 1、影响的组有代表出席。
- 2、照软件项目的工作陈述和分配需求，评审软件项目策划活动的状态和当前结果。
- 3、析组间的依赖关系。
- 4、析较低层次上未解决的矛盾和问题。
- 5、审软件项目风险。
- 6、配和评审措施条款，并跟踪到结束。
- 7、备每次会议的摘要报告，并将其散发给受影响的组和个人。

验证 3

软件质量保证组评审和（或）审计软件项目策划的活动和工作产品，并报告其结果。

参考软件质量保证关键过程区域。

至少，评审和审计要查证：

- 1、件估计和策划的活动。
- 2、制定项目约定的活动。
- 3、备软件开发计划的活动。
- 4、于准备软件开发计划的标准。
- 5、件开发计划的内容。

软件项目跟踪和监督

等级 2（可重复的）的一个关键过程区域

软件项目跟踪和监督的目的是建立对实际进展的适当的可视性,使管理者能在软件项目性能明显偏离软件计划时采取有效措施。

软件项目跟踪和监督包括对照已文档化的估计、约定、和计划评审和跟踪软件完成情况和结果。基于实际的完成情况和结果调整这些计划。

软件项目的已文档化的计划（即软件开发计划，正如在软件项目策划关键过程区域中所描述的）用作跟踪软件活动、传送状态和修订计划的基础。管理者监控软件活动。主要通过所选出的软件工作产品完成时和在所选择的里程碑处，将实际的软件规模、工作量、成本和时间表与计划相比较，来确定进展情况。当确定未实现软件项目计划时，采取纠正措施。这些措施可以包括修订软件开发计划以反映实际的完成情况和重新策划遗留的工作或者采取改进性能的措施。

目标

目标 1

服软件计划，跟踪实际结果和性能。

目标 2

当实际结果和性能明显偏离软件计划时，采取纠正措施并加以管理直到结束。

目标 3

对软件约定的更改得到受到影响的组和个人的认可。

执行约定

约定 1

指派一个项目软件经理，对项目的软件活动和结果负责。

约定 2

项目遵循书面的组织用以管理软件项目的方针。

该方针一般规定：

- 1、用并维护一个已文档化的软件开发计划作为跟踪软件项目的基础。
- 2、时向项目经理报告软件项目的状态和问题。
- 3、软件计划未实现时，采取纠正措施，或者调整性能，或者调整计划。
- 4、受影响的组参与和认可的情况下对软件约定进行更改。

受影响的组的例子有；

- 一软件工程组（包括所有小组，例如软件设计小组），
- 一软件估计组，
- 一软件工程组，

- 系统测试组，
- 软件质量保证组，
- 软件配置管理组，
- 合同管理组，和
- 文档支持组。

5、级管理者评审所有的约定更改和软件项目对组织外部的个人和组所作的新的约定。

执行能力

能力 1

对软件项目的软件开发计划已建立文档和批准。

参考软件项目策划关键过程区域的活动 6 和 7 以便找到包括软件开发计划的实践。

能力 2

项目软件经理明确地分配关于软件工作产品和活动的任务。

分配的任务包括：

- 1、开发的软件工作产品或将提供的服务。
- 2、些软件活动的工作量和成本。
- 3、些软件活动的进度。
- 4、些软件活动的预算。

能力 3

提供足够的用以跟踪软件项目的资源和投资。

- 1、派给软件经理和软件作业领导跟踪软件项目的具体职责。
- 2、得支持软件跟踪的工具是可用的。

支持工具例子包括：

- 电子表格程序，和
- 项目策划 / 制定进度表程序。

能力 4

在管理软件项目的技术和人员方面，软件经理受到培训。

培训的例子包括；

- 管理技术项目，
- 跟踪和监督软件规模、工作量、成本及进度，和
- 管理职员。

能力 5

一线软件经理在软件项目的技术方面受到定向培训。

走向培训的例子包括：

- 项目的软件工程标准和规程，和
- 项目的应用领域。

执行的活动

活动 1

将已文档化的软件开发计划用于跟踪软件活动和传送状态。

参考软件项目策划关键过程区域的活动 7 以便找到包括软件开发计划内容的实践。

这个软件开发计划：

- 1、着工作进展而更新，以使反映完成情况，特别当实现里程碑时。
- 2、能容易被下列人员使用：

- ✍✍ 软件工程组（包括所有的小组，例如软件设计小组），
- ✍✍ 软件经理，
- ✍✍ 项目经理，
- ✍✍ 高级管理者，和
- ✍✍ 其它受影响的组。

活动 2

✍✍ 按照已文档化的规程修订项目的软件开发计划。

参考软件项目策划关键过程区域的活动 6 以便找到包括制定软件开发计划的活动的实践。

该规程一般规定：

- 1、合适时，修订软件开发计划，以便包括对计划的精炼和更改，特别当计划有重大更改时。需将在分配给软件的系统需求、设计约束、资源、成本和进度之间的相关性反映在所有计划更改中。
- 2、新软件开发计划以便包括所有新的软件项目约定和对约定的更改。
- 3、每次修订时，评审软件开发计划。
- 4、软件开发计划进行管理和控制。

“进行管理和控制”意味着在给定时间（过去或现在）使用的工作产品的版本是已知的（即版本控制），而且以受控的方式引进更改（即更改控制）。

如果希望有比“进行管理和控制”所蕴含的更高程度的控制，则工作产品可置于配置管理的完备的纪律之下，正如在软件配置管理关键过程区域中所描述的。

活动 3

级管理者参与按照已文档化的规程评审那些对组织外部的个人和组所作的软件项目约定和约定的更改。

活动 4

将经批准的、影响软件项目约定的更改传达给软件工程组和其它软件一有关组的成员。

其它软件一有关组的例子包括：

- 一软件质量保证组，
- 一软件配置管理组，和
- 一文档支持组。

活动 5

跟踪软件工作产品的规模（或者软件工作产品更改的规模），必要时采取纠正措施。

参考软件项目策划关键过程区域的活动 9 以便找到包括导出规模估计的实践。

1. 踪所有主要软件工作产品的规模（或更改的规模）。
2. 实际代码规模（生成的、经完全测试的和交付的）和在软件开发计划中所记载的估计规模对比。
3. 实际交付的文档单元数据与在软件开发计划中所记载的估计数相比较。
4. 期精炼、监控和调整软件工作产品的整体预测规模（与实际值相结合的估计值）。
5. 受影响的组协商那些能影响软件约定的对软件工作产品规模估计的更改，并对该更改建立文档。

活动 6

跟踪项目的软件工作量和成本，必要时采取纠正措施。

参考软件项目策划关键过程区域的活动 10 以便找到包括导出成本估计的实践。

1. 根据一段时间内已完成的工作得出的实际工作量和成本的开销与在软件开发计划中的估计量相比较以识别潜在的超支和欠支。
2. 跟踪软件成本并将其与软件开发计划中记载的估计相比较。
3. 工作量及人员配置与软件开发计划中记载的估计相比较。
4. 受影响的组协商那些影响软件约定的在人员配置和其它软件成本方面的更改，并对该更改建立文档。

活动 7

跟踪目的关键计算机资源，必要时采取纠正措施。

参考软件项目策划关键过程区域的活动 11 以便找到包括导出计算机资源估计的实践。

1. 对于每一个主要的软件成分，正如在软件开发计划中记载的那样，跟踪项目关键计算机资源的实际使用情况和预计使用情况，并将其与估计相比较。
2. 和受影响的组协商那些影响软件约定的对关键计算机资源估计的更改，并对其建立文档。

活动 8

跟踪项目的软件进度，必要时采取纠正措施。

参考软件项目策划关键过程区域的活动 12 以便找到包括导出进度的实践。

1. 将软件活动、里程碑和其它约定的实际完成情况与软件开发计划作比较。
2. 评价软件活动、里程碑和其它约定等迟后和提前完成的后果对将来的活动和里程碑的影响。
3. 和受影响的组协商那些影响软件约定的对软件进度的修订，并对其建立文档。

活动 9

跟踪软件工程技术活动，必要时采取纠正措施。

1. 软件工程组的成员定期向他们的一线经理报告他们的技术状态。
2. 将为后续工作所提供的软件发行内容与软件开发计划中记载的计划相比较。
3. 在任何软件工作产品中所识别出的问题均予以报告和建立文档。
4. 问题报告直至结束。

活动 10

跟踪与项目的成本、资源、进度及技术方面有关的软件风险。

参考软件项目策划关键过程区域的活动 13 以便找到包括风险识别的实践。

1. 当有补充信息时，调整风险及风险可能性的优先级。
2. 项目经理定期参与评审高风险的区域。

活动 11

记录软件项目的实际测量数据和重新策划的数据。

参考软件项目策划关键过程区域的活动 15 以便找到包括记录项目数据的实践。

1. 记录的信息包括估计、及为重构该估计和验证其合理性所必须的辅助信息。
2. 对软件重新策划的数据进行管理和控制。
3. 将软件策划数据、重新策划数据和实际测量数据归档供正在进行的和未来的项目使用。

活动 12

件工程组进行定期的内部评审以便对照软件开发计划跟踪技术进展、计划、性能和问题。这些评审由下列人员共同进行：

- 1、一线软件经理和他们的软件作业领导。

- 2、当合适时，项目软件经理、一线软件经理和其它软件经理。

活动 13

按照已文档化的规程在所选择的项目里程碑处进行正式评审以评价软件项目的完成情况和结果。

这些评审：

- 1、被安排在软件项目进度的有意义的点上进行，例如在所选阶段的开头或结束处。
- 2、合适时，由雇客、最终用户和组织内部受影响的组参与进行。

这些实践中所指的最终用户是顾客指定的最终用户或最终用户的代表。

- 3、使用经过负责的软件经理评审和批准的材料。
- 4、分析软件活动的约定、计划和状态：
- 5、导致对重大问题、措施条款和决策的识别和建立文档。
- 6、分析软件项目风险。
- 7、导致在必要时对软件开发计划的改进。

测量和分析

测量 1

进行测量并将测量结果用于确定软件跟踪和监督活动的状态。

测量的例子包括：

- 在完成跟踪和监督活动时所化资的工作量和其它资源；和
- 对软件开发计划的更改活动，‘包括对以下项的更改：软件工作产品的规模估计，软件成本估计、关键计算机资源估计和进度。

验证实施

验证 1

高级管理者定期参与评审软件项目跟踪和监督和活动。

高级管理者定期评审的主要目的是在合适的抽象层次上并以及时的方式了解和洞察软件过程活动。评审间隔应该满足组织的需要，只要已存在报告例外情况的合适机制，间隔可以长。

- 1、评审技术、成本、人员配置和进度等性能。
- 2、分析较低层次上无法解决的矛盾和问题。
- 3、分析软件项目风险。
- 4、安排和评审措施条款，并跟踪到结束。
- 5、准备每次会议的摘要报告，并将其散发给受影响的组。

验证 2

项目经理既定期地又事件码区动地参与评审软件项目跟踪和监督的活动。

- 1、受影响的组有代表出席。
- 2、对照软件开发计划评审技术、成本、人员配置和进度等性能。
- 3、评审关键计算机资源的使用；对照原来的估计，报告有关这些计算机资源的当前估计和实际使用。
- 4、分析组间的依赖关系。
- 5、分析在较低层次上无法解决的矛盾和问题。

- 6、分析软件项目风险。
- 7、分配和评审措施条款，并跟踪到结束。
- 8、准备每次会议的摘要报告，并将其散发给受影响的组。

验证 3

软件质量保证组评审和（或）审计软件项目跟踪和监督的活动和工作产品，并报告其结果。

参考软件质量保证关键过程区域。

至少，该评审和（或）审计要查证：

- 1、评审和修正约定的活动。
- 2、修正软件开发计划的活动。
- 3、经修正的软件开发计划的内容。
- 4、跟踪以下各项的活动：软件项目的成本、进度、风险、技术和设计限制、及功能性和性能。
- 5、已计划好的技术评审和管理评审的活动。

软件子合同管理

等级 2（可重复的）的一个关键过程区域

软件子合同管理的目的是选择合格的软件子承包商并有效地管理他们。

软件子合同管理包括选择软件予承包商、建立和子承包商的约定，及跟踪和评审子承包商的性能和结果。这些实践包括对纯软件子合同的管理，也包括对子合同的软件成分的管理，后者含有软件、硬件和可能有的其它系统成分。

基于子承包商完成工作的能力选择子承包商。许多因素对将主承包商工作的一部分签为子合同的决策产生影响。选择子承包商可以基于战略经营同盟及技术上的考虑。这个关键过程区域的实践阐述与将工作的一个确定部分签子合同给另一个组织相联系的传统的采购过程。

当签子合同时，建立一个包括技术和非技术（例如交付日期）需求的已文档化的协议并将其用作管理于合同的基础。对将由于承包商完成的工作和关于该工作的计划建立文档。子承包商将遵循的标准和主承包商的标准一致。

子承包商的软件的策划、跟踪和监督活动由于承包商完成。主承包商确保恰当地完成这些策划、跟踪和监督活动并且确保子承包商交付的软件产品满足其验收准则。主承包商和子承包商一起工作去管理他们的产品和过程界面。

目标

目标 1

主承包商选择合格的软件予承包商。

目标 2

主承包商和软件子承包商认同他们相互的约定。

目标 3

主承包商和软件子承包商保持不断的通信。

目标 4

主承包商对照约定跟踪软件予承包商的实际结果和性能。

执行约定

约定 1

项目遵循书面的管理软件予合同的组织上的方针。

该方针一般规定；

1、在选择软件予承包商和管理软件子合同时采用已文档化的标准和规程。

2、合同协议形成管理干合同的基础。

3、对子合同的更改需由主承包商和子承包商共同介入和认同。

约定 2

指派一个子合同经理负责建立和管理软件子合同。

1、子合同经理本人在软件工程方面是博学的和有经验的，或者他已配备有具有这方面知识和经验的个人。

2、子合同经理和受影响的当事人一起负责协调待签子合同工作的技术范围及子合同的条件和条款。

项目的系统工程组和软件工程组确定待签子合同工作的技术范围。适当的经营功能组一例如采购、财务和法律组，建立和监控于合同的条款和条件。

3、子合同经理负责：

✍✍ 选择软件予承包商，

✍✍ 管理软件予合同，和

✍✍ 安排子合同产品的后一子合同支持。

执行能力

能力 1

✍✍ 为选择软件予承包商和管理于合同提供足够的资源和投资。

1、确定软件经理和其它个人管理子合同的具体职责。

2、使得支持管理子合同的工具可用。

支持工具的例子包括：

一估计模型，

一电子表格程序，和

一项目管理和编制进度程序。

能力 2

参与建立及管理软件予合同的软件经理和其它个人受到完成这些活动的培训。

培训的例子的有：

一准备和策划软件予合同的签订，

一评价子合同投标者的软件过程能力，

一评价子合同投标者的软件估计和计划，

一选择子承包商，和

一管理子合同。

能力 3

参与管理软件子合同的软件经理和其它个人在子合同的技术方面接受定向培训。

走向培训的例子有：

一应用领域，

一正运用的软件技术，

一正使用的软件工具，

一正使用的方法论，

一正使用的标准，和

一正使用的规程。

执行的活动

活动 1

按照已文档化的规程定义和规划待签子合同的工作。

该规程一般规定：

1、基于对项目的技术特征和非技术特征所作的综合评估，选择待签子合同的软件产品和活动。

✍✍ 选择待签子合同的功能或子系统，使其与潜在于承包商的技能和能力相匹配。

✍✍ 基于对系统和软件需求所作的系统性分析和恰当的划分，确定对待签子合同的软件产品和活动的规格说明。

2、从项目的以下各项导出对待签子合同工作的规格说明和将遵循的标准和规程：

✍✍ 工作陈述，

✍✍ 分配给软件的系统需求，

✍✍ 软件需求，

✍✍ 软件开发计划，和

✍✍ 软件标准和规程。

3、对子合同的工作陈述进行：

✍✍ 准备，

✍✍ 评审，

✍✍ 认同，

评审和认同子合同工作陈述的个人的例子包括：

一项目经理，

一项目软件经理，

一应负责的软件经理，

一软件配置管理经理，

一软件质量保证经理，和

一子合同经理

✍✍ 当需要时修订，和

✍✍ 管理和控制。

“进行管理和控制”意味着在给定时间（过去或现在）使用的工作产品的版本是已知的（即版本控制），而且以受控的方式引进更改（即更改控制）。

如果希望有比“进行管理和控制”所蕴含的更高程度的控制，则工作产品可置于配置管理的完备的纪律之下，正如在软件配置管理关键过程区域中所描述的。

参考软件项目组织关键过程区域的能力1以便找到包括工作陈述的典型内容的实践。

4、在准备子合同的工作陈述的同时准备一份选择子承包商的计划，合适时要进行评审。

活动2

按照已文档化的规程，在评价子合同投标者完成该工作的能力的基础上选择软件子承包商。

这个规程包括对以下各项的评价：

1、所提交的对计划中的子合同的建议。

2、以前在类似工作方面的性能记录，如果有的话。

3、子合同投标者的组织相对于主承包商的地理位置。

对某些子合同的有效管理可能要求频繁的面对面的相互交流。

4、软件工程和软件管理能力。

评价子承包商能力的一个例子是 SEI 软件能力评价方法。

5、为完成该工作可得到职员。

6、以前在类似应用领域的经验，包括子承包商的软件管理队伍所具有的软件专门知识。

7、可用资源。

资源的例子包括

- 设施，
- 硬件，
- 软件，和
- 培训。

活动 3

将主承包商和软件子承包商间的合同协议用作管理子合同的基础。

该合同协议用文档记载以下各项：

- 1、条款和条件。
- 2、工作陈述。
- 参考软件项目策划关键过程区域的能力 1 以便找到包括工作陈述的典型内容的实践。
- 3、对待开发产品的需求。
- 4、子承包商和主承包商之间的依赖关系表。
- 5、将交付给主承包商的子合同产品。

产品的例子有：

- 源代码，
- 软件开发计划，
- 仿真环境，
- 设计文档，和
- 验收测试计划。

- 6、提交产品修订的条件。
- 7、在主承包商接收子合同产品之前评价子合同产品时将用的验收规程和验收准则。
- 8、主承包商用来监控和评价子承包商性能的规程和评价准则。

活动 4

主承包商评审和批准已文档化的子承包商软件开发计划。

- 1、这个软件开发计划包括（直接地或通过引证）主承包商软件开发计划中的适当条款。

在某些情况下，主承包商的软件开发计划可能包括于承包商的软件开发计划，因此无需单独的子承包商软件开发计划。

参考软件项目策划关键过程区域的活动 7 以便找到包括项目软件开发计划的内容的实践。

活动 5

将已文档化的且经批准的子承包商软件开发计划用于跟踪软件活动和通信状态。

活动 6

按照已文档化的规程判定对软件子承包商的工作陈述、子合同条款和条件、以及其它约定的更改。

- 1、该规程一般规定，主承包商和子承包商的所有受影响的组都要参与。

活动 7

主承包商的管理者和软件子承包商的管理者一起执行定期的状态或协调评审。

- 1、当合适时，向子承包商提供对产品顾客及最终用户的需要及希望的可视性。这些实践中所指的最终用户是顾客指定的最终用户或最终用户的代表。
- 2、对照子承包商软件开发计划评审子承包商的技术、成本、人员配置和进度等性能。
- 3、评审那些指定为对项目是至关重要的计算机资源；跟踪子承包商对当前估计所起的作用，并将它与在子承包商软件开发计划中所记载的每一个软件成分的估计相比较。

- 4、分析在子承包商的软件工程组和其它子承包商组之间的关键的依赖关系和约定。
 - 5、分析主承包商和子承包商之间的关键的依赖关系和约定。
- 既评审子承包商对主承包商的承诺，也评审主承包商对于承包商的承诺。
- 6、分析不符合于合同的问题。
 - 7、分析与子承包商工作有关的项目风险。
 - 8、分析不能由于承包商内部解决的矛盾和问题。
 - 9、安排和评审措施条款，并跟踪到结束。

活动 8

软件于承包商参与定期技术评审和交流。

这些评审；

- 1、当合适时，向子承包商提供对顾客和最终用户的需要和希望的可视性。
- 2、监控于承包商的技术活动。
- 3、验证子承包商对技术要求的解释和实施是符合主承包商的要求的。
- 4、验证约定正得到满足。
- 5、验证技术问题是以及及时的方式得到解决。

活动 9

按照已文档化的规程在所选择的里程碑处进行正式评审，评价子承包商的软件工程完成情况和结果。

这些规程一般规定：

- 1、评审是预先计划好的，并在工作陈述中加以记载。
- 2、评审分析子承包商关于软件活动的约定、计划和状态。
- 3、识别重大的问题、措施条款及决策，并将它们记入文档。
- 4、分析软件风险。
- 5、当合适时，精炼子承包商的软件开发计划。

活动价主承包商的软件质量保证组按照已文档化的规程监控于承包商的软件质量保证活动。

该规程一般规定：

- 1、定期地评审子承包商用于软件质量保证的计划、资源、规程和标准，以保证它们适于监控于承包商的性能。
- 2、执行对子承包商的常规评审以保证经批准的规程和标准正得到遵循。
✍ 主承包商的软件质量保证组抽查子承包商的软件工程活动和产品。
✍ 当合适时，主承包商的软件质量保证组审计于承包商的软件质量保证记录。
- 3、定期审计于承包商有关其软件质量保证活动的记录，以评估软件质量保证计划、标准和规程正受到遵循的好坏。

活动们主承包商的软件配置管理组按照已文档化的规程监控于承包商的软件配置管理活动。

该规程一般规定：

- 1、评审子承包商关于软件配置管理的计划、资源、规程和标准，以保证它们是满足要求的。
- 2、主承包商和子承包商就有关软件配置管理的情况协调它们的活动，以保证子承包商的产品能容易地集成到或并入主承包商的项目环境。
- 3、定期地审计于承包商的软件基线库，以便评估用于软件配置管理的标准和规程正受到遵循的程度和它们在管理软件基线方面如何有效。

活动 12

主承包商按照已文档化的规程进行验收测试，这是子承包商软件产品交付的一部分。
该规程一般规定：

1、在测试前，主承包商和子承包商共同定义、评审和批准对每个产品的验收规程和验收准则。

2、对验收测试的结果建立文档。

3、对任何未通过其验收测试的产品制定措施计划。

活动 13

定期评价软件予承包商的性能，并与子承包商一起评审该评价工作。

对于承包商性能的评价提供一个机会使子承包商能获得关于它是否正在满足其雇客(即主承包商)需求的反馈信息。诸如性能有奖评审(Performance award fee reviews)那样的机制提供这种类型的反馈，它与贯穿于整个项目的定期的协调和技术评审不同。这些评价的文档也作为将来子承包商选择活动的输入。

测量与分析

测量 1

进行测量，将测量结果用于确定管理软件予合同的活动状态。

测量的例子有：

—管理子合同活动的成本与计划相比较，

—子合同产品的实际交付日期与计划相比较，和

—主承包商的交付物交给子承包商的实际日期与计划相比较。

验证实施

验证 1

高级管理者定期参与评审管理软件予合同的活动。

高级管理者定期评审的主要目的是在合适的抽象层次上并以及时的方式了解和洞察软件过程活动。评审间隔应满足组织的需要，只要已存在报告例外情况的合适机制，间隔可以长。

参考软件项目跟踪和监督关键过程区域的验证 1 以便找到包括高级管理者监督评审的典型内容的实践。

验证 2

项目经理既定期地也事件驱动地参与评审管理软件子合同的活动。

参考软件项目跟踪和监督关键过程区域的验证 2 以便找到包括项目管理者监督评审的典型内容的实践。

验证 3

软件质量保证组评审(或)审计管理软件于合同的活动和工作产品，并报告其结果。

参考软件质量保证关键过程区域。

至少，该评审和(或)审计要查证；

1、选择子承包商的活动。

2、管理软件予合同的活动。

3、协调主承包商和子承包商的配置管理活动的活动。

4、已计划好的和子承包商一起进行的评审的执行情况。

- 5、用以确立完成子合同的关键项目里程碑或阶段的那些评审的执行情况。
- 6、子承包商软件产品的验收过程。

软件质量保证

等级 2（可重复的）的一个关键过程区域

软件质量保证的目的是向管理者提供适当的对软件项目正使用的过程和正构造产品的可视性。

软件质量保证包括评审和审计软件产品和活动以验证它们符合适用的规程和标准，给项目和其它有关的经理提供这些评审和审计的结果。

在软件项目的早期阶段，软件质量保证组与软件项目一起工作制定计划、标准和规程等，这些计划、标准、和规程将增加软件项目的价值并将满足项目和组织方针上的限制。通过参与制定计划、标准和规程，软件质量保证组帮助确保它们适合项目的需要，并且帮助验证它们对完成整个软件生存周期中的评审和审计将是适用的。软件质量保证组在整个生存周期评审项目活动，审计软件工作产品，并就软件项目是否正遵守已制定的计划、标准和规程等给管理者提供可视性。

首先在软件项目内部处理符合性问题，如可能的话就地解决它。对于那些无法在软件项目内部解决的问题，软件质量保证组逐级上递该问题到管理者的恰当层次以求得解决。

这个关键过程区域只包括该组履行软件质量保证功能的实践。而识别软件质量保证组要评审和（或）审计的具体的活动和工作产品的实践一般包含在其它关键过程区域的验证实施共同特点中。

目标

目标 1

软件质量保证活动是有计划的。

目标 2

软件产品和活动遵守适用的标准、规程和需求的情况得到客观的验证。

目标 3

受影响的组和个人接到软件质量保证活动和结果的通知。

目标 4

高级管理者处理在软件项目内部不能解决的不符合问题。

执行约定

约定 1

项目遵循书面的实施软件质量保证（SQA）的组织方针。

该方针一般规定；

1、对全部软件项目，SQA 功能到位。

2、SQA 有一个向高级管理者报告的渠道，它独立于：

✍✍ 项目经理，

✍ 项目的软件工程组，和

✍ 其它的软件一有关组。

其它软件一有关组的例子有：

一软件配置管理组，和

一文档支持组。

组织必须确定一种组织机构，它在组织的战略经营目标和经营环境的上下文中支持那些要求独立性的活动，例如 SQA。

独立性应该：

一给担当 SQA 角色的个人提供组织上的自由度，使他们成为高级管理者在软件项目上的“耳目”。

一使得担当 SQA 角色的个人免受他们正在评审的软件项目的管理者所作的性能评价的影响。

一使高级管理者相信正在报告的有关项目过程和产品的信息是客观的。

3、高级管理者定期地评审 SQA 活动和结果。

执行能力

能力 1

存在负责协调和实施项目的 SQA 的组（即 SQA 组）。

一个组是负责一组作业或活动的部门、经理、和个人的集合。组的规模可以变化，从一个受指派的非全日制的单个人，到几个从不同部门指派来的非全日制的个人，到几个全日制的个人。建立一个组时应考虑的因素包括指派的作业和活动、项目的规模、组织机构和组织文化。某些组，例如软件质量保证组，集中注意力于项目活动，而其它组，例如软件工程过程组，则集中关注全组织的活动。

能力 2

为进行 SQA 活动提供足够的资源和投资。

1、指派一个经理专门负责项目的 SQA 活动。

2、指派一个在 SQA 任务方面是博学的，并有权力采取适当的监督行动的高级经理接收和处理软件不符合问题。

在 SQA 向高级经理报告链上的全部经理均是在 SQA 的任务、责任和权力方面有见识的。

3、使得支持 SQA 活动的工具合用。

支持工具的例子有：

一工作站，

一数据库程序，

一电子表格程序，和

一审计工具。

能力 3

SQA 组的成员受到培训以完成他们的 SQA 活动。

培训的例子有：

一软件工程技巧和实践；

一软件工程组和其它软件一有关组的岗位任务及职责；

一用于软件项目的标准、规程和方法；

一软件项目的应用领域；

- SQA 的对象。规程和方法；

- SQA 组对软件活动的参与；
- SQA 方法和工具的有效使用；和
- 人员间的交流。

能力 4

软件项目的成员接受有关 SQA 组的任务、职责、权力和价值等的定向培训。

执行的活动

活动 1 按照已建档的规程为软件项目制订 SQA 计划。

该规程一般规定：

- 1、SQA 计划的制定是在整个项目策划的早期阶段，并平行于整个项目策划。
- 2、受影响的组和个人评审该 SQA 计划。

受影响的组及个人的例子有：

- 项目软件经理；
 - 其它软件经理；
 - 项目经理；
 - 顾客的 SQA 代表；
 - SQA 组对其报告不符合问题的高级经理；和
 - 软件工程组（包括全部小组，诸如软件设计小组及软件作业领导）。
- 3、对 SQA 计划进行管理和控制。

“进行管理和控制”意味着在给定时间（过去或现在）使用的工作产品的版本是已知的（即版本控制），而且以受控的方式引进更改（即更改控制）。

如果希望有比“进行管理和控制”所蕴含的更高程度的控制，则工作产品可置于配置管理的完备的纪律之下，正如在软件配置管理关键过程区域中所描述的。

活动 2

按照 SQA 计划进行 SQA 组的活动。

该计划包括：

- 1、SQA 组的职责和权力。
- 2、SQA 组的资源要求（包括职员、工具和设施）。
- 3、项目的 SQA 组活动的进度表和投资。
- 4、SQA 组参加制定项目的软件开发计划、标准和规程的情况。
- 5、将由 SQA 完成的评价。

待评价的产品和活动的例子有：

- 运行软件和支持软件，
- 可交付的和不交付的产品，
- 软件和非软件产品（例如文档），
- 产品开发和产品验证活动（例如运行测试用例），和
- 生成产品时所从事的活动。

- 6、将由 SQA 组进行的审计和评审。
- 7、将用作 SQA 组评审和审计的基础的项目的标准和规程。
- 8、用于对不符合性问题建立文档和进行跟踪直至结束的规程。

这些规程可能作为计划的一部分而纳入，也可以通过索引那些包含它们的其它文档的方式而纳入。

- 9、要求 SQA 组生成的文档。

10、就 SQA 活动给软件工程组和其它软件一有关组提供反馈信息的方法和频率。

活动 3

SQA 组参与准备和评审项目的软件开发计划、标准和规程。

L、SQA 就以下几个方面对计划、标准和规程提供咨询和评审：

- ✍ 对组织方针的符合性，
- ✍ 对外部强加的标准和要求的符合性（例如工作陈述所要求的标准），
- ✍ 适合项目使用的标准，
- ✍ 在软件开发计划中应阐述的专题，和
- ✍ 项目指定的其它领域。

2、SQA 组验证计划、标准和规程已到位并可用于评审与审计软件项目。

活动 4

SQA 组评审软件工程活动以验证符合性。

1、对照软件开发计划和指定的软件标准和规程去评价活动。

参考其它关键过程区域中的验证实施共同特点以便找到包括由 SQA 组进行特定评审和审计的实践。

2、对偏差进行鉴别和建立文档，并跟踪到结束。

3、验证纠正措施。

活动 5

SQA 组审计指定的软件工作产品以验证符合性。

1、在交付给顾客之前，评价可交付的软件产品。

2、对照指定的软件标准、规程和合同要求评价软件工作产品。

3、对偏差进行鉴别和建立文档，并跟踪到结束。

4、验证纠正措施。

活动 6

SQA 组定期向软件工程组报告其活动的结果。

活动 7

按照巴文档化的规程对在软件活动和软件工作产品中所鉴别出的偏差建立文档并加以处理。

该规程一般规定：

1、将不符合软件开发计划和指定的项目标准及规程的问题写成文档，并在可能处，与适当的软件作业领导、软件经理或项目经理一起，加以解决。

2、有些不符合软件开发计划和指定的标准及规程的问题不能与软件作业领导、软件经理或项目经理一起加以解决，将这些不符合问题写成文档并提交给指定的接收不符合问题的高级经理。

3、定期评审提交给高级经理的不符合问题直至解决它们为止。

4、对不符合问题的文档进行管理和控制。

活动 8

当合适时，SQA 组与顾客的 SQA 人员一起对它的活动和发现进行定期评审。

测量与分析

测量 1

进行测量并将测量结果用于确定 SQA 活动的成本和进度状态。

测量的例子有：

- SQA 活动的里程碑的完成情况与计划作比较；
- 在 SQA 活动中所完成的工作、所花费的工作量和所消耗的资金与计划作比较；和
- 产品审计和活动评审的次数与计划相比较。

验证实施

验证 1

高级管理者定期参与评审 SQA 活动。

高级管理者定期评审的主要目的是在合适的抽象层次上并以及时的方式了解和洞察软件过程活动。评审间隔应该满足组织的需要，只要已存在报告例外情况的合适机制，间隔可以长。

参考软件项目跟踪和监督关键过程区域的验证 1 以便找到包括高级管理者监督评审的典型内容的实践。

验证 2

项目经理既定期地也事件驱动地参与评审 SQA 活动。

参考软件项目跟踪和监督关键过程区域的验证 2 以便找到包括项目管理者监督评审的典型内容的实践。

验证 3

独立于 SQA 组的专家定期评审项目 SQA 组的活动和软件工作产品。

软件配置管理

等级 2（可重复的）的一个关键过程区域

软件配置管理的目的是建立和维护在项目的整个软件生存周期中软件项目产品的完整性。

软件配置管理包括标识在给定时间点上软件的配置(即选定的软件工作产品及其描述),系统地控制对配置的更改、并维护在整个软件生存周期中配置的完整性和可跟踪性。置于软件配置管理之下的工作产品包括交付给顾客的软件产品(例如软件需求文档和代码),以及与这些软件产品等同的产品项或生成这些软件产品所要求的产品项(例如编译程序)建立一个软件基线库,当软件基线形成时就将它们纳入该库。通过软件配置管理的更改控制和配置审计功能,系统地控制基线的更改和那些利用软件基线库构造成的软件产品的发行。

这个关键过程区域仅包括实施软件配置管理功能的实践。而标识具体的配置项或单元的实践则包含在描述每个配置项或单元的开发和维护的关键过程区域中。

目标

目标 1

软件配置管理活动是有计划的。

目标 2

所选定的软件工作产品是已标识的、受控的和适用的。

目标 3

对已标识的软件工作产品的更改是受控的。

目标 4

受影响的组和个人得到软件基线的状态和内容的通知。

执行约定

约定 1

项目遵循书面的用以实施软件配置管理(SCM)的组织方针。

该方针一般规定;

- 1、明确指派每个项目的 SCM 职责。
- 2、在项目的整个生存周期内实行 SCM。
- 3、对于对外可交付的软件产品、指定的内部软件工作产品和指定在项目内部使用的支持工具(例如编译器)都实行 SCM。

4、项目建立或可以利用一个仓库,用来存储配置项/单元和相关联的 SCM 记录。

在这些实践中这个仓库的内容称为“软件基线库”。

存取该仓库的工具和规程在这些实践中称为“配置管理库系统”。

置于配置管理之下并作为单个实体予以处理的工作产品称为配置项。配置项一般分解为

配置成分，而配置成分一般分解为单元。在一个硬件 / 软件系统中，所有的软件可看成一个单个配置项，或者可将该软件分解为多个配置项。在这些实践中术语“配置项牌元”用于指示在配置管理下的元素。

5、定期审计软件基线和 SCM 活动。

执行能力

能力 1

存在或者建立一个有权力管理项目软件基线的委员会（即软件配置控制委员会—SCCB）。

该 SCCB：

- 1、审定软件基线的建立和配置项 / 单元的标识。
- 2、代表项目经理和所有可能受到软件基线更改影响的组的利益。
受影响的组的例子有；
 - 硬件质量保证组，
 - 硬件技术状态（配置）管理组，
 - 硬件工程组，
 - 制造工程组，
 - 软件工程组（包括所有的小组，例如软件设计小组），
 - 系统工程组，
 - 系统测试组，
 - 软件质量保证组，
 - 软件配置管理组，
 - 合同管理组，和
 - 文档支持组。
- 3、评审和审定对软件基线的更改。
- 4、审定由软件基线库制造的产品的生成。

能力 2

存在负责协调和实施项目的 SCM 的组（即 SCM 组）。

一个组是负责一组作业或活动的部门、经理、和个人的集合。组的规模可以变化：从一个受指派的非全日制的单个人，到几个从不同部门指派来的非全日制的个人，到几个全日制的个人。建立一个组时应考虑的问题有：指派的作业和活动、项目的规模、组织机构和组织文化。某些组，例如软件质量保证组，集中注意力于项目的活动，而其它组，例如软件工程过程组，集中关注全组织的活动。

SCM 组协调或实现：

- 1、项目的软件基线库的生成和管理。
- 2、SCM 计划、标准和规程的制定、维护和散发。
- 3、将置于 SCM 之下的软件工作产品集会的标识。
一个工作产品是由定义、维护、或使用一个软件过程所生成的任何人工制品。
- 4、对存取软件基线库的管理。
- 5、软件基线的更新。
- 6、由软件基线库制造的产品的生成。
- 7、SCM 行动的记录。
- 8、SCM 报告的生成和散发。

能力 3

为进行 SCM 活动提供足够的资源和投资。

- 1、安排一个经理专门负责 SCM。
- 2、使得支持 SCM 活动的工具合用。

支持工具的例子有：

- 工作站，
- 数据库程序，和
- 配置管理工具。

能力 4

SCM 组的成员在有关进行其 SCM 活动的对象、规程和方法方面受到培训。

培训的例子包括：

- SCM 标准、规程和方法；和
- SCM 工具。

能力 5

软件工程组和其它软件一有关组的成员受到培训以便完成其 SCM 活动。

其它软件一有关组的例子有：

- 软件质量保证组，和
- 文档支持组。

培训的例子有：

- 在软件工程组和其它软件一有关组的内部进行 SCM 活动要遵循的标准、规程和方法，和
- SCM 组的角色、职责和权力。

执行的活动

活动 1

按照已文档化的规程对每个软件项目准备一份 SCM 计划。

这个规程一般规定：

- 1、SCM 计划的制定是在整个项目策划的早期阶段，并平行于整个项目策划。
- 2、受影响的组评审 SCM 计划。
- 3、对 SCM 计划进行管理和控制。

“进行管理和控制”意味着在给定时间（过去或现在）使用的工作产品的版本是已知的（即版本控制），而且以受控的方式引进更改（即更改控制）。

如果希望有比“进行管理和控制”所蕴含的更高程度的控制，则工作产品可置于配置管理的完备的纪律之下，正如在本关键过程区域中所描述的。

活动 2

用已文档化的经批准的 SCM 计划作为进行 SCM 活动的基础。

该计划包括：

- 1、将进行的 SCM 活动、活动的日程表、指派的职责和所要求的资源（包括职员、工具和计算机设施）。
- 2、SCM 需求和将由软件工程组及其它软件一有关组进行的 SCM 活动。

活动 3

建立一个配置管理库系统作为软件基线的仓库。

该库系统：

1、支持 SCM 的多个控制层次。

导致多个控制层次的情况例如：

—在生存周期的不同时间所需要的控制层次不同（例如，随着产品成熟要更加严密的控制），

—纯软件系统和既包括硬件又包括软件的系统所需要的控制层次不同。

2、提供对配置项 / 单元的存储和检索功能。

3、在受影响的组之间和在库内部的控制层次之间提供配置项 / 单元的共享和传送。

4、帮助使用配置项 / 单元的产品标准。

5、对配置项 / 单元的归档版本提供存储和恢复功能。

6、帮助保证由软件基线库制造的产品的正确生成。

7、对 SCM 记录提供存储、更新和检索功能。

8、支持 SCM 报告的编制。

9、提供对库结构和内容的维护。

库维护功能的例子有：

—库文件的备份 / 重建，和

—从库的错误中恢复。

活动 4

标识将置于配置管理之下的软件工作产品。

1、基于已文档化的准则选择配置项 / 单元。

可标识为配置项 / 单元的软件工作产品的例子有：

—过程—有关文档（例如：计划、标准或规程），

—软件需求，

—软件设计，

—软件代码单元，

~ 软件测试规程，

~ 为软件测试活动所构造的软件系统，

—为交付给顾客或最终用户所构造的软件系统，

—编译程序，和

—其它支持工具。

2、安排给每个配置项 / 单元唯一的标志符。

3、详细说明每个配置项 / 单元的特征。

4、详细说明每个配置项 / 单元所属于的软件基线。

5、详细说明在开发中将每个配置项 / 单元置于配置管理之下的时间点。

6、标识每个配置项 / 单元的负责人（即从配置管理的角度来说的所有者）。

活动 5

按照巴文档化的规程，起动、记录、评审、批准和跟踪对所有配置项或单元的更改请求和问题报告。

活动 6

按照已文档化的规程控制对基线的更改。

该规程一般规定：

1、进行评审和（或）回归测试以保证更改不会造成对基线的未料到的影响。

2、仅仅那些经 SCCB 批准的配置项 / 单元才能进入软件基线库。

3、以能保持软件基线库的正确性和完整性的方式进行配置项或单元的登入和退出。

登入或退出步骤的例子有；

- 验证修改是经审定的，
- 建立更改日志，
- 保持一份更改拷贝，
- 更新软件基线库，和
- 建立被取代的软件基线的档案。

活动 7

按照已文档化的规程生成由软件基线库制造的产品并控制它们的发行。

该规程一般规定：

- 1、SCCB 审定由软件基线库制造的产品的生成。
- 2、不论为内部使用或外部使用，由软件基线库制造的产品仅仅由软件基线库中的配置项或单元组成。

活动 8

按照已文档化的规程记录配置项或单元的状态。

该规程一般规定：

- 1、足够详细地记录配置管理行动，使每个配置项 / 单元的内容和状态，都是清楚的并且能恢复以前的版本。
- 2、对每个配置项 / 单元维护其当前状态并保留其历史（即更改和其它行动）。

活动 9

编制用文档记载 SCM 活动和软件基线内容的标准报告，并使受影响的组和个人可以使用它。

报告的例子包括：

- SCCB 会议记录，
- 更改申请的摘要和状态，
- 问题报告的摘要和状态（包括解决情况），
- 软件基线更改的摘要，
- 配置项 / 单元的修改历史，
- 软件基线状态，和
- 软件基线审计结果。

活动们按照已文档化的规程进行软件基线审计。

该规程一般规定：

- 1、为审计作好充分准备。
- 2、评估软件基线的完整性。
- 3、评审配置管理库系统的结构和设施。
- 4、验证软件基线库内容的完备性和正确性。
- 5、验证与适用的 SCM 标准和规程的符合性。
- 6、向项目软件经理报告审计结果。
- 7、跟踪得自审计的措施条款直至结束。

测量和分析

测量 1

进行测量并将测量结果用于确定 SCM 活动的状态。

测量的例子有：

- 每单位时间处理的更改申请数；

- 一 SCM 活动的里程碑的完成情况与计划相比较；和
- 一在 SCM 活动中所完成的工作、花费的工作量和消耗的资金。

验证实施

验证 1

高级管理者定期参与评审 SCM 活动。

高级管理者定期评审的主要目的是在合适的抽象层次上并以及时的方式了解和洞察软件过程活动。评审间隔应该满足组织的需要，只要已存在报告例外情况的合适机制，间隔可以长。

参考软件项目跟踪和监督关键过程区域的验证 1 以便找到包括高级管理者监督评审的典型内容的实践。

验证 2

项目经理既定期地也事件驱动地参加评审 SCM 活动。

参考软件项目跟踪和监督关键过程区域的验证 2 以便找到包括项目管理者监督评审的典型内容的实践。

验证 3

SCM 组定期审计软件基线以验证它们符合定义它们的文档。

验证 4

软件质量保证组评审和审计有关 SCM 的活动和工作产品，并报告其结果。

参考软件质量保证关键过程区域。

至少，评审和审计要验证；

1、以下各组对 SCM 标准和规程的依照情况：

✍✍ SCM 组，

✍✍ SCCB，

✍✍ 软件工程组，和

✍✍ 其它软件一有关组。

2、定期进行软件基线审计的情况。

等级 3 关键实践

组织过程焦点

等级 3（已定义的）的一个关键过程区域

组织过程焦点的目的是建立组织对软件过程活动的责任,这些活动能改进组织的整体软件过程能力。

组织过程焦点包括增进和保持对组织的和项目的软件过程的了解,协调那些评估、制定、维护和改进这些过程的活动。

组织提供长期的保证和资源,通过诸如软件工程过程组那样的组来协调那些横跨当前和未来软件项目的软件过程的制定和维护工作。这个组负责组织的软件过程活动,具体负责制定和维护组织的标准软件过程和有关的过程财富(正如在组织过程定义关键过程区域中所描述的),同时它和软件项目一起协调过程活动。

目标

目标 1

在整个组织中,过程的制定和改进活动是协调的。

目标 2

已标识出所使用的软件过程与一个过程标准相比的强处和弱点。

目标 3

组织层上过程的制定和改进活动是有计划的。

执行约定

约定 1

组织遵循书面的用以协调整个组织中的软件过程的制订和改进活动的组织方针。

该方针一般规定:

- 1、建立一个组,它负责组织层上的软件过程活动以及与项目协调这些活动。
- 2、定期评估项目所使用的软件过程以确定其强处和弱点。
- 3、项目所使用的软件过程是由组织的标准软件过程经适当剪裁而来。

参考集成软件管理关键过程区域的活动 1 以便找到包括剪裁组织标准软件过程的实践。

4、对于每个项目的软件过程、工具和方法的改进及其它有用的信息,其它项目都可以得到。

约定 2

高级管理者主办组织的软件过程制定和改进活动。

高级管理者;

- 1、对组织的职员和经理表明他对这些软件过程活动的承诺。
- 2、制定关于投资、人员配置和其它资源的长期规划和约定。
- 3、制定管理和实施有关过程制定和改进活动的策略。

约定 3

高级管理者监督组织的有关软件过程制定和改进的活动。

高级管理者：

- 1、保证组织的标准软件过程支持它的经营目标和战略。
- 2、就设置软件过程制定和改进的优先组问题提出建议。
- 3、参与制定用于软件过程制定和改进的计划。

✍✍ 高级管理者同较高层次的职员和经理协调软件过程需求和问题。

✍✍ 高级管理者与组织的经理协商以保证得到经理和职员的支持和参与。

执行能力

能力 1

存在负责组织的软件过程活动的组。

一个组是负责一组作业或活动的部门、经理和个人的集合。组的规模可以变化：从一个受指派的非全日制的单个个人，到几个从不同部门指派来的非全日制的个人，到几个全日制的个人。建立一个组时应考虑的方面包括指派的作业和活动、项目的规模、组织机构和组织文化。某些组，例如软件质量保证组，集中注意力于项目的活动，而其它的组，例如软件工程过程组，集中关注全组织的活动。

1、在可能处，该组配备有一个软件技术专业人员的核心，指定这些专业人员为该组的全日制职员，他们可能还要得到其它一些非全日制职员的支持。

该组的最常见的例子是软件工程过程组（SEPG）。

2、这个组的人员配置能代表软件工程学科和软件一有关学科。

软件工程和软件一有关学科的例子有：

- 一软件需求分析，
- ~ 软件设计，
- ~ 编码，
- ~ 软件测试，
- ~ 软件配置管理，和
- 一软件质量保证。

能力 2

提供足够的用于组织软件过程活动的资源和投资。

1、责成具有特定领域专长的有经验的个人承担支持该组的任务。

特定领域的例子有：

- 一软件重用，
- 一计算机辅助软件工程（CASE）技术，
- 一测量，和
- 一培训课程的编制。

2、支持组织软件过程活动的工具是合用的。

支持工具的例子有：

- 一统计分析工具，
- 一桌面出版工具，
- 一数据库管理系统，和
- 一过程建模工具。

能力 3

负责组织的软件过程活动的组员接受为完成这些活动所要求的培训。

培训的例子有；

- 一 软件工程实践，
- 一 过程控制技术，
- 一 组织的更改管理，
- 一 策划、管理和监控软件过程，和
- 一 技术转换。

参考培训大纲关键过程区域。

能力 4

软件工程组和其它软件一有关组的成员接受有关组织的软件过程活动及他们在这些 '活动中的角色的定向培训。

参考培训大纲关键过程区域。

执行的活动

活动 1

定期地评估软件过程，制定处理评估发现的行动计划。一般每隔 1—1 年半到每隔三年进行一次评估。评估应察看组织中所用的全部软件过程，但可以通过采样过程区域和项目来完成它。

评估组织软件过程能力的方法的一个例子是 SEI 软件过程评估方法。

行动计划确定；

- 一 将处理哪一项评估发现，
- 一 实施更改的指南，这些更改处理评估所发现的问题，和
- 一 负责实施更改的组或个人。

活动 2

组织制订和维护用于其软件过程制定和改进活动的计划。

该计划：

- 1、采用根据软件过程评估和其它组织改进倡议所制定的行动计划作为主要输入。
- 2、定义待进行的活动和这些活动的时间表。
- 3、规定负责这些活动的组及个人。
- 4、确定所要求的资源，包括职员和工具。
- 5、当首次发行时及每当作出主要修改时，经受同行评审。

参考同行评审关键过程区域。

- 6、受到组织的软件经理和高级经理的评审和认同。

活动 3

在组织层上协调组织和项目制定及改进其软件过程的活动。

这项协调包括对以下各项的制定和改进：

- 1、组织的标准软件过程。

参考组织过程定义关键过程区域的活动 1 和 2 以便找到包括组织标准软件过程的实践。

- 2、项目定义软件过程。

参考集成软件管理关键过程区域的活动 1 和 2 以便找到包括项目定义软件过程的实践。

活动 4

在组织层上协调组织的软件过程数据库的使用。

组织的软件过程数据库用来收集有关组织和项目的软件过程及由此产生的软件产品的

信息。

参考组织过程定义关键过程区域的活动 5 以便找到包括组织的软件过程数据库的实践。

活动 5

对组织中限制使用的新的过程、方法和工具进行监控和评价，并当合适时，将其推广到组织的其它部分。

活动 6

在整个组织内协调有关组织和项目的软件过程的培训。

1. 制定与组织和项目的软件过程有关的专题培训计划。

2. 当合适时，可以由负责组织的软件过程活动的组（例如软件工程过程组）或培训组织准备和进行培训。

参考培训大纲关键过程区域。

活动 7

将组织和项目的有关软件过程制定和改进的活动通知与实施软件过程有关的组。

通知这些人和使这些人介入的方法的例子有：

—关于过程的电子布告栏，

—过程咨询委员会，

—工作组，

—信息交流会，

—述评，

—过程改进群组（team），和

—非正式的讨论。

测量和分析

测量 1

进行测量并将测量结果用于确定组织的过程制定和改进活动的状态。

测量的例子有：

—在组织的有关过程的评估、制定、改进的活动中所完成的工作、化费的工作量和消耗的资金与有关这些活动的计划相比较；和

—每次软件过程评估的结果，和以前评估的结果和建议相比较。

验证实施

验证 1

高级管理者定期参与评审有关软件过程制定和改进的活动。

高级管理者定期评审的主要目的是在合适的抽象层次上并以及时的方式了解和洞察软件过程活动。评审间隔应该满足组织的需要，只要已存在报告例外情况的合适机制，间隔可以长。

1. 对照计划评审有关制定和改进软件过程的活动进展和状态。

2. 分析在较低层次上无法解决的矛盾和问题。

3. 对行动条款进行安排和评审，并跟踪至结束。

4. 准备每次评审的摘要报告，并将其散发给受影响的组和个人。

组织过程定义

等级 3（已定义的）的一个关键过程区域

组织过程定义的目的是开发和维护一组便于使用的软件过程财富，这些财富改进横跨各项目的过程性能，并为组织积累性的长期得益打下基础。

组织过程定义包括制定和维护组织的标准软件过程，以及相关的过程财富，例如：软件生存周期的描述，过程剪裁指南和准则，组织软件过程数据库和软件过程一有关文档库。

这些财富能以多种方式汇集，取决于对组织过程定义的组织实施。例如软件生存周期的描述可以是组织标准软件过程的一个必备部分，或者软件过程一有关文档库的某些部分可以存储在组织软件过程数据库中。

在开发、实施和维护项目定义软件过程中，可以使用组织软件过程财富。（与项目定义软件过程的制定和维护有关的实践在集成软件管理关键过程区域中加以描述。）

目标

目标 1

制定和维护组织的标准软件过程。

目标 2

收集和评审有关软件项目使用组织标准软件过程的信息，并使其合用。

执行约定

约定 1

组织遵循书面的制定和维护标准软件过程及相关过程财富的方针。

组织的软件过程财富包括：

- 一组织的标准软件过程，
- 一关于项目剪裁组织标准软件过程的指南和准则，
- 一对批准使用的软件生存周期的描述，
- 一组织的软件过程数据库，和
- 一以前开发的、适合重用的软件过程一有关文档的库。

这个方针一般规定：

- 1．标准软件过程是对组织定义的。

标准软件过程的主要目的是在项目间能最大限度地共享过程财富和经验，并使得能在组织层上定义和聚集各项目共用的一组标准的过程测量。

组织标准软件过程可以包含多种软件过程。因为软件项目可以以多种方式组织应用、生存周期、方法论和工具，所以为了阐述对不同应用领域、生存周期、方法论和工具的需要，就可能需要多种软件过程。

- 2．项目定义软件过程是组织标准软件过程经剪裁的版本。

参考集成软件管理关键过程区域的活动 1 以便找到剪裁组织标准软件过程的实践。

3．组织的软件过程财富得到维护。

4．对从各个项目收集来的信息加以组织，并用于改进组织的标准软件过程。

所收集的信息的例子有：

—过程和产品测量，

—经验教训，和

—其它过程—有关的文档。

执行能力

能力 1

为制定和维护组织的标准软件过程和有关过程财富提供足够的资源和投资。

1．负责组织软件过程活动的组（例如软件工程过程组）完成或协调组织标准软件过程及有关过程财富的开发和维护。

参考组织过程焦点关键过程区域以便找到包括负责组织软件过程活动的组的实践。

2．使得支持过程的开发和维护的工具合用。

支持工具的例子有：

—桌面出版工具，

—数据库管理系统，和

—过程建模工具。

能力 2

制定和维护组织的标准软件过程和有关过程财富的个人受到为完成这些活动所要求的培训。

培训的例子有：

—软件工程实践和方法，

—过程分析和建立文档的方法，和

—过程建模。

参考培训大纲关键过程区域。

执行的活动

活动 1

按照已文档化的规程制定和维护组织的标准软件过程。

该规程一般规定：

1．组织的标准软件过程符合（当合适时）强加给组织的软件方针、过程标准和产品标准。

2．组织的标准软件过程满足（当合适时）通常由其顾客要求项目遵守的软件过程标准和产品标准。

3．当合适时，将最先进的软件工程工具和方法纳入组织的标准软件过程。

4．描述软件科目间的内部过程界面。

软件工程科目的例子包括：

—软件需求分析，

—软件设计，

- 编码，
- 软件测试，
- 软件配置管理，和
- 软件质量保证。

5．描述在软件过程和其它受影响的组的过程之间的外部过程界面。

其它受影响的组的例子有：

- 系统工程组，
- 系统测试组，
- 合同管理组，和
- 文档支持组。

6．对负责组织软件过程活动的组（例如软件工程过程组）关于组织标准软件过程的更改建议在更改被纳入标准过程之前建立文档、进行评审和批准。

造成更改的根源的例子有：

- 软件过程评估的发现和和建议，
- 项目对组织标准软件过程的剪裁结果，
- 从监控组织的和项目的软件过程活动中所得到的经验教训，
- 组织的职员和经理所提议的更改，和
- 被分析和解释的过程和产品的测量数据。

7．当合适时，制定将更改引入到现正进行的项目的软件过程中的计划。

8．在最初制定组织的标准软件过程时和每当对其作出重大更改或补充时，该标准软件过程的描述经过同行评审。

参考同行评审关键过程区域。

9．将组织的标准软件过程的描述置于配置管理之下。

参考软件配置管理关键过程区域。

活动 2

按照已制定的组织标准对组织的标准软件过程建立文档。

这些标准一般规定：

1．将过程分解为过程构成元素，这些元素具有为了了解和描述过程所需要的粒度。

每个过程元素包括一妥善定义的、有界的、紧密相关的活动集合。

过程元素的例子包括：

- 软件估计元素，
- 软件设计元素，
- 编码元素，和
- 同行评审元素。

过程元素的描述可以是待填充的样板，待完成的片段，待精炼的抽象，待修改或已使用的未修改的完整描述。

2．描述每个过程元素并阐述：

- ▮ 所要求的规程、条例（practice）方法和技术；
- ▮ 适用的过程和产品标准；
- ▮ 实施过程的职责；
- ▮ 所要求的工具和资源；
- ▮ 输入；
- ▮ 所生产的软件工作产品；
- ▮ 应经受同行评审的软件工作产品；

准备就绪和完成准则；和

待来集的产品和过程数据。

3．描述过程元素间的关系，并阐述其：

排序，

界面，和

内部相关性。

过程元素间的这种关系有时称作软件过程体系结构。

活动 3

对经批准供项目使用的软件生存周期的描述建立文档并进行维护。

软件生存周期的例子有：

一瀑布型，

一部分重叠的瀑布型，

一螺旋型，

一串型构造，和

一单个原型 / 部分重叠瀑布型。

1．软件生存周期与组织标准软件过程相容。

2．对于所提议的对软件生存周期描述的更改，在它们被采纳之前，由负责组织软件过程活动的组（例如，软件工程过程组）建立文档、进行评审和批准。

3．软件生存周期的描述，在最初形成文档时和每当作重大更改和补充时，经受同行评审。

参考同行评审关键过程区域。

4．对软件生存周期的描述进行管理和控制。

“进行管理和控制”意味着在给定时间（过去或现在）使用的工作产品的版本是已知的（即版本控制），而且以受控的方式引进更改（即更改控制）。

如果希望有比“进行管理和控制”所蕴含的更高程度的控制，则工作产品可置于配置管理的完备的纪律之下，正如在软件配置管理关键过程区域中所描述的。

活动 4

制定和维护项目剪裁组织标准软件过程的指南和准则。

1．剪裁指南和准则包括：

选择和剪裁用于项目的软件生存周期，

剪裁组织的标准软件过程以适应软件生存周期和项目的特征，和

剪裁的例子包括：

一为新的生产线或宿主环境修改过程，

一为特定项目或项目类定制过程，和

一详细描述过程和在过程上增加细节使得所产生的项目定义软件过程能够通过。

用于对项目定义软件过程建立文档的标准。

2．对剪裁的指南和准则所提议的更改，在它们被采纳之前，由负责组织软件过程活动的组（例如软件工程过程组）建立文档、进行评审和批准。

3．对剪裁指南和准则进行管理和控制。

活动 5

建立和维护组织的软件过程数据库。

1．建立数据库去收集有关软件过程和产生的软件工作产品的数据，并使该数据适用。

过程和工作产品的数据的例子有：

一软件规模、工作量和成本的估计；

- 有关软件规模、工作量和成本的实际数据；
- 效率数据；
- 质量度量；
- 同行评审的范围（coverage）和效率；
- 测试范围和效率；
- 软件可靠性测量；
- 在软件需求中发现的缺陷的数目和严重性；和
- 在软件代码中发现的缺陷的数目和严重性。

2．评审进入数据库的数据以保证数据库内容的完整性。

此外，数据库也包含或索引那些实际的测量数据以及为了解释测量数据和评估其合理性和适用性所需要的信息和数据。

3．管理和控制数据库。

4．控制用户存取数据库的内容以保证数据的完备性、完整性和精度。

存取仅限于需要进入、更改、评审、分析或抽取数据的人。

保护敏感数据并适当控制对这些数据的存取。

活动 6

建立和维护软件过程—有关文档库。

软件过程—有关文档的例子包括：

- 项目定义软件过程的描述，
- 项目的标准，
- 项目的规程，
- 项目软件开发计划，
- 项目测量计划，及
- 项目的过程培训材料。

1．评审候选的文档产品，并将未来可能有用的适当产品放入库中。

2．对文档产品进行分类编目以便容易存取。

3．评审对当前库内的文档产品所作的修正，在合适时，更新库内容。

4．使得库内容可供软件项目和其它软件—有关组使用。

软件—有关组的例子有；

- 软件质量保证组，
- 软件配置管理组，
- 软件测试组，和
- 文档支持组。

5．定期评审每个文档产品的使用，将其结果用于维护库内容。

6．对库内容进行管理和控制。

测量和分析

测量 1

进行测量并将测量结果用于确定组织的过程定义活动的状态。

测量的例子有：

- 过程开发和维护的进度里程碑的状态，和
- 过程定义活动的成本。

验证实施

验证 1

软件质量保证组评审和（或）审计组织的开发和维护组织标准软件过程和有关过程财富的活动和工作产品，并报告结果。

参考软件质运保证关键过程区域。

至少，这些评审和（或）审计要验证：

- 1．在开发、建立文档和维护组织标准软件过程和有关过程财富时遵循适当的标准。
- 2．控制和恰当地使用组织的标准软件过程和有关的过程财富。

培训大纲

等级 3（已定义的）的一个关键过程区域

培训大纲关键过程区域的目的是培育个人的技能和知识 ,使他们能有效地和效率高地履行其职责。

培训大纲首先分别识别出组织、项目和个人所需要的培训 ,然后展开或设法获得培训以满足所识别的需求。

每个软件项目评价它当前的和未来的技能需求 ,并决定如何才能获得这些技能。某些技能可有效地和效率高地通过非正式的载体传递 (例如在职培训和非正式指导) ,而其它的技能则需要较正式的培训载体 (例如课堂培训和受指导的自学) 才能有效地和效率高地传递。必须选择和使用恰当的载体。

这个关键过程区域仅包括培训组完成培训功能的实践 ,而识别特定培训专题所需要的知识或技能) 的实践则包含在各个有关的关键过程区域的执行能力共同特点中。

目标

目标 1

培训活动是有计划的。

目标 2

提供用以培育为履行软件管理和技术角色所需要的技能和知识的培训。

目标 3

软件工程组和软件一有关组中的个人受到履行其角色所必须的培训。

执行约定

约定 1

组织遵循书面的满足其培训需要的方针。

该方针一般规定 :

- 1 . 识别每个软件管理和技术角色所需的技能和知识。
- 2 . 识别和批准传递技能和知识的培训载体。

经批准的培训载体的例子有 ;

- 课堂培训 ,
- 计算机辅助教学 ,
- 受指导的自学 ,
- 正式的学徒和教练计划 , 和
- 使用便利的录像。

人提供培训以建立组织的技能基础 , 满足项目的特定需求和培育个人的技能。

- 4 . 可在组织内展开培训或当合适时从组织外获得培训。

外部培训源的例子有：

- 一顾客提供的培训，
- 一商业上现有的培训课程，
- 一科学计划，
- 一专业性会议，和
- 一讨论班。

执行能力

能力 1

存在一个组负责实现组织的培训需求。

培训组的成员可能包括全日制或非全日制的来自本组织的指导者，也可能来自外部。一个组是负责一组作业或活动的部门、经理和个人的集合。组的规模可以变化：从一个受指派的非全日制的单个人，到几个从不同部门指派来的非全日制的个人，到几个全日制的个人。建立一个组时应考虑的方面包括指派的作业或活动、项目的规模、组织机构和组织文化。某些组，例如软件质量保证组，集中注意力于项目的活动，而其它组，例如软件工程过程组，集中关注全组织的活动。

能力 2

为实施培训大纲提供足够的资源和投资。

培训大纲的要素的例子包括：

- 一组织的培训计划，
- 一培训材料，
- 一培训的展开或获得方法，
- 一培训的进行，
- 一培训设施，
- 一培训的评价，和
- 一培训记录的维护。

1．指派一个经理负责实施组织的培训大纲。

2．使得支持培训大纲活动的工具合用。

支持工具的例子有：

- 一工作站，
- 一教学的设计工具，
- 一数据库程序，和
- 一编写介绍材料的程序包。

3．使得适当的设施可用于进行培训。

课堂培训设施应和学生的工作环境分开以消除打扰。

在合适处，培训应在非常类似于实际性能条件的设置中进行，培训包括仿真实际工作情况的的活动。

能力 3

培训组的成员具有完成其培训活动所需要的技能和知识。

提供这些技能和知识的方式的例子有。

- 一在教学技巧方面的培训，和
- 一在题材方面的知识更新培训。

能力 4 软件经理接受有关培训大纲的定向培训。

执行的活动

活动 1

每个软件项目制定和维护一个规定其培训需要的培训计划。

该计划包括：

1. 所需要的一组技能及何时需要那些技能。
2. 要求培训的技能和将经由其它载体获得的技能。

某些技能能有效地和效率高地通过非正式载体（例如非正式的培训 and 报告，阅读书和杂志，“信手画谈”，午餐讨论会，在职培训和非正式的指导）加以传递；而另一些技能，要有效地和效率高地传递就需要基于较正式的培训载体（例如，课堂培训，计算机辅助教学，有指导的自学，使用便利的录像和正式的学生和教师计划）。

3. 所要求的培训，被要求接受培训的人，及何时进行培训。

参考全部其它的关键过程区域中的执行能力共同特点以便找到特定培训需求的例子。

在合适处，对个人的培训与其工作职责紧密结合，使培训后的一段合理的时间内，在职活动或其它外部的经历将增强培训。

4. 将如何提供培训。

培训可以由软件项目、组织的培训组或一个外部组织提供。

适合软件项目完成的培训例子有：

- 在项目特定的应用领域和要求方面的培训，
- 在项目的软件体系结构方面的培训，和
- 在项目层上能更有效地和更效率高地进行的其它培训。

活动 2

按照已文档化的规程制定和修正组织的培训计划。

该规程一般规定：

1. 计划利用在软件项目培训计划中所指出的培训需求。
2. 以组织所需要的技能及何时需要这些技能为根据，确定将提供的特定培训。
3. 当合适时对组织的培训计划进行修订以纳入更改。
4. 发表组织的培训阶段时和每当作出主要的修改时，受影响的个人评审该计划。

受影响的个人包括：

- 高级管理者，
- 软件经理，和
- 软件—有关组的经理。

5. 对组织的培训计划进行管理和控制。

“进行管理和控制”意味着在给定时间（过去或现在）使用的工作产品的版本是已知的（即版本控制），而且以受控的方式引进更改（更改控制）。

如果希望有比“进行管理和控制”所蕴含的更高程度的控制，则工作产品可置于配置管理的完备的纪律之下，正如在软件配置管理关键过程区域中所描述的。

6. 对受影响的组和个人，组织的培训计划是易于利用的。

受影响的组 and 个人的例子有：

- 高级管理者，
- 培训组，
- 软件—有关组的经理，
- 软件工程组（包括全部小组，例如软件设计小组），

- 软件估计组，
- 系统工程组，
- 系统测试组，
- 软件质量保证组，
- 软件配置管理组，
- 合同管理组，和
- 文档支持组。

活动 3

按照组织的培训计划进行组织的培训。

该计划包括；

- 1．组织内所需要的特定培训及何时需要。
- 2．将从外部源获得的培训和将由培训组提供的培训。
- 3．准备和进行培训或从外部获得培训所需要的资金和资源（包括职员、工具和设施）。
- 4．培训组将开发的培训课程中所使用的教材标准。
- 5．对于将由培训组开发的培训课程，其开发和修订的时间表。
- 6．进行培训的日程表，它以所预测的需要日期和预计的学生数为根据。
- 7．用于以下各项的规程：

- 选择将接受培训的个人，
- 注册和参加培训，
- 保持所提供培训的记录，和
- 收集、评审和使用对培训的评价及其它的培训反馈信息。

活动 4

按照组织标准开发和维护在组织层上准备的培训课程。

这些标准要求：

- 1．编写对每个培训课程的描述。
该描述所阐述的专题的例子有：
 - 须计的听众，
 - 为参加培训所作的准备工作，
 - 培训目的，
 - 培训期的长度，
 - 课程计划，
 - 确定学生满意地完成学业的准则，
 - 定期评价培训有效性的规程，和
 - 特殊的考虑，例如：先导性试验或现场试验培训课程，更新培训的需要和后续培训的机舍。
- 2．评审用于培训课程的材料。
评审培训材料的个人的例子有；
 - 训练专家，
 - 主题方面的专家，和
 - 待评审的培训课程先导试验期学员代表。
- 3．对培训课程的材料进行管理和控制。

活动 5

制定免修所要求的培训应遵循的规程，并将其用于确定是否那些个人已具有履行其岗位职责所要求的知识和技能。

活动 6

保持培训记录。

1. 保留所有成功地完成每个培训课程或其它经批准的培训活动的学生的记录。
2. 保留所有成功地完成指定要求培训的学生的记录。
3. 使得在考虑任命职员和经理时可以利用成功完成培训的记录。

测量和分析

测量 1

进行测量并将测量结果用于确定培训大纲活动的状态。

测量的例子有：

- 每次培训课程的实际参加情况与预计出席情况相比较，
- 在提供培训课程方面的进展与组织的和项目的培训计划相比较，和
- 某时间段内批准的培训免修者的数目。

测量 2

进行测量并将测量结果用于确定培训大纲的质量。

测量的例子有：

- 培训后测试的结果，
- 来自学生的对课程的评审，和
- 来自软件经理的反馈。

验证实施

验证 1

高级管理者定期参与评审培训大纲活动。

高级管理者定期评审的主要目的是在合适的抽象层次上以及时的方式了解和洞察软件过程活动。评审间隔应该满足组织的需要，只要存在报告例外情况的合适机制，间隔可以长。

参考软件项目跟踪和监督关键过程区域的验证 1 以便找到包括高级管理者监督评审的典型内容的实践。

验证 2

定期独立地评价培训大纲；看其与组织需要的一致性和贴切性。

验证 3

评审和（或）审计培训大纲活动和工作产品，并报告其结果。

至少，评审和（或）审计要验证：

1. 制定和修订组织培训计划的过程得到遵循。
2. 制定和修订培训课程的过程得到贯彻。
3. 培训记录得到适当地维护。
4. 被指定为要求受某种特定培训的个人完成了那项培训。
5. 组织的培训计划得到遵循。

集成软件管理

等级 3（已定义的）的一个关键过程区域

集成软件管理的目的是将软件工程活动和管理活动集成为一个协调的、已定义的软件过程，该软件过程是从组织的标准软件过程和有关的过程财富剪裁而得到的，这在组织过程定义中加以描述。

集成软件管理包括制订项目定义软件过程并采用此已定义的软件过程去管理软件项目。项目定义软件过程是组织标准软件过程经剪裁的版本，剪裁是为了针对项目的具体特征。

软件开发计划是基于项目定义软件过程并描述将如何实施和管理项目定义软件过程的活动。软件项目的规模、工作量、成本、进度、人员配置和其它资源等与项目定义软件过程的作业紧密相连。

由于项目定义软件过程全部是由组织标准软件过程剪裁而得到的，因此软件项目能共享过程数据和所吸取的经验教训。

关于估计、策划和跟踪软件项目的基本实践均在软件项目策划和软件项目跟踪和监督等关键过程区域中描述。当问题出现时它们集中注意力于识别问题及调整计划和（或）性能以便解决这些问题。本关键过程区域的实践建立在那两个关键过程区域的实践之上，是对它们的补充。集成软件管理的重点转移到采取预防措施防止问题发生，避免这些问题的后果或使之影响最小。

目标

目标 1

项目定义软件过程是组织标准软件过程经剪裁的版本。

目标 2

按照项目定义软件过程对项目进行计划和管理。

执行约定

约定 1

项目遵循书面的组织方针，该方针要求采用组织的标准软件过程和有关的过程财富去计划和管理软件项目。

参考组织过程定义关键过程区域以便找到包括组织标准软件过程和有关过程财富的实践。

该方针一般规定：

- 1．每个项目通过剪裁组织的标准软件过程形成项目定义软件过程的文档。
- 2．对项目偏离组织标准软件过程的情况建立文档并且它要经过批准。
- 3．每个项目按照项目定义软件过程完成其软件活动。
- 4．项目收集合适的项目测量数据并将其存储在组织的软件过程数据库中。

参考组织过程定义关键过程区域的活动 5 以便找到包括组织软件过程数据库的实践。

执行能力

能力 1

为使用项目定义软件过程去管理软件项目提供足够的资源和投资。

参考软件项目策划关键过程区域的能力 3 和软件项目跟踪和监督关键过程区域的能力 3 以便找到包括用于软件项目策划、跟踪和监督的资源和投资的实践。

能力 2

负责制定项目定义软件过程的个人在如何裁剪组织标准软件过程和使用有关的过程财富方面接受所要求的培训。

培训的例子包括：

- 使用软件过程数据库，
- 使用组织的标准软件过程，和。
- 使用有关裁剪组织标准软件过程以满足软件项目需要的指南和准则。

参考培训大纲关键过程区域。

能力 3

软件经理在基于项目定义软件过程管理软件项目的技术、行政和人员等方面接受所要求的培训。

参考软件项目策划关键过程区域的能力 4 和软件项目跟踪和监督关键过程区域的能力 4 以便找到包括关于软件项目策划、跟踪和监督等方面的培训的实践。

培训的例子包括；

- 在项目定义软件过程的基础上用于软件估计、策划和跟踪的方法和规程，和
- 用以识别、管理和交流软件风险的方法和规模。

参考培训大纲关键过程区域。

执行的活动

活动 1

按照已文档化的规程剪裁组织标准软件过程来制定项目定义软件过程。

参看组织过程定义关键过程区域的活动 2 以便找到包括组织标准软件过程内容的实践。

该规程一般规定：

1．软件生存周期：

从组织已批准的那些生存周期中选择，以满足项目的合同限制和运行限制；

参考组织过程定义关键过程区域的活动 3 以便找到包括经批准的软件生存周期的实践。

如需要的话以组织剪裁指南和准则允许的方式加以修改；和

按照组织的标准建立文档。

参考组织过程定义关键过程区域的活动 4 以便找到包括组织的剪裁指南和准则的实践。

2．对项目定义软件过程的描述建立文档。

参考组织过程定义关键过程区域的活动 2 以便找到包括过程定义的期望内容的实践。

当合适时，利用组织的过程财富进行剪裁。

3．负责协调组织软件过程活动的组（例如软件工程过程组）评审为项目而进行的对组织标准软件过程的剪裁工作，该工作由高级管理者批准。

参考组织过程定义关键过程区域的活动 6 以便找到包括软件过程一有关文档库的实践

高级管理者对允许偏离组织标准软件过程的情况建立文档、进行评审和批准。

4. 当合适时, 高级管理者和软件项目的顾客对允许偏离合同的软件过程需求建立文档, 进行评审和批准。

5. 对项目定义软件过程的描述进行管理和控制。

“进行管理和控制”意味着在给定时间(过去或现在)使用的工作产品的版本是已知的(即版本控制), 而且以受控的方式引进更改(即更改控制)。

如果希望有比“进行管理和控制”所蕴含的更大程度的控制, 则工作产品可置于配置管理的完备的纪律之下, 正如在软件配置管理关键过程区域中所描述的。

活动 2

按照已文档化的规程修订每个项目定义软件过程。

该规程一般规定:

1、对下列各项导致的更改建立文档和进行系统地评审;

从监控组织的项目软件活动中得到的经验教训,

软件项目所提议的更改, 和

过程和工作产品的测量数据。

2. 对项目定义软件过程的更改在其被纳入前进行评审和批准。

评审更改的个人的例子有;

一负责组织软件过程活动的组响如软件工程过程组)的成员,

一软件经理, 和

一项目软件经理。

批准更改的个人的例子有;

一项目软件经理, 和

一项目经理。

活动 3

按照已文档化的规程制定和修改项目的软件开发计划, 该计划描述项目定义软件过程的使用。

参考软件项目策划关键过程区域的活动 6 和 7 及软件项目跟踪和监督关键过程区域的活动 1 和 2 以便找到包括软件开发计划的实践。

活动 4

按照项目定义软件过程管理软件项目。

参考软件项目策划和软件项目跟踪和监督关键过程区域以便找到包括管理软件项目的基本实践。

项目定义软件过程一般规定:

1. 为采集、分析和报告那些管理软件项目所需要的测量数据作好准备。

2. 使软件估计、策划和跟踪的活动与项目定义软件过程的关键作业及工作产品相关。

3. 制定准备就绪准则和完成准则、加以文档化、并用于批准关键作业的起动和确定关键作业的完成。

4. 定义一个已文档化的准则用以指示什么时候重新计划软件项目。

5. 对所得到的技术方面和管理方面的经验教训建立文档并将它存储在组织的软件过程一有关的文档库中。

参考组织过程定义关键过程区域的活动 6 以便找到包括组织的软件过程一有关文档库的实践。

6. 对组织中其它项目的监控活动所取得的技术上和管理上的经验教训 1 系统地进行评

审，并用来估计、计划、跟踪和重新计划本软件项目。

7．人员配置计划阐述软件项目对具有特殊技能和应用领域知识的个人的需要。

8．别适合软件项目的特殊需要的培训需求并写成文档。

参考培训大纲关键过程区域的活动 1 以便找到包括识别项目培训需要的实践。

9．调整那些在与其它组打交道时所遵循的软件计划和过程，以便说明与这些组的差异和它的潜在问题。

差异及问题的例子有：

- 过程成熟度上的差别，
- 过程不相容性，和
- 不同的业务因素。

活动 5

将组织的软件过程数据库用于软件策划和估计。

参考组织过程定义关键过程区域的活动 5 以便找到包括组织的软件过程数据库的实践。

1．将数据库用作估计、计划、跟踪和重新计划软件项目的源；当可能时使用类似软件项目的数据库。

包含在组织的软件过程数据库中的数据例子有：

- 软件工作产品的规模，
- 软件工作量，
- 软件成本，
- 进度，
- 人员配置，和
- 技术活动。

2．将用于推导有关软件规模、工作量、成本、进度及关键计算机资源的使用等估计量的参数值与其它软件项目的相应的值相比较，以便评估它们的正确性。

评估和记录在应用领域和设计方法方面的类似性和差异。

记录对参数值之间出现类似性和差异的合理解释。

记录那些用来判断项目估计值可靠性的推理过程。

3．软件项目提供合适的软件策划数据、重新策划数据和实测数据，以便存储在组织的软件过程数据库中。

软件项目所记录的数据的例子有：

- 作业描述，
- 假定，
- 估计值，
- 经修改的估计值，
- 实测数据，和
- 为重构估计、评估其合理性和推导新工作的估计值所需要的相关信息。

活动 6

按照已文档化的规程管理软件工作产品规模（或软件工作产品的更改的规模）。

参考软件项目策划关键过程区域的活动 9 和软件项目跟踪和监督关键过程区域的活动 5 以便找到包括策划和跟踪软件工作产品规模的基本实践。

该规程一般规定：

1．一个独立于软件工程组的组评审那些估计软件工作产品规模的规程，并在如何使用组织的软件过程数据库中的历史数据去确定可靠的估计值方面提供指导。

独立组的一个例子是软件估计组。

评价软件规模估计值可靠性的方法的 ~ 个例子是 ,和 ~ 个已完成的系统作功能对功能的比较。

- ✍✍ 为规模估计作准备的个人保证用于估计的规程和数据是合适的。
- ✍✍ 当对某个规模估计的正确性有疑问时,一组同行和专家评审该估计。
- 2. 对每个被鉴别为软件风险的软件元素,其规模估计上附加一个偶然性因子。
- ✍✍ 对偶然性的合理解释建立文档。
- ✍✍ 对与减小或消除偶然性相关联的风险进行评估和建立文档。
- 3. 识别现成的或可重用的软件成分。
- ✍✍ 重用性测量说明需求、设计、代码、测试计划和测试规程等等的重用。
- ✍✍ 将修改和采纳可重用成分的工作量化为因子放入到规模估计中去。
- 4. 识别能显著地影响软件工作产品规模的因素并仔细地监控它们。
- 5. 对每个受管理的软件元素建立一个规模阈值,当预计会超出该阈值时,要求采取措施。

活动 7

按照已文档化的规程管理项目的软件工作量和成本。

参考软件项目策划关键过程区域的活动 10 和软件项目跟踪和监督关键过程区域

活动 6

以便找到包括策划和跟踪软件工作量和成本的基本实践。

这个规程一般规定:

1. 将软件工作量、成本和人员配置等剖面模型(如果采用它的话)修改成适应项目情况,在合适处使用现有的历史数据。
 2. 调整所参考的生产率和成本数据以便将项目变量包括进去。
- 项目变量的例子有;
- 一项目组和组织(例如子承包商)的地理位置,
 - 一系统的规模和复杂性,
 - 一需求的稳定性,
 - 一开发用的宿主计算机环境,
 - 一系统的目标环境,
 - 一开发者对应用的熟悉程度和经验,
 - 一资源的适用性,和
 - 一其它特殊限制。
3. 根据需要将整个软件工作量和成本分配给那些独立管理的作业或阶段,以便有效地管理工作量和成本。
 4. 当评审软件工作量和成本状态并修改估计时,将在一段时间内针对已完成工作的实际消耗与软件开发计划相比较,并将其用于精炼对残留工作的工作量和成本的估计。
- ✍✍ 每当对软件需求作重大更改时,更新在估计软件工作量和成本时所用模型的参数值。
 - ✍✍ 在合适处采用有关项目生产率和其它新的软件成本的实际数据。
5. 对每个独立管理的软件作业或阶段建立一个工作量和成本的阈值,当预计它将被超出时,要求采取措施。

活动 8

按照已文档化的规模管理项目的关键计算机资源。

参考软件项目策划关键过程区域的活动 11 和软件项目跟踪和监督关键过程区域的活动 7 以便找到包括策划和跟踪关键计算机资源的基本实践。

这个规程一般规定：

1．当合适时，“基于历史经验、仿真、建立原型分析，去导出对项目的关键计算机资源的估计。

✍✍ 对估计的源和合理解释建立文档。

✍✍ 对项目 and 提供历史数据的源之间在应用领域和设计方法方面的类似和差异进行评估和记录。

✍✍ 记录用于判断估计可靠性的推理过程。

2．调整所计划的计算机资源、分配给软件的系统需求和（或）软件设计，以满足项目的关键计算机资源需要。

3．将现有的计算机资源分配给软件成分。

4．当进行初始估计时，可用的关键计算机资源的能力提供规定的储备能力。

5．对于每个关键计算机资源建立一个阈值，当预计它将被超过时，要求采取措施。

活动 9

按照已文档化的规程管理项目软件进度的关键依赖关系和关键路径。

参考软件项目策划关键过程区域的活动 12、软件项目跟踪和监督关键过程区域的活动 8 和组间协调关键过程区域的活动 4 以便找到包括协商和跟踪关键依赖关系的实践。

该规程一般规定：

1．以和项目定义软件过程相一致的方式在进度表中布署里程碑、作业、约定、关键的依赖关系、人员配置、成本和评审。

✍✍ 软件进度标识那些特定的作业和里程碑，其完成状态能客观地予以确定（即二值或者是 / 否判定）。

制定不同层次的彼此适当相连结的进度细节，以适应不同组和个人的需要。

2．定义和协商关键的依赖关系，并在软件进度表中加以反映。

关键依赖关系既包括软件工程组内部的（即在小组之间的），也包括软件工程组和其它受影响的组之间的关键依赖关系。

3．定义进度表中的关键路径并在软件进度表中加以反映。

4．定期跟踪软件项目的关键依赖关系和进度表的关键路径。

5．对每个关键路径建立其特有的、已建立文档的阈值标准，当预计它将被超过时，要求采取措施。

措施的例子有：

一进行分析和仿真，在功能、质量、成本、进度、人员配置和其它资源间作折衷；

一分配偶然性和进度的备用部分，如果存在的话；

一评价打算采取的措施对所有关键路径的作用；和

一使得决策对受影响的组可视。

活动 10

按照以文档化的规程对项目的软件风险进行识别、评估、建立文档和管理。

参考软件项目策划关键过程区域的活动 13 和软件项目跟踪和监督关键过程区域的活动 10 以便找到包括识别和跟踪风险的基本实践。

要加以管理的软件风险的例子包括那些重要的偶然性，即软件项目在诸如下列领域内不能满足它的目标的可能性；

一进度，

一成本，

一功能性，

一吞吐量或实时性能，

- 可靠性或可用性，和
- 关键计算机资源的使用。

风险管理活动的例子包括：

- 对高风险项目对象的早期识别；
- 对能引进或增加风险的事件的识别；
- 对高风险模块建立原型或早期实现；和
- 对关键项目风险标志的仔细监控。

这个规程一般规定：

- 1．对软件风险管理计划建立文档并用于识别和管理软件风险。

在软件风险管理计划中的条款的例子有：

- 所要求的资源（包括职员和工具）；
- 和风险管理方法（例如；识别、分析、优先级排序、策划、监控、和解决等方法）；
- 已识别风险的清单（包括评估、优先级排序、状态、和计划等清单）；
- 风险管理日程表；
- 职责和权力；
- 交流风险状态和活动等的方法和频率；和
- 测量。

- 2．偶然性策划基于项目定义软件过程，并在项目软件整个生存周期内予以执行。

偶然性策划活动所包括的范围的例子有：

- 备选项的识别，
- 备选项效果的评估，
- 备选项的技术可行性，
- 对管理备用物的安排，和
- 关于何时采取备选项的决策准则。

- 3．对每个软件风险规定其替代物，在可能处，同时规定在替代物中进行选择的准则。

- 4．软件风险管理计划的初次发行和主要修改经受同行评审。

参考同行评审关键过程区域。

- 5．对软件风险管理计划进行管理和控制。

- 6．在所选择的项目里程碑处，在指定的风险检查点和在能影响软件项目的那些重大更改的策划期间，对软件风险进行跟踪、重新评估和重新计划。

在这些重新评估的点上，评审和修订风险的优先级和软件风险管理计划。

将监控风险所获得的信息用于精炼风险评估和软件风险管理计划。

- 7．在有关软件风险、软件风险管理计划和减轻风险工作的结果等的通信中包括软件工程组和其它受影响的组与个人。

受影响的组与个人的例子有：

- 顾客，
- 子承包商，
- 最终用户，
- 软件估计组，
- 系统工程组，
- 系统测试组，
- 软件质量保证组，
- 软件配置管理组，
- 合同管理组，和

一文档支持组。

活动 11

定期执行对软件项目的评审以便确定那些必须的行动,它们使得软件项目的性能和结果与经营的、顾客和最终用户的当前需求和设想需求相一致(当合适时候)。

行动的例子有:

一加快进度,

一更改系统需求以响应市场机会的变化成顾客和最终用户的需求变化,和

一终止项目。

在这些实践中最终用户指的是顾客所指定的最终用户或最终用户的代表。

测量和分析

测量 1

进行测量并将测量结果用于确定集成软件管理活动的有效性。

测量的例子有:

一在一段时间内管理软件项目所消耗的工作量与计划相比较;

一重新计划工作的频率、原因和量值;

一对于每个已识别的软件风险所了解到的有害影响与所估计的损失相比较;和

一在一段时间内跟踪到的非预期的对软件项目有重大不利影响的数目和量值。

验证实施

验证 1

高级管理者定期参与评审管理软件项目的活动。

参考软件项目跟踪和监督关键过程区域的验证 1 以便找到包括高级管理者监督评审典型内容的实践。

验证 2

项目经理既定期地也事件驱动地参与评审管理软件项目的活动。

参考软件项目跟踪和监督关键过程区域的验证 2 以便找到包括项目管理者监督评审的典型内容的实践。

验证 3

软件质量保证组评审和(或)审计管理软件项目的活动和工作产品,并报告其结果。

参考软件质量保证关键过程区域。

至少,评审和(或)审计要查证:

1. 制定和修正项目定义软件过程的过程。
2. 准备项目软件开发计划和软件风险管理计划的过程。
3. 按照项目定义软件过程管理项目的过程。
4. 收集和提供合适数据给组织软件过程数据库的过程。
5. 使用组织软件过程数据库支持软件项目的策划、估计和跟踪活动的过程。

软件产品工程

等级 3（已定义的）的一个关键过程区域

软件产品工程的目的是—致地执行一个妥善定义的工程过程，该过程集成全部软件 Engineering 活动，以便有效地且高效率地生产正确的、—致的软件产品。

软件产品工程包括采用项目定义软件过程（在集成软件管理关键过程区域中加以描述）和适当的方法及工具去进行构造与维护软件的 Engineering 作业。

软件 Engineering 作业包括：分析分配给软件的系统需求（这些系统需求在需求管理关键过程区域中加以描述）、制定软件需求、开发软件的体系结构、设计软件、用代码实现软件、集成软件成分、和测试软件以验证它满足所规定的需求（即分配给软件的系统需求和软件需求）。

制定并评审为进行软件 Engineering 作业所必须的文档（即软件需求文档、软件设计文档、测试计划和测试规程）以保证每个作业都针对其前面作业的结果，而它所产生的结果又适合于后续的作业（包括运行软件和维护软件的作业）。当更改被批准时，受影响的软件工作产品、计划、约定、过程和活动也要修改以反映该被批准的更改。

目标

目标 1

定义和集成软件 Engineering 作业并—致地执行它们以生产软件。

目标 2

软件工作产品彼此间保持一致。

执行约定

约定 1

项目遵循书面的、有关进行软件 Engineering 活动的组织方针。

该方针—般规定：

1．按照项目定义软件过程进行软件 Engineering 作业。

参考集成软件管理关键过程区域的活动 1 和 2 以便找到包括项目定义软件过程的实践。

2．采用合适的方法和工具去构造和维护软件产品。

3．软件计划、作业和产品均可追踪到分配给软件的系统需求。

在这些实践中分配给软件的系统需求称为“分配需求”。

参考需求管理关键过程区域，以便找到包括分配给软件的系统需求的实践。

执行能力

能力 1

为进行软件 Engineering 作业提供足够的资源和投资。

1．能完成以下各种软件工程技术作业的技术熟练的个人是可以得到的：

- ✍✍ 软件需求分析，
- ✍✍ 软件设计，
- ✍✍ 编码，
- ✍✍ 测试，和
- ✍✍ 软件维护。

2．使得支持软件工程技术作业的工具合用。

一般支持工具的例子有：

- 工作站，
- 数据库管理系统，
- 在线帮助工具，
- 作图工具，
- 交互式文档工具，和
- 文字处理系统。

软件需求分析的支持工具的例子有：

- 需求跟踪工具，
- 规格说明工具，
- 建立原型的工具，
- 建模工具，和
- 仿真工具。

软件设计的支持工具的例子有：

- 规格说明工具，
- 建立原型的工具，
- 仿真工具，和
- 程序设计语言。

编码的支持工具的例子有：

- 编辑程序，
- 编译程序，
- 定义索引产生器，和
- 巧妙的打印程序。

软件测试的支持工具的例子有；

- 测试管理工具，
- 测试生成程序，
- 测试驱动程序，
- 测试剖面，
- 符号调试程序，和
- 测试范围分析程序。

能力 2

软件工程技术人员接受为完成他们的工作任务所要求的培训。

软件工程技术人员应接受应用领域方面的培训。

在软件需求分析方面培训的例子有：

- 分析软件需求的原理；
- 对任何持维护的现存软件，其现有的软件需求；
- 为了制定软件需求而访问最终用户和应用领域专家所需的技巧（即需求的导出）；

—项目选择的、用于分析软件需求的工具、方法、惯例和标准的使用。

在软件设计方面培训的例子有：

—设计概念；

—对任何待维护的现存软件的现有设计；和

—项目选择的、用于设计软件的工具、方法、惯例和标准的使用。

在编码方面培训的例子有：

—所选择的编程语言；

—评审待维护的任何现存代码的现有源代码；

—项目选择的、用于编程的工具、方法、惯例和标准的使用；和

—单元测试技术。

在软件测试和其它验证技术方面培训的例子有：

—验证方法（分析、演示和审查，还有测试）；

—测试策划；

—项目选择的、用于测试和验证软件的工具、方法、惯例和标准的使用；

—测试准备就绪和完成的准则；和

—测试范围的测量。

参考培训大纲关键过程区域。

能力 3

软件工程技术人员接受相关软件工程科目方面的定向培训。

相关软件工程科目的例子有：

—软件需求分析，

—软件设计，

—编码，

—测试，

—软件配置管理，和

—软件质量保证。

参考培训大纲关键过程区域。

能力 4

项目经理和全部软件经理接受软件项目技术方面的定向培训。

定向培训的例子包括：

—软件工程的方法和工具，

—应用领域，

—可交付的和不交付的软件及相连的工作产品，和

—关于如何使用所选定的方法和工具管理项目的指南。

参考培训大纲关键过程区域。

执行的活动

活动 1

将合适的软件工程方法和工具集成到项目定义软件过程中去。

参考集成软件管理的活动 1 和 2 以便找到包括项目定义软件过程的实践。

1．按照项目定义软件过程集成软件工程作业。

2，选择适用于软件项目的方法和工具。

选择候选方法和工具是基于它们对以下各方面的适用性；组织的标准、项目定义

软件过程、现有技能基础、可得到的培训、合同需求、能力、易于使用、和支持服务。

✍✍ 将选择一特定工具或方法的理由写入文档。

3. 选择和使用适合于软件项目的配置管理模型。

配置管理模型的例子有：

—退出 / 登入模型，

—合成模型，

—事务处理模型，和

—更改设置（changeset）模型。

4. 将用于开发和维护软件产品的工具置于配置管理之下。

参考软件配置管理关键过程区域。

活动 2

按照项目定义软件过程，通过系统地分析分配需求来制定软件需求并对其加以维护、建立文档和验证。

1. 与制定软件需求有关的个人评审分配需求以保证影响软件需求分析的问题得到识别和解决。

软件需求包括软件功能和性能、对硬件、软件、以及其它系统成分（例如人员）的界面。

2. 采用有效的需求分析方法去识别和推导软件需求。

需求分析方法的例子有：

—功能分解，

—面向对象的分解，

—折衷研究，

—仿真，

—建模，

—建立原型，和

—情景生成。

3. 对需求分析的结果和关于所选择的替代物的合理解释建立文档。

4. 为了保证软件需求是可行的且适宜于用软件实现的、清楚地阐述的、彼此间相互一致的和完全的（当作为一个集合考虑时），分析软件需求。

✍✍ 负责系统需求的组参与鉴别和评审与软件需求有关的问题；对分配需求和软件需求作出合适的更改。

参考需求管理关键过程区域。

5. 将软件需求写成文档。

6. 负责系统测试和验收测试的组分析每一项软件需求，验证它能被测试。

7. 确定验证和确认每项软件需求得到满足的方法并写入文档。

验证和确认的方法的例子有：

—演示，

—系统测试，

—验收测试，

—分析，和

—审查。

8. 对软件需求文档在认为其已完成之前进行同行评审。

参考同行评审关键过程区域。

9. 评审和批准软件需求文档。

评审和批准软件需求文档的个人的例子有：

- 项目经理，
- 系统工程经理，
- 项目软件经理，和
- 软件测试经理。

10．当合适时，顾客和最终用户参与评审软件需求文档。

在这些实践中所指的最终用户是顾客指定的最终用户或者最终用户的代表。

11．将软件需求文档置于配置管理之下。

参考软件配置管理关键过程区域。

12．每当所分配的需求变化时，适当地更改软件需求。

参考需求管理关键过程区域。

活动3

按照项目定义软件过程，开发、维护、用文档记载和验证软件设计，以适应软件需求和形成编码的框架。

软件设计由软件体系结构和软件详细设计组成。

1．制定和评审设计准则。

设计准则的例子有：

- 可验证，
- 遵守设计标准，
- 易于构造，
- 简单，和
- 易于策划。

2．参与软件设计的个人评审软件需求，以保证影响软件设计的问题得到鉴别和解决。

3．在合适处使用应用标准。

应用标准的例子有：

- 关于操作系统界面（operating interfaces）的标准，
- 关于人机界面的标准，和
- 关于网络界面的标准。

4．设计软件时采用有效的方法。

软件设计方法的例子有：

- 建立原型，
- 结构模型，
- 设计重用，
- 面向对象设计，和
- 基本（essential）系统分析。

5．在所采用的软件生存周期和技术的限制内，尽早地开发软件体系结构。

软件体系结构建立软件的顶层框架，它具有妥善定义的内部接口和外部接口。

6．评审软件体系结构以保证那些影响软件详细设计的体系结构问题得到鉴别和解决。

7．基于软件体系结构进行软件的详细设计。

8．将软件设计（即软件的体系结构和详细设计）写成文档。

✍✍ 软件设计文档包括软件成分；软件成分之间的内部接口；软件对其它软件系统、硬件和其它系统成分（例如人）之间的界面。

9．在认为设计已完成之前，对软件设计文档进行同行评审。

参考同行评审关键过程区域。

10．将软件设计文档置于配置管理之下。

参考软件配置管理关键过程区域。

11. 每当软件需求变化时，适当地更改软件设计文档。

活动 4

按照项目定义软件过程，对软件代码进行开发、维护、建立文档和验证，以实现软件需求和软件设计。

1. 参与编码的个人评审软件需求和软件设计，以保证影响编码的问题得到识别和解决。

2. 使用有效的编程方法进行软件编码。

编程方法的例子有：

— 结构化编程，和

— 代码重用。

3. 开发代码单元的时序是基于一个计划，该计划说明诸如关键性、难度、集成和测试问题，在合适时还要说明顾客和最终用户的需要等因素。

4. 在认为一个代码单元已完成之前，对该代码单元进行同行评审和单元测试。

参考同行评审关键过程区域。

5. 将代码置于配置管理之下。

参考软件配置管理关键过程区域。

6. 每当软件需求或软件设计改变时，适当更改代码。

活动 5

按照项目定义软件过程进行软件测试。

1. 当合适时，顾客和最终用户参与制定和评审测试准则。

2. 采用有效的方法进行软件测试。

3. 基于下列因素确定测试的充分性：

—— 所进行测试的层次，

测试层次的例子有：

— 单元测试，

— 集成测试，

— 系统测试，和

— 验收测试。

—— 所选择的测试策略，和

测试策略的例子有；

— 功能测试（黑盒），

— 结构测试（白盒），和

— 统计测试。

—— 将达到的测试范围。

测试范围的方法的例子有：

— 语句范围，

— 路径范围，

— 分支范围，和

— 使用剖面。

4. 对于软件测试的每个层次，建立和使用测试准备就绪判据。

确定测试准备就绪的判据的例子有：

— 在软件单元进入集成测试之前，它们已成功地完成代码同行评审和单元测试；

— 在软件进入系统测试之前，软件已成功地完成集成测试；

— 在软件进入验收测试之前，进行测试准备就绪的评审。

5. 每当（正）被测试的软件或其环境改变时在每个合适的测试层次上进行回归测试。
6. 在认为测试计划、测试规程和测试用例已准备好可供使用之前，对它们进行同行评审。

参考同行评审关键过程区域。

7. 对测试计划、测试规程和测试用例进行管理和控制。

“进行管理和控制”意味着在给定时间（过去或现在）使用的工作产品的版本是已知的（即版本控制），而且以受控的方式引进更改（即更改控制）。

如果希望有比“进行管理和控制”所蕴含的更高程度的正规性，则工作产品可置于配置管理之下，正如在软件配置管理关键过程区域中所描述的。

8. 每当所分配的需求、软件需求、软件设计、或正被测试的代码改变时，适当地更改测试计划，测试规程和测试用例。

活动 6

按照项目定义软件过程，计划和执行软件集成测试。

1. 根据软件开发计划制定集成测试计划并写成文档。
2. 负责软件需求、软件设计、以及系统和验收测试的个人参与评审集成测试用例和测试规程。
3. 对照软件需求文档和软件设计文档的指定版本，进行软件的集成测试。

活动 7

计划和执行软件的系统测试和验收测试以显示软件满足其需求。

进行系统测试以保证软件满足软件需求。

进行验收测试，向顾客和最终用户表明软件满足所分配的需求。

1. 及早地安排用于测试软件的资源，它的量应足以提供充分的测试准备。

准备测试所要求的活动的例子有：

- 一准备测试文档，
- 一调度测试资源，
- 一开发测试驱动程序，和
- 一开发仿真器。

2. 在测试计划中用文字说明系统测试和验收测试，在合适时顾客和最终用户参与评审并批准测试计划。该测试计划包括：

- ▮ 全面的测试和验证方法；
- ▮ 开发组织、子承包商的职责，适当时还有顾客和最终用户的职责；
- ▮ 测试设施，测试仪器和测试保障需求；和
- ▮ 验收准则。

3. 一个独立于软件开发者的测试组制定测试计划并准备测试用例和测试规程。

4. 对测试用例建立文档并在测试开始之前，当合适时，顾客和最终用户参与评审并批准测试用例。

5. 对照基线化软件和基线化分配需求及软件需求的文档，进行软件测试。

6. 用文档记载在测试期间所鉴别出的问题并跟踪直到结束。

参考软件项目跟踪和监督关键过程区域的活动 9 和软件配置管理关键活动区域的活动 5 以便找到包括用文档记载和跟踪问题的实践。

7. 将测试结果写成文档并用作为确定软件是否满足其需求的基础。

8. 对测试结果进行管理和控制。

活动 8

按照项目定义软件过程编写和维护将用于运行和维护软件的文档。

1. 使用合适的方法和工具去编写文档。

方法和工具的例子有；

—文字处理，

—事例研究 (casestudies)，和

—文档重用。

2. 文档专家积极参加策划、编写和维护文档。

3. 在软件生存周期的早期编写文档的初步版本并使顾客、最终用户和软件维护者，在合适时，可用它来进行评审并提供反馈。

文档的例子有：

—培训文档，

—在钱文档，

—用户手册，

—操作手册，和

—维护手册。

4. 对照已基线化的用于软件验收测试的软件，验证文档的最终版本。

5. 对文档进行同行评审。

参考同行评审关键过程区域。

已对文档进行管理和控制。

7. 当合适时，顾客、最终用户和软件维护者评审和批准最终文档。

活动 9

按照项目定义软件过程收集和分析有关同行评审和测试中所鉴别出的缺陷的数据。

待收集和分析的数据种类的例子有：

—缺陷描述，

—缺陷的类别，

—缺陷的严重性，

—包含缺陷的单元，

—受缺陷影响的单元，

—引进缺陷的活动；

—鉴别出缺陷的同行评审或测试用例。

—对鉴别出缺陷的现场情景的描述，和

—鉴别出缺陷的预期结果和实际结果。

活动 10

维护软件工作产品间的一致性，工作产品包括：软件计划、过程描述、分配需求、软件需求、软件设计、代码、测试计划和测试规程。

1. 将软件工作产品写成文档，且文档是容易得到的。

2. 对软件需求、设计、代码和测试用例倒追踪到导出它们的源及后继软件工程活动的产品。

3. 对通过软件需求、设计、代码和测试用例等追踪分配需求的文档进行管理和控制。

4. 随着对软件了解的增加，对软件工作产品、计划、过程描述和活动提出更改建议。分析这些更改建议，适当时予以采纳。

☞ 在作出更改之前，项目确定更改的影响。

☞ 在对分配需求作更改的地方，必须在对任何软件工作产品或活动作更改之前，使分配需求的更改得到批准和采纳。

☞ 对所有软件产品、计划、过程描述和活动的更改进行协调。

✍✍ 与受影响的组协商和交流更改。

受影响的组的例子有；

- 一软件工程组，
- 一软件估计组，
- 一软件测试组，
- 一软件质量保证组，
- 一软件配置管理组，
- 一、合同管理组，和
- 一文档支持组。

✍✍ 更改被跟踪直至完成。

测量和分析

测量 1

进行测量并将测量结果用于确定软件产品的功能性和质量。

测量的例子有：

一在被跟踪的软件产品中积累统计和按阶段统计所识别出的缺陷的数目、类型和严重性；和

一按类别（例如：保密性、系统配置、性能、和可靠性）总计的分配需求，和可追踪到软件需求和系统测试用例的分配需求。

测量 2

进行测量并将测量结果用于确定软件产品工程活动的状态。

测量的例子包括：

一在整个项目的生存期中每个分配需求的状态；

一按问题的严重性和它们存在的时间长度所作的问题报告；

一对分配需求的更改活动；

一对每个提议的更改进行分析的工作量和累计的更改分析总工作量；

一按类别（例如：界面、保密性、系统配置、性能和合用性）计算被纳入软件基线的更改所需的数目；和

一用以实现和测试所采纳的更改所需的规模 and 成本，包括初始估计的和实际的规模 and 成本。

验证实施

验证 1

高级管理者定期参与评审软件产品工程的活动。

参考软件项目跟踪和监督关键过程区域的验证 1 以便找到包括高级管理者监督评审的典型内容的实践。

验证 2

项目经理既定期地也事件驱动地参与评审软件产品工程的活动。

参考软件项目跟踪和监督关键过程区域的验证 2 以便找到包括项目管理者监督评审的典型内容的实践。

验证 3

软件质量保证组评审和（或）审计软件产品工程的活动和工作产品，并报告其结果。

参考软件质量保证关键过程区域。

至少，评审和（或）审计要验证：

1．软件需求已被评审，保证它们是：

✍✍ 完全的，

✍✍ 正确的，

✍✍ 一致的，

✍✍ 可行的，和

✍✍ 可测试的。

2．满足每个软件工程作业的准备就绪准则和完成准则。

3．软件产品符合对它们所规定的标准和要求。

4．完成所要求的测试。

5．按照已文档化的计划和规程，完成系统测试和验收测试。

6．正如在软件测试计划中所记载的那样，测试满足其验收准则。

7．已满意地完成和记录测试。

8．所检测出的问题和缺陷已记入文档，并进行了跟踪和解决。

9．执行了通过软件需求、设计、代码和测试用树对分配需求进行跟踪。

10．在软件产品交付给顾客或最终用户之前，对照软体基线和任何适用的分配需求，验证了用于运行和维护软件的文档。

组间协调

等级 3（已定义的）的一个关键过程区域

组间协调的目的是建立软件工程组与其它工程组一起积极参与的方式以使得项目更能够有效地和效率高地满足顾客的需要。

组间协调包括软件工程组和其它项目工程组一起参与阐述系统层的需求、对象和问题。项目工程组的代表，当合适时与顾客和最终用户一起工作，参与建立系统层的需求、对象和计划，这些需求、对象和计划成为全部工程活动的基础。

对组间的技术工作界面和相互作用加以计划和管理以保证整个系统的质量和统一性。各项目工程组的代表参与定期的技术评审和内部交流，以保证所有工程组都清楚各组的状态和计划，并保证系统和组间的问题受到恰当的关注。

和这些工程作业有关的软件—专门实践在需求管理和软件产品工程关键过程区域中描述。

目标

目标 1

顾客需求得到全部受影响的组的认同。

目标 2

工程组之间的约定得到受影响的组的认同。

目标 3

工程组识别、跟踪和解决组间问题。

执行约定

约定 1

项目遵循书面的组织上关于建立跨学科工程队伍的方针。

该方针一般规定：

1．全部受影响的组定义和评审关于项目的系统需求和项目层的目标。g

受影响的组的例子有：

- 软件工程组，
- 软件估计组，
- 系统测试组，
- 软件质量保证组，
- 软件配置管理组，
- 合同管理组，和
- 文档支持组。

2．工程组协调其计划和活动。

3. 经理负责建立和维护一个环境，它使得在项目工程组之间、当合适时在项目和顾客或最终用户之间、以及贯穿整个组织的相互作用、协调、支持、和群组工作（teamwork）更加方便。

在这些实践中所指的最终用户是顾客一指定的最终用户或最终用户的代表。

执行能力

能力 1

为和其它工程组一起协调软件工程活动提供足够的资源和投资。

能力 2

不同工程组所用的支持工具是相容的，能够进行有效的通信和协调。

应相容的支持工具的例子有；

- 文字处理系统，
- 数据库系统，
- 作图工具，
- 电子表格程序，
- 问题跟踪软件包，和
- 库管理工具。

能力 3

组织中的所有经理接受在以群组形式进行工作方面所要求的培训。

培训的例子有：

- 建立群组；
- 管理群组；
- 建立、促进和便利群组工作；和
- 群组动态特性（groupdynamics）。

参看培训大纲关键过程区域。

能力 4

在每个工程组中的全部作业领导在有关其它工程组所用的过程、方法和标准方面接受定向培训。

参考培训大纲关键过程区域。

能力 5

工程组的成员在作为一个群组进行工作方面接受定向培训。

参考培训大纲关键过程区域。

执行的活动

活动 1

软件工程组和其它工程组，在合适时与顾客和最终用户一起参与建立系统需求。

具体说，这些组：

1. 当合适时定义顾客和最终用户需求的关键特征。
2. 商订关键的依赖关系。
3. 当合适时，对每个将交付给顾客或最终用户的产品验收准则建立文档。

活动 2

项目软件工程组的代表和其它工程组的代表一起工作去监控和协调技术活动,以及解决技术问题。

1. 这些组的代表监控和协调技术活动通过:

- ✍✍ 调整规格说明和提供对系统需求和系统设计的技术评审和批准;

系统需求和系统设计一般是系统工程组的职责,但是希望其它工程组的代表能有意义地参与这些作业。

系统需求和系统设计包括:

- 一全面的系统需求,

- 一系统配置报(即硬件、软件和其它的系统成分),

- 一将需求分配到这些系统成分,并进行跟踪,和

- 一这些系统成分间界面的定义。

- ✍✍ 提供在项目的整个生存周期内,为管理和控制系统需求和项目层目标的更改所必须的项目层上的技术评审和分析;

- ✍✍ 跟踪和评审关于硬件、软件和其它系统成分的设计和开发活动;和

- ✍✍ 评估、制定对与多个工程组有关的技术风险的建议,并跟踪它。

参考集成软件管理关键过程区域的活动 10 以便找到包含风险管理的实践。

2. 组的代表处理技术问题,通过:

- ✍✍ 解决项目层上的矛盾和澄清系统需求和设计问题;

- ✍✍ 提出解决问题的联合建议;和

- ✍✍ 阐述横跨项目多个工程组的过程问题。

活动 3

将已文档化的计划用于交流组间约定,协调和跟踪所进行的工作

该计划:

1. 是下列各项的基线:

- ✍✍ 项目进度,

- ✍✍ 项目的合同和技术方面,和

- ✍✍ 对工程组的职责安排。

2. 用于协调不同工程组之间的活动。

3. 易于被所有工程组的成员使用。

4. 被更新以便包括全部组间的约定和对这些约定的更改。

5. 随着工作进展而更新以反映项目层上的进展和计划更改,特别当完成项目的里程碑时和当计划有重大变化时。

6. 由所有的工程组和项目经理评审和认同。

活动 4

按照已文档化的规程识别、协调和跟踪工程组之间的关键依赖关系。

参考其成软件管理关键过程区域的活动 9 以便找到包括管理关键依赖关系的实践。

这个规程一般规定:

1. 明确地定义每个关键依赖关系,包括:

- ✍✍ 拟提供的产品项,

- ✍✍ 谁将提供它,

- ✍✍ 何时提供它,和

- ✍✍ 验收准则。

2. 在软件工程组和项目及组织中的其它工程组之间协商关键依赖关系。

3. 对关键依赖关系产品项的需要日期和可使用日期与项目进度和软件进度密切相关。

4．有关每个关键依赖关系的协议由接收组和负责提供关键依赖关系产品项的组双方共同建立文档、评审和批准。

5．定期跟踪关键依赖关系，当合适时采取纠正措施。

✍✍ 将状态和实际的或预测的完成情况与用来协调组间约定的计划相比较。

✍✍ 评价迟后完成和过早完成的后果对将来的活动和里程碑的影响。

✍✍ 向合适的经理报告实际的和潜在的问题。

活动 5

所生产的作为其它工程组的输入的工作产品由接收组的代表评审以保证该产品满足他们的需要。

活动 6

对项目工程组的个别代表不能解决的组间问题按已文档化的规程加以处理。

组间问题的例子有：

—不相容的进度，

—不充分的投资，

—技术风险，

—系统层的设计和需求缺陷，和

—系统层问题。

活动 7

项目工程组的代表进行定期的技术评审和交流。

在这些会议上，参加者：

1．当合适时，提供对顾客和最终用户的需求和希望的可视性。

2．监控项目的技术活动。

3．保证各组对技术需求的解释和实现符合系统需求。

4．评审约定以确定它们是否正被满足。

参考软件项目跟踪和监督关键过程区域以便找到包括评审的实践。

5．评审技术风险和其它的技术问题。

参考集成软件管理关键过程区域的活动 10 以便找到风险管理的实践。

测量和分析

测量 1

进行测量并将测量结果用于确定组间协调活动的状态。

测量的例子有：

—软件工程组为支持其它工程组所化资的实际工作量和其它资源；

—其它工程组为支持软件工程组所化费的实际工作量和其它资源；

—软件工程组为支持其它工程组的活动，对特定作业和里程碑的实际完成情况；和

—其它工程组为支持软件工程组的活动对特定作业和里程碑的实际完成情况。

验证实施

验证 1

高级管理者定期参与评审组间协调的活动。

参考软件项目跟踪和监督关键过程区域的验证 1 以便找到包括高级管理者监督评审的

典型内容的实践。

验证 2

项目经理既定期地也事件驱动地参与评审组间协调的活动。

参考软件项目跟踪和监督关键过程区域的验证 2 以便找到包括项目管理者监督评审的实践。

验证 3

软件质量保证组评审和（或）审计组间协调的活动和工作产品，并报告其结果。

参考软件质量保证关键过程区域。

这个关键过程区域的软件质量保证职责可以归入到一项复盖全部项目工程组的质量保证的功能中。

至少，该评审和（或）审计要查证：

- 1．用于识别、协商和跟踪项目工程组之间的关键依赖关系的规程。
- 2．组间问题的处理。

同行评审

等级 3（已定义的）的一个关键过程区域

同行评审的目的是为了及早地和高效率地从软件工作产品中消除缺陷。一个重要的伴随结果是对软件工作产品及可防止的缺陷得到更好的了解。

同行评审包括生产者的同行对软件工作产品进行系统地考察，以便识别缺陷和需作更改的区域。将经受同行评审的具体产品在项目定义软件过程中加以标识，并作为软件项目策划活动的一部分来安排进度，正如在集成软件管理中所描述的。

这个关键过程区域仅包括执行同行评审的实践。而识别需经受同行评审的具体软件工作产品的实践则包含在描述每个软件工作产品的开发和维护的关键过程区域中。

目标

目标 1

同行评审活动是有计划的。

目标 2

识别和消除在软件工作产品中的缺陷。

执行约定

约定 1

项目遵循书面的、组织上关于执行同行评审的方针。

该方针一般规定：

- 1．组织确定将经受同行评审的软件工作产品的一组标准。
- 2．每个项目确定将经受同行评审的软件工作产品。

参考集成软件管理关键过程区域的活动 1 和组织过程定义关键过程区域的活动 2 以便找到包括识别要经受同行评审的软件产品的实践。

软件工作产品的例子包括：

- 一运行软件和支持软件，
 - 一可交付的和不可交付的软件工作产品，
 - 一软件（例如源代码）和非软件工作产品（例如文档），和
 - 一过程描述。
- 3．由经培训的同行评审领导者领导同行评审。
 - 4．同行评审关注正被评审的软件工作产品，而不关注生产者。
 - 5．管理者不使用同行评审的结果去评价个人的行为。

执行能力

能力 1

对每个待评审的软件工作产品，提供足够用以执行同行评审的资源和投资。

提供资源和投资用以：

1. 准备和散发同行评审的材料。
2. 领导同行评审。
3. 评审那些材料。
4. 参加同行评审和根据同行评审识别出的缺陷所要求的任何后续评审。
5. 监控根据同行评审识别出的缺陷对软件工作产品进行的返工。
6. 收集和报告同行评审中所产生的数据。

能力 2

同行评审领导者接受如何领导同行评审的培训。

培训的例子包括：

- 同行评审的对象、原理和方法；
- 策划和组织同行评审；
- 评价同行评审的准备就绪准则和完成准则；
- 执行和促进同行评审；
- 报告同行评审的结果。
- 跟踪和批准针对同行评审所确定的措施的那些返工；和
- 收集和报告同行评审所要求的数据。

参考培训大纲关键过程区域。

能力 3

参加同行评审的评审者接受在同行评审的对象、原理和方法方面所要求的培训。

培训的例子有：

- 同行评审的类型（例如：软件需求评审、软件设计评审、编码评审和软件测试规程评审）；
- 同行评审的对象、原理和方法；
- 评审者的任务；和
- 估计用于准备和参加同行评审的工作量。


参考培训大纲关键过程区域。

执行的活动

活动 1

计划同行评审，并将计划写成文档。

这些计划：

1. 确定将经受同行评审的软件工作产品。
 所选定的软件工作产品包括在组织标准软件过程中所标识出的那组产品。
参考组织过程定义关键过程区域的活动 2 以便找到包括组织的标准软件过程的实践。
2. 规定同行评审的进度。

对于按进度在最近的将来要执行的同行评审，确定参加每一个同行评审的经培训的同行评审领导者和其它的评审者。

活动 2

按照已文档化的规程进行同行评审。

该规程一般规定：

- 1．由经培训的同行评审领导者计划和领导同行评审。
- 2．预先将评审材料散发给评审者以便他们能为同行评审作好充分的准备。

评审材料应包括要经受同行评审的软件工作产品的开发工作所需的有关输入。有关输入的例子有：

- 软件工作产品的对象，
 - 适用的标准，
 - 对设计模块的有关要求，或
 - 对代码模块的有关详细设计。
- 3．已对评审者分派了在同行评审中的任务。
 - 4．详细说明和执行用于同行评审的准备就绪准则和完成准则。

✍✍ 向合适的经理报告在满足这些准则方面的问题。

- 5．使用检查单，以便以一致的方式确定用于评审软件工作产品的准则。

✍✍ 对于特定类型的工作产品和同行评审，需对检查单加以剪裁。

剪裁检查单所要考虑的问题的例子有：

- 对标准和规程的符合性，
- 完备性，
- 正确性，
- 构造的规则，和
- 维护性。

✍✍ 由检查单制定者的同行和潜在的用户评审检查单。

- 6．跟踪同行评审中所确定的措施直至它们得到解决。
- 7．同行评审的成功完成，包括解决同行评审中所识别出的问题的返工，被用作为相关作业的完成准则。

活动 3

记录有关同行评审的执行情况和结果的数据。

数据的例子包括；

- 被评审的软件工作产品的标识，
- 软件工作产品的规模，
- 评审队伍的规模和组成，
- 每个评审者的准备时间，
- 评审会议的长度，
- 发现和改正缺陷的类型和数目，和
- 返工工作量。

测量和分析

测量 1

进行测量并将测量结果用于确定同行评审活动的状态。

测量的例子包括：

- 所完成的同行评审数与计划相比较，
- 同行评审所化费的总的工作量与计划相比较，和

一被评审的工作产品数与计划相比较。

验证实施

验证 1

软件质量保证组评审和（或）审计同行评审的活动和工作产品，并报告其结果。

参考软件质量保证关键过程区域。

至少，评审和（或）审计要验证：

- 1．所计划的同行评审已被执行。
- 2．同行评审的领导者得到适当的有关其角色的培训。
- 3．评审者得到在其角色方面的合适的培训或者对其角色是有经验的。
- 4．准备同行评审、执行同行评审和完成后继措施等的过程得到遵循。
 - 5．同行评审数据的报告是完全的、精确的和及时的。

等级 4 关键实践

定量过程管理

等级 4（已管理的）的一个关键过程区域

定量过程管理的目的是定量地控制软件项目的过程性能。软件过程性能表示遵循一软件过程所得到的实际结果。

定量过程管理包括建立集成软件管理关键过程区域所描述的项目定义软件过程的性能目标、测量过程性能、分析这些测量结果并作出调整以保持过程性能在可接受的范围内。当在可接受范围内过程性能是稳定时，项目定义软件过程、相关连的测量和测量的可接受范围就作为基线建立起来，并被用来定量地控制过程性能。

组织从软件项目采集过程性能数据，并使用这些数据对组织过程定义关键过程区域所描述的组织标准软件过程的过程能力（即，一个新项目能预计获得的过程性能）进行特征描述。

过程能力描述从遵循一个软件过程所得到须期结果的范围（即，预计组织承担下一个软件项目时最可能的输出）。软件项目采用这些过程能力数据来建立和修订其过程性能目标和分析项目定义软件过程的性能。

目标

目标 1

定量过程管理活动是有计划的。

目标 2

项目定义软件过程的过程性能受到定量地控制。

目标 3

组织标准软件过程的过程能力是定量地已知的。

执行约定

约定 1

项目遵循书面的组织上关于测量和定量控制项目定义软件过程性能的方针。

该方针一般规定：

豆．为将项目定义软件过程置于定量控制之下，每个项目实施一个已文档化的计划。

术语“定量控制”的含意是指任何一项适用于分析软件过程、识别软件过程性能变化的特殊原因、和将软件过程性能置于妥善定义的范围内的、定量的或基于统计的技术。

2．有关个人行为的敏感数据受到保护，并适当控制对这些数据的存取。

使用测量数据去评价个人，将对所报告的测量数据的正确性和有用性起不良影响。

约定 2

组织遵循书面的有关分析组织标准软件过程的过程能力的方针。

该方针一般规定；

1．分析项目的过程性能测量结果以建立和维护组织标准软件过程的过程能力基线。

过程能力基线包括：

- 一组织标准软件过程的描述，
- 一测量的标准定义，和
- 一期望的测量值范围。

2．软件项目在建立其过程性能目标上使用组织标准软件过程的过程能力基线。

执行能力

能力 1

存在一个负责协调组织的定量过程管理活动的组。

一个组是负责一组作业或活动的部门、经理、和个人的集会。组的规模可以变化：从一个受指派的非全日制的单个个人，到几个从不同部门指派来的非全日制的个人，到几个全日制的个人。建立一个组时考虑的因素包括指派的作业和活动、项目的规模、组织机构、和组织文化。某些组，例如软件质量保证组，集中注意力于项目的活动，而其它的组，例如软件工程过程组，则集中关注全组织的活动。

1．这个组或者是负责组织软件过程活动的组（例如，软件工程过程组）的一部分，或者它的活动与后者紧密地配合。

能力 2

为定量过程管理活动提供足够的资源和投资。

1．软件工程组和其它软件一有关组的经理和作业领导执行项目定量过程管理活动。

软件一有关组的例子包括：

- 一软件质量保证组；
- 一软件配置管理组，和
- 一文档支持组。

2．存在全组织范围的测量大纲。

组织的测量大纲包括：

- 一定义全组织范围的测量，
- 一收集组织的测量数据，
- 一分析组织的测量数据，和
- 一组织的定量测量目标。

3．使得支持定量过程管理的工具合用。

支持工具的例子包括：

- 一软件源代码分析器；
- 一自动的测试范围分析器，
- 一数据库系统，
- 一定量分析程序包，和
- 一问题跟踪程序包。

能力 3

对于已选择的过程和产品度量，具备对于收集、记录、和分析数据的支持。

在这些实践中所指的产品数据是在分析软件过程时所用的产品度量。

能力 4

实施或支持定量过程管理的个人接受执行这些活动所要求的培训。

培训的例子包括：

一对软件过程建模和分析；
一对过程测量数据进行挑选、收集和确认；和
一使用基本的定量的方法和分析技术（例如，估计模型，Paret。图，和控制图）。
参看培训大纲关键过程区域。

能力 5

软件工程组和其它软件一有关组的成员接受有关定量过程管理的目标和价值的定向培训。

参看培训大纲关键过程区域。

执行的活动

活动 1

按照已文档化的规程制定软件项目的定量过程管理计划。

该规程一般规定；

1．定量过程管理计划基于：

- ✍✍ 组织的有关产品质量、生产率和产品开发周期时间的战略目标；
- ✍✍ 组织的测量大纲；
- ✍✍ 组织的标准软件过程；
- ✍✍ 项目的有关软件产品的质量、生产率和产品开发周期时间的目标；
- ✍✍ 其它的项目定义软件过程的已测量的性能；和
- ✍✍ 项目定义软件过程的描述。

2．该计划经受同行评审。

参看同行评审关键过程区域。

3．该计划由负责组织软件过程活动的组（例如软件工程过程组）评审。

4．对该计划进行管理和控制。

“进行管理和控制”意味着在给定时间（过去或现在）使用的工作产品的版本是已知的（即版本控制），而且以受控的方式引进更改（即更改控制）。

如果希望有比“进行管理和控制”所蕴含的更加正规的手续，则工作产品可置于配置管理的完备的纪律之下，正如在软件配置管理关键过程区域中所描述的。

活动 2

按照项目的定量过程管理计划执行软件项目的定量过程管理活动。

该计划包括：

1．定量过程管理活动的目标（goals）和目的（objectives）。

2．将被测量的软件作业或其它的软件活动。

3．项目定义软件过程的测量装备（instrumentation）。

测量装备基于组织的测量大纲、组织标准软件过程的描述和项目定义软件过程的描述。

4．将执行的定量过程管理活动和关于这些活动的时间表。

除了当前组织和项目的需要之外还应包括对将来的工作可能有用的测量。

5．负责定量过程管理活动的组和个人。

已执行定量过程管理活动所需要的资源，包括人员和工具。

7.在执行定量过程管理活动中应遵循的规程。

活动 3

在项目定义软件过程的基础上，确定数据采集的策略和应作的定量分析。所考虑的项目定义软件过程的属性包括：

1. 作业、活动和它们彼此间的关系。
2. 软件工作产品和它们彼此间的及它们对项目定义软件过程的关系。
3. 过程控制点和数据采集点。

活动 4

按照已文档化的规程，采集用于定量地控制项目定义软件过程的测量数据。

该规格一般规定：

1. 所采集的测量数据支持组织的和软件项目的测量目标和目的。
2. 要确定应采集的具体测量数据、它们的精确定义、对每项测量所预想的使用和分析、以及这些测量数据的过程控制点。

测量数据的例子包括：

- 一在软件规模、成本和进度方面，估计的 / 计划的数据与实际数据相比较；
- 一生产率数据；
- 一正如在软件质量计划中所定义的质量度量；
- 一同行评审的范围和效率；
- 一培训的有效性；
- 一测试的范围和效率；
- 一软件可靠性度量；
- 一在软件需求中所发现的缺陷的数目及严重性；
- 一在软件代码中所发现的缺陷的数目及严重性；
- 一措施条款中已结束条款的数目和比例。

3. 从整个软件生存周期（例如，既有开发阶段又有后开发阶段）选择测量。
4. 测量包括关键软件过程活动和主要软件工作产品的特性。
5. 在遍及所有软件项目上一致地收集有关组织标准软件过程的测量数据。
6. 要控制的度量都是软件活动的自然的结果（在可能处）。
7. 选择支持预先确定的分析活动的测量。

在某些情况，测量可以是面向研究的但应明确指明这点。

8. 独立地评估测量数据的有攻连。
 9. 当合适时，将所采集到的测量数据存储到组织的软件过程数据库中。
- 参阅组织过程定义关键过程区域的活动 5 以便找到包括组织软件过程数据库的实践。

活动 5

按照已文档化的规程分析项目定义软件过程，并将其置于定量控制之下。

该规程一般规定：

1. 预先确定具体的数据分析活动。

数据分析活动的描述包括：

- 一所要求的输入数据，
- 一使用的工具，
- 一所进行的数据操作，
- 一要导出的信息，和
- 一在进行分析时所用的决策判据和作为分析结果所决定的将要采取的措施。

分析技术的例子包括：

- 一 Pareto 图（Paretodiagrams），
- 一控制图（contrdcharts），
- 一趋向图（trenddiagrams）和
- 一散布图（scatterdiasrams）

2. 识别、采集和分析贯穿整个项目定义软件过程的过程活动的测量数据。
3. 所选出的测量恰当地描述它们所表示的过程的特征。
4. 对每个度量确定其预计的均值和方差值。
5. 确定每个度量的可接受范围，并建立项目的过程性能基线。

建立可接受的范围的一个例子是从过程的平均性能计算过去的偏差。

6. 将每个度量的实际值与预计的均值和方差值相比较。

实际过程性能与已确定的可接受范围相比较的例子有：

一将在每千行源代码上所化费的同行评审小时数与通过分析过去数据所确定的上限和下限作比较；和

一将由软件需求（例如“必须”实现的功能数(“shall”数)）换到源代码行数的扩展率与通过分析过去的的数据所确定的上限和下限作比较。

7. 当合适时作出调整以使得实际过程性能与已确定的可接受范围相一致。

8. 当项目定义软件过程受到定量控制时，建立以下各项的基线：

▮ 测量的定义，

▮ 实际测量数据，和

▮ 度量的可接受范围。

9. 管理和控制软件项目的过程性能基线。

活动 6

准备和散发对软件项目的定量过程管理活动结果的书面报告。

1. 在将数据分析结果报告给任何其它人之前，和受该数据影响的那些人一起评审该数据分析结果。

2. 软件经理、软件作业领导和高级管理者收到常规的适合其需要的报告。

3. 软件质量保证组收到常规的适合其需要的报告。

4. 项目经理、高级经理、软件经理和软件作业领导收到应其请求的专门报告。

活动 7

按照已档化的规程建立和维护组织标准软件过程的过程能力基线。

该规程一般规定；

1. 在组织的软件过程数据库中记录项目的软件过程数据，正如在过程性能基线中所概括的那样。

参看组织过程定义关键过程区域的活动 5 以便找到包括组织软件过程数据库的实践。

2. 当合适时，将每个项目定义软件过程的过程性能基线纳入到组织标准软件过程的过程能力基线中。

3. 将组织标准软件过程的过程能力基线写成文档。

4. 考查组织标准软件过程的过程能力趋势以预测最可能的问题或改进的机会。

使用能力趋势的例子有：

一预计软件缺陷的发生并将预计的和实际的相比较，和

一基于来自同行评审和（或）测试的数据预计产品中残留缺陷的分布和特征。

多半是缺陷源的区域的例子有：

一用于估计和策划的条款，

一在软件生存周期的早期（例如需求分析）所进行的活动，

一主要的文档条款，

一在过去一直是易于引入缺陷的条款和活动，

一实施更改和修正缺陷的活动，和

一人工密集的活动。

多半是改进机会的区域的例子有：

- 其它项目和组织已经成功地自动化的活动，
- 不交付的支持性的产品项（item）和活动，例如培训和工具，
- 面向质量的活动，例如同行评审和测试，及
- 人工密集的活动。

5．对组织标准软件过程的过程能力基线进行管理和控制。

6．当承担的软件项目与过去的项目有本质不同时，作为剪裁组织标准软件过程的一部分建立那个项目的新的过程性能基线。

参看集成软件管理关键过程区域的活动 1 以便找到包括项目剪裁组织标准软件过程的实践。

本质不同的例子有：

- 新的应用领域，
- 采用根本不同的技术，和 183
- 在应用规范上的显著变化。

7．跟踪和分析对组织标准软件过程的更改以便评估其对过程能力基线的影响。

测量和分析

测量 1

进行测量并将测量结果用以确定定量过程管理活动的状态。

测量的例子包括：

- 一段时间内定量过程管理的成本与计划作比较的结果；
- 定量过程管理活动的进度里程碑的完成情况，用与已批准的计划（例如确立对有关项目将采用的过程测量、确定将如何收集过程数据，以及采集过程数据）作比较的结果来表示。

验证实施

验证 1

高级管理者定期参加评审定量过程管理活动。

参看组织过程焦点关键过程区域的验证 1 和软件项目跟踪和监督关键过程区域的验证 1 以便找到包括高级管理者监督评审的典型内容的实践。

验证 2 项目经理既定期地也事件驱动地参与评审软件项目的定量过程管理的活动。

参看软件项目跟踪与监督关键过程区域的验证 2 以便找到包括项目管理者监督评审的典型内容的实践。

验证 3

软件质量保证组评审和（或）审计定量过程管理的活动和工作产品，并报告其结果。

参看软件质量保证关键过程区域。

至少，评审和（或）审计要验证：

- 1．定量过程管理活动的计划得到遵循。
- 2．定量过程管理的规程得到遵循。
- 3．正如所要求的那样进行定量过程管理数据的采集和分析工作，包括验证：

存在所需要的数据，

- ✍✍ 已收集所需要的数据，
- ✍✍ 所收集的数据是需要的，
- ✍✍ 所收集的数据支持组织测量大纲的目标和目的，
- ✍✍ 通过数据的有用性对收集数据成本的合理性作出解释，
- ✍✍ 在软件生存周期的恰当点上收集数据，
- ✍✍ 数据是精确的且正确的，
- ✍✍ 数据是及时的，和
- ✍✍ 数据的保密性受到恰当的保护。

软件质量管理

等级 4（已管理的）的一个关键过程区域

软件质量管理的目的是建立对项目的软件产品质量的定量了解和实现特定的质量目标。

软件质量管理包含确定软件产品的质量目标，制定实现这些目标的计划，并监控及调整软件计划。软件工作产品、活动和质量目标，以满足顾客和最终用户对高质量产品的需要及愿望。

软件质量管理的实践基于三个关键过程区域，前二个是建立和实施项目定义软件过程的关键过程区域——集成软件管理和软件产品工程，而后一个是定量过程管理关键过程区域，它建立对项目定义软件过程实现所希望的结果能力的定量了解。

基于组织、顾客和最终用户的需要来建立软件产品的质量目标，这样这些目标才能实现。为实现质量目标，组织制定战略和计划，项目则具体调整其已定义的软件过程。

目标

目标 1

项目的软件质量管理活动是有计划的。

目标 2

软件产品质量的可测目标及其优先级是确定的。

目标 3

实现软件产品的质量目标的实际进程是用数量表示的和受到管理的。

执行约定

约定 1

项目遵循书面的用以管理软件质量的组织方针。

该方针一般规定：

1．项目的软件质量管理活动支持组织对改善软件产品质量的承诺。

那些能提高软件产品质量的过程改进具有组织的最高优先级。

每个新发行的软件产品应该是可测量地好于它的前任产品或主要竞争对手的产品。

2．项目基于项目定义软件过程确定和收集用于软件质量管理的测量数据。

3．项目确定软件产品的质量目标，并监控其实现质量目标的进程。

4．确定软件质量管理的职责并将其分派给软件工程组和其它的软件-有关组。

软件-有关组的例子包括：

—软件质量保证组，

译注：原文打印程式有错：这一段中间的句号应为逗号，定量目标应为质量目标，黑体应为正常体。

—软件配置管理组，和

一文档支持组。

✍✍ 建立使得这些组，能确定他们在满足软件产品质量目标上的成功情况的准则。

执行能力

能力 1

为管理软件产品的质量提供足够的资源和投资。

1．可以得到在诸如安全性和可靠性领域有专长的工程师以便帮助设置软件质量目标和评审朝着此目标前进的进程。

2 使得支持预测、测量、跟踪和分析软件质量的工具合用。

支持工具的例子包括：

一数据采集工具，

一数据库系统，

一电子表格程序，

一软件生存周期仿真器，

一定量分析工具，和

一代码检查工具。

能力 2

实施和支持软件质量管理的个人接受在执行其活动方面所要求的培训。

培训的例子有：

一策划产品的质量约定和目标，

一测量产品和过程质量，和

一用已定义软件过程去控制产品质量。

参看培训大纲关键过程区域。

能力 3

软件工程组和其它软件一有关组的成员接受在软件质量管理方面所要求的培训。

培训的例子包括：

一理解定量管理产品质量的目标和好处。

一收集测量数据，

一理解对于软件过程和产品的质量度量，和

一策划和控制软件产品的质量。

参看培训大纲关键过程区域。

执行的活动

活动 1

按照已文档化的规程制定和维护项目的软件质量计划。

该规程一般规定：

1．当合适时，建立对组织、顾客和最终用户的软件质量需求的理解。

在这些实践中所指的最终用户是顾客指定的最终用户或最终用户的代表。

测量顾客和最终用户的软件质量需求的方式的例子有：

一调查，

一焦点组 (focusgroups), 和

一由用户作的产品评价。

2. 组织、顾客和最终用户的软件质量需求及其优先级可以追踪到分配给软件的系统需求及软件质量目标。

追踪这些需求和优先级的方法的一个例子是质量功能展开 (QualitrFunctionDeployment (QFD))。

追踪需求和优先级到产品的软件质量目标的一个例子是建立交付后允许出现缺陷数目的目标, 随着产品逐渐成熟不断进行预报练习以便评估满足这些目标的可能性。

在这些实践中, 分配给软件的系统需求称为“分配需求”。

参看需求管理关键过程区域以便找到包括分配给软件的系统需求的实践。

3. 对项目定义软件过程满足软件质量目标的能力加以评估并记入文档。

诸如质量功能展开和关于设计的田口 (Taguchi) 方法等技术可用于显示产品质量目标和过程能力的关系。

4. 当合适时, 软件质量计划满足组织的质量计划。

5. 当合适时, 软件质量计划基于组织中以前项目的或当前项目的计划。

6. 在项目的开头, 在主要的项目里程碑处, 以及每当分配需求有重大改变时, 更新软件质量计划。

7. 软件质量计划经受同行评审。

参看同行评审关键过程区域。

8. 受影响的组和个人评审软件质量计划。

受影响的组 and 个人的例子有:

一顾客,

一最终用户,

一软件工程组 (包括所有的小组, 例如软件设计小组),

一软件估计组,

一系统工程组,

一系统测试组,

一软件质量保证组,

一软件配置管理组,

一合同管理组, 和

一文档支持组。

9. 高级管理者评审软件质量计划。

10. 对软件质量计划进行管理和控制。

“进行管理和控制”意味着在给定时间 (过去或现在) 使用的工作产品的版本是已知的 (即版本控制), 而且以受控的方式引进更改 (即更改控制)。

如果希望有比“进行管理和控制”所蕴含的更加正规的手续, 则工作产品可置于配置管理的完备纪律之下, 正如在软件配置管理关键过程区域中所描述的。

11. 对所有受影响的组和个人, 软件质量计划是可以得到的。

活动 2

项目的软件质量计划是项目软件质量管理活动的基础。

该计划包括:

1. 过程中进行软件质量度量的点。

2. 软件产品的有重大影响 (high—leverage) 的质量目标。

软件产品的有重大影响的质量目标是那些能以最少成本提供最大的顾客满意度的质量目标, 或者是顾客或最终用户定的“必须有”的质量目标。

3. 为了在过去质量性能的基础上作出改进，软件项目将实施的行动。

4. 旨在测量软件产品质量的活动。

旨在测量软件产品质量的软件活动的例子有：

- 同行评审，
- 原型开发，
- 产品仿真，和
- 测试。

5. 当合适时，有关软件工作产品的质量目标。

适合记载在项目的软件质量计划中的软件产品质量目标的例子有：

- 计划将实现的特征，和
- 关键特征，如果不实现这些特征，就会产生顾客或最终用户不希望或不需要的产品。

6. 当预计软件产品质量不满足质量目标时将采取的措施。

活动3

在整个软件生存周期，确定、监控和修定项目的软件产品的定量质量目标。

L. 识别哪些描述软件产品将如何好地运行或如何好地被开发和维护的产品质量特征。

软件质量特征的例子包括：

- 功能性，
- 可靠性，
- 维护性，和
- 适用性。

2. 确定用于量化软件产品质量特征的测量。

确定产品质量测量的活动的例子包括：

- 评审以前的性能数据和顾客需求，
- 开发原型，
- 用形式化表示方法描述中间软件产品，
- 采用正式的软件工程方法，和
- 进行测试。

3. 对于每个软件产品质量特征，根据所要求的和所希望的值，选择可测的数字值作为产品的质量目标。

在软件产品可靠性方面可能的质量目标的例子有：

- 如同在需求中所规定的平均故障间隔时间，
- 必须实现的平均故障间隔时间（由分析和实验所确定），和
- 计划达到的平均故障间隔时间。

4. 在项目软件质量计划中用文字说明软件产品质量目标。

适合记载在项目软件质量计划中的软件产品质量目标的例子有：

- 计划要实现的特征；和
- 关键特征，如果不实现这些特征，就会生产顾客或最终用户不希望或不需要的产品。

5. 确定每个软件生存周期阶段的质量目标并写成文档。

软件生存周期阶段的例子包括：

- 软件需求，
- 软件设计，
- 编码，和
- 软件测试。

和软件生存周期阶段有关的质量目标的例子有：

一与每个软件生存周期阶段有关的产品缺陷数将比以前发行的产品的缺陷数降低某个预先确定的百分比。

一预先确定的在测试周期结束时预期将发现缺陷的百分比。

6．随着对产品的了解和对组织、顾客、及最终端用户的需求的了解逐渐增加，修订软件产品和软件生存周期阶段的质量目标。

活动 4

在事件驱动的基础上，对项目软件产品的质量进行测量、分析，并将其与产品的定量质量目标相比较。

参着定量过程管理关键过程区域以便找到包括使用测量数据的实践。

1．计划和执行旨在阐述项目的软件质量目标的软件作业、在软件作业的开头，执行该作业的群组；

✍✍ 评审软件产品的质量目标，

✍✍ 确定适用于软件作业的质量目标，

✍✍ 确定它用于实现软件质量目标的计划，和

✍✍ 评审为实现软件质量目标所作的对过程的更改。

更改的一个例子是修订同行评审的检查单以便阐述已发现的逃过同行评审的缺陷。

2．测量每个软件生存周期阶段的软件工作产品的质量。

测量工作产品质量的方法的例子有：

一同行评审，

一仿真，和

一测试。

3．分析质量测量结果，并将它与软件质量目标作比较以确定是否满足质量目标。

4．采取合适的与软件质量计划相一致的措施，以便使得产品的质量测量结果与软件质量目标相符合。

5．当已确定软件质量目标相互矛盾时（即，不损害另一个目标就不能实现某个目标），采取解决矛盾的措施。

✍✍ 分析实现软件质量目标的成本。

日根据长期经营战略和短期优先级，考虑替代的软件质量目标。

✍✍ 当合适时，顾客和最终用户参与质量权衡决策。

✍✍ 当合适时，修订软件工作产品和计划以反映权衡的结果。

活动 5

将软件项目的产品定量质量目标信当地分配给那些向项目交付软件产品的子承包商。

参看软件子合同管理关键过程区域的活动 1。

测量和分析

测量 1

进行测量并将测量结果用于确定软件质量管理活动的状态。

测量的例子包括；

一质量差的费用（基于已知的质量度量，其精度是能采集到的精度）；和

一实现质量目标的成本。

验证实施

验证 1

高级管理者定期参与评审软件质量管理的活动。

参看软件项目跟踪和监督关键过程区域的验证 1 以便找到包括高级管理者监督评审的典型内容的实践。

验证 2

项目经理既定期地也事件驱动地参与评审软件质量管理活动。

参看软件项目跟踪和监督关键过程区域的验证 2 以便找到包括项目管理者监督评审的典型内容的实践。

验证 3

软件质量保证组评审和（或）审计软件质量管理的活动和工作产品，并报告结果。

参看软件质量保证关键过程区域。

至少，评审和（或）审计要查证：

- 1．项目的软件质量计划的准备工作。
- 2．用于建立和跟踪软件质量目标的过程。

缺陷预防

等级 5（优化的）的一个关键过程区域

缺陷预防的目的是鉴别缺陷的原因并防止它们再出现。

缺陷预防包括分析过去曾遇到的缺陷和采取特别措施以防止将来再出现此类缺陷。缺陷可能曾在其它项目中和在当前项目的早期阶段或作业中已经鉴别出来。缺陷预防活动也是在项目间传播经验教训的一种机制。

为了跟踪已遇到过的缺陷类型和识别出多半会再出现的缺陷，必须对趋势进行分析。基于对项目定义软件过程和其实施方式（正如在集成软件管理和软件产品工程关键过程区域中所描述的）的理解，确定缺陷的根本原因和对于未来活动缺陷的推断。

项目和组织均采取预防缺陷再出现的特别措施。某些组织的措施可以像在过程更改管理关键过程区域中所描述的那样处理。

目标

目标 1

缺陷预防活动是有计划的。

目标 2

缺陷的一般原因已经找出和标识。

目标 3

缺陷的一般原因已按优先级排序和系统地消除。

执行约定

约定 1

组织遵循书面的用于缺陷预防活动的方针。

该方针一般规定：

- 1．制定有关缺陷预防的投资、人员配置和其它资源的长期计划和约定。
- 2．安排缺陷预防活动所需要的资源。
- 3．在全组织范围内实施缺陷预防活动以改进软件过程和产品。
- 4．评审缺陷预防活动的结果以保证这些活动的有效性。
- 5．阐述那些作为缺陷预防活动的结果所鉴别出的管理和技术方面的措施。

约定 2

项目遵循书面的关于缺陷预防活动的组织上的方针。

该方针一般规定：

- 1．在每个项目的软件开发计划中包括缺陷预防活动。
- 2．安排缺陷预防活动所需要的资源。
- 3．阐述那些作为缺陷预防活动的结果所鉴别出的项目管理和技术方面的措施。

执行能力

能力 1

存在组织层上的协调缺陷预防活动的群组。

1. 该群组可以是负责组织软件过程活动的组（例如，软件工程过程组）的一部分，或者其活动与后者紧密地配合。

参看组织过程焦点关键过程区域。

能力 2

存在协调软件项目的缺陷预防活动的群组。

1. 这个群组与负责开发和维护项目定义软件过程的群组紧密结合。

协调缺陷预防活动的群组的成员通常是非全日制的，他们有其它的软件工程活动作为其主要的职责。

参看集成软件管理关键过程区域的活动 1 和 2 以便找到包括开发和维护项目定义软件过程的实践。

能力 3

在项目层和组织层都要为缺陷预防活动提供足够的资源和投资。

1. 当合适时，将缺陷预防活动设计为每个人的职责。

缺陷预防活动的例子包括：

- 作业起动会议，
- 原因分析会议，
- 评审和策划所提议的措施，和
- 实施措施。

2. 制定了管理者参与缺陷预防活动的计划。

3. 当合适时，每个软件项目均派有代表参加协调组织的缺陷预防活动的群组。

4. 使得支持缺陷预防活动的工具适用。

支持工具的例子包括：

- 统计分析工具，和
- 数据库系统。

能力 4

软件工程组和其它软件一有关组的成员接受执行其缺陷预防活动所要求的培训。

软件一有关组的例子包括：

- 软件质量保证组，
- 软件配置管理组，和
- 文档支持组。

培训的例子包括：

- 缺陷预防方法，
- 作业起动会议的开法，
- 原因分析会议的开法，和
- 统计方法（例如因 / 果图和 Pareto 分析）。

参着培训大纲关键过程区域。

执行的活动

活动 1

软件项目制定和维护一个有失其缺陷预防活动的计划。

该计划：

1. 确定将开展的缺陷预防活动（例如作业起动会议和原因分析会议）。
2. 规定缺陷预防活动的日程表。
3. 包括所指派的职责和所要求的资源——包括人员和工具。
4. 经历同行评审。

参阅同行评审关键过程区域。

活动 2 在一个软件作业开始时，承担此作业的群组的成员聚在一起准备该作业的活动和有关的缺陷预防活动。

举行起动会议，使该组成员熟悉过程实施的细节，以及对过程的任何最新更改。

这些起动会议的内容包括：

1. 适用于该作业的软件过程、标准、规模、方法和工具，重点放在最近的更改上。作为评价以前原因分析会议所作出建议的一项实验，可能进行更改。
2. 该作业所要求的且可得到的输入。
3. 要产生的输出，如果可能用例子说明。
4. 拟用来评价输出的方法。
5. 拟用来验证对软件过程的遵守情况的方法。
6. 在当前阶段通常会作出的或引入的差错的清单，以及推荐的预防这些差错的措施。
7. 群组的任命。
8. 作业日程表。
9. 作业和软件项目的软件产品质量目标。

参看软件质量管理关键过程区域。

活动 3

按照已文档化的规程进行原因分析会议。

该规程一般规定：

1. 每个执行软件作业的群组均召开原因分析会议。
 - ✍✍ 在作业完成后紧接着召开原因分析会议。
 - ✍✍ 在软件作业期间当已发现的缺陷的数目表明需要附加的会议时，也召开会议。
 - ✍✍ 在软件产品交付给顾客后，当合适时召开定期的原因分析会议。
 - ✍✍ 对于长软件作业，当合适时，召开定期的过程中缺陷预防会议。

长作业的一个例子是依赖于努力等级的（a level-of-effort）顾客支持作业。

2. 由在举办原因分析会议方面经过培训的人领导此会议。
3. 鉴别和分析缺陷以确定其根本原因。

确定根本原因的方法的一个例子是因果图。

4. 将缺陷归入根本原因的各个类别中。

缺陷根本原因的类别的例子有：

- 不恰当的培训，
- 通信中断，
- 没有说明问题的全部细节，和
- 在人工手续（例如，打印）中出错。

5. 制定为预防将来出现已识别出的缺陷和类似的缺陷而建议的措施并写成文档。

建议的措施的例子包括对以下各项的修改：

- 过程，
- 培训
- 工具，
- 方法，
- 通信，和
- 软件工作产品。

6. 对缺陷的一般原因加以识别并写成文档。

一般原因的例子包括：

- 在引用某个系统功能时出现的频发的差错，和
- 在软件单元的某个相关组中出现的频发的差错。

7. 将会议结果记录在案，供组织和其它项目用。

活动 4

受指派协调缺陷预防活动的每个群组定期相聚，评审和协调得自原因分析会议的措施建议的实施情况。

所涉及的群组可以是在组织层上的，也可以是在项目层上的。

这些群组：

1. 评审原因分析会议的结果和选择将采纳的措施建议。

2. 评审组织中其它协调缺陷预防活动的群组已经分派给它们的措施建议，并挑选出将采纳的措施建议。

3. 评审组织中其它群组所采取的措施以便评估这些措施是否适用于它们的活动和过程。

4. 对措施建议进行初步分析并设置其优先级。

优先级一般是不严格的，基于对以下各项的理解：

- 缺陷的原因，
- 不解决这些缺陷所蕴含的问题，
- 实施旨在预防这些缺陷的过程改进的成本，和
- 预期的对软件质量的影响。

用于设置措施建议的优先组的技术的一个例子是 Pareto 分析。

5. 当合适时在组织的另一个层次上再给群组重新安排措施建议。

6. 将决策的理论根据记入文档，并将决策和其根据提供给措施建议的提议者。

7. 为实施措施建议的行动条款分配职责。

✍✍ 行动条款的实施包括对本群组的权限范围内的活动作出立即的更改和为其它的更改作安排。

✍✍ 通常群组的成员实施行动条款，但在某些情况下，群组也能安排其它人去实施一项行动条款。

8. 评审缺陷预防实验的结果，并当合适时，采取措施将成功实验的结果纳入项目或组织的其余部分。

缺陷预防实验的例子有：

- 使用一个暂时的经修改的过程，和
- 使用一个新工具。

9. 跟踪措施建议和行动条款的状态。

10. 当合适时，对组织标准软件过程和项目定义软件过程的软件过程改进建议建立文档。指定措施建议的提议者作为软件过程改进建议的提议者。

参阅过程更政管理关键过程区域的活动 5 以便找到包括处理软件过程改进建议的实践。

11. 在行动条款结束之前，评审和验证已完成的行动条款。

12. 保证那些在预防缺陷上的重要工作和成功得到公认。

活动 5

在所有群组协调缺陷预防的活动中均将缺陷预防数据记入文档并对其进行跟踪。

1. 将在原因分析会上确定的措施建议写成文档。

在措施建议的描述中包含的数据的例子有：

—措施建议的发起者，

—缺陷的描述，

—缺陷原因的描述，

—缺陷原因的类别，

—缺陷引入的阶段，

—缺陷被识别出的阶段，

—措施建议的描述，和

—措施建议的类别。

2. 将措施建议中建议的行动条款写入文档。

行动条款的描述中包含的数据的例子有：

—负责实施的人，

—对受它影响的区域的描述，

—不断收到有关它状态报告的个人，

—其状态的下一次评审日期，

—关键决策的依据，

—实施行动的描述，

—识别缺陷和改正它所花费的时间和成本，和

—不修复此缺陷带来的损失。

3. 对缺陷预防数据进行管理和控制。

“进行管理和控制”意味着在给定时间（过去或现在）使用的工作产品的版本是已知的（即版本控制），而且以受控的方式引进更改（即更改控制）。

如果希望有比“进行管理和控制”所蕴含的更高程度的控制，则工作产品可置于配置管理的完备的纪律之下，正如在软件配置管理关键过程区域中所描述的。

活动 6

按照已文档化的规程采纳由缺陷预防措施所导致的对组织标准软件过程的修改。

参着组织过程定义关键过程区域的活动 1 以便找到包括组织标准软件过程的实践。

活动 7

按照已文档化的规程采纳由缺陷预防措施所导致的对项目定义软件过程的修改。

参看集成软件管理关键过程区域的活动 2 以便找到包括项目定义软件过程的实践。

活动 8

软件工程组和软件一有关组的成员定期收到有关组织的和项目的缺陷预防活动的状态和结果的反馈信息。

该反馈提供：

1. 主要缺陷类别的摘要。

2. 缺陷在主要缺陷类别中的频率分布。

3. 为处理主要缺陷类别所采取的重大的创新和措施。

4. 措施建议和行动条款的概括的状态。

提供这项反馈的方法的例子有：

- 电子布告栏，
- 业务通信，
- 信息流会议（informat. onflowmeetings）。

测量和分析

测量 1

进行测量并将测量结果用于确定缺陷预防活动的状态。

测量的例子包括：

- 缺陷预防活动（例如，举办原因分析会议和实施行动条款）的成本，累计地计算；
- 识别缺陷和纠正它们所化费的时间和成本，与不更正这些缺陷的估计损失作比较；
- 刻面，它们测量所建议的、未解决的和已完成的行动条款的数目；
- 总计每个阶段插入的缺陷数目，并与类似产品的发表的数目相比较，和

验证实施

验证 1

高级管理者定期参与评审组织的缺陷预防活动。

高级管理者定期评审的主要目的是在合适的抽象层次上并以及时的方式了解和洞察

软件过程活动，评审间隔应满足组织的需要，只要已存在报告例外情况的合适机制，间隔可以长。

这些评审包括：

- 1．主要缺陷类别和在这些类别中缺陷的频率分布的摘要。
- 2．主要措施类别和在这些类别中措施的频率分布的摘要。
- 3．为解决主要缺陷类别所采取的重要措施。
- 4．有关所建立的、未解决的、已完成的行动条款的概括的状态。
- 5．对于缺陷预防活动的有效性及由缺陷预防活动所得到节省的概述。
- 6．已完成的缺陷预防活动的实际成本和所计划的缺陷预防活动的预测成本。

验证 2

项目经理既定期地也事件驱动地参与评审软件项目的缺陷预防活动。

参看软件项目跟踪和监督关键过程区域的验证 2 以便找到包括项目管理者监督评审的典型内容的实践。

验证 3

软件质量保证组评审和（或）审计缺陷预防的活动和工作产品并报告其结果。

参看软件质量保证关键过程区域。

至少，评审和（或）审计要验证：

- 1．软件经理和技术人员受到关于其缺陷预防任务的培训 1。
- 2．适当地举行作业开始会议和原因分析会议。
- 3．评审措施建议和实施行动条款的过程得到遵循。

技术改革管理

等级 5（优化的）的一个关键过程区域

技术改革管理的目的是识别出新技术（即工具、方法和过程），并以有序的方式将其引进到组织中去。

技术改革管理包括鉴别、选择和评价新技术，并将有效的技术采用到组织中去。其目的是改进软件质量、提高生产率和缩短产品开发周期。

组织建立一个组（例如软件工程过程组或技术支持组），它与软件项目一起引进和评价新技术并管理对现有技术的改革。特别注重那些多半能改善组织标准软件过程能力的技术改革（有关标准软件过程的描述在组织过程定义关键过程区域中）。

通过保持对软件一有关的技术创新的了解和对它们进行系统的评价与实验，组织选择合适的技术用以改进其软件的质量和其软件活动的效率。在新的和未经证明的技术引入到标准实践中去之前，进行先导性试验以评估它们。当合适时，在组织管理者的恰当支持下，将选出的技术纳入组织的标准软件过程和当前的项目。

按照在过程更改管理关键过程区域中所描述的那样，处理由这些技术改革所造成的对组织标准软件过程（正如在组织过程定义关键过程区域所描述的）和项目定义软件过程（正如在集成软件管理关键过程区域所描述的）的更改。

目标

目标 1

技术改革的采纳是有计划的。

目标 2

评价新技术以确定他们对质量和生产率的作用。

目标 3

将合适的新技术引入到全组织的标准实践中。

执行约定

约定 1

组织遵循书面的有关改善其技术能力的方针。

该方针一般规定：

1. 制定技术改革管理的目标并写成文档。
2. 一个巴文档化的计划阐述技术改革管理的目标。

约定 2

高级管理者支持组织的有关技术改革管理的活动。

高级管理者：

1. 帮助确定阐述组织有关产品质量、生产率和产品开发周期目标的战略。

2. 当合适时，帮助确定阐述顾客和最终用户需要和愿望的战略。

译者注：愿文 trark 为 transfer 之误。

这些实践中所指的最终用户是顾客指定的最终用户或最终用户的代表。

3. 和组织的经理们协调，确定他们实现组织战略的目标和方法。

4. 对技术改革管理工作作出全组织都看得见的承诺。

5. 制定关于投资、人员配置和其它资源的长期计划和约定。

约定 3

高级管理者监督组织的技术改革管理活动。

高级管理者：

1. 帮助制定技术改革管理的方针，并评审和批准这些方针。

2. 安排用于技术改革管理活动的资源。

3. 帮助建立组织的战略和目标与技术改革管理的战略之间的关系。

4. 参加制定技术改革管理的计划。

✍✍ 高级管理者在组织的所有合适的层次上协调有关技术改革管理的需求和问题。

✍✍ 高级管理者和组织的经理们协商以保证经理和职员的支持和参与。

执行能力

能力 1

存在负责组织的技术改革管理活动的组。

一个组是负责一组作业或活动的部门、经理和个人的集会。组的规模可以变化：从一个受指派的非全日制的单个人，到几个从不同部门指派来的非全日制的个人，到几个全日制的个人。建立一个组时考虑的因素包括指派的作业或活动、项目的规模、组织机构和组织文化。某些组，例如软件质量保证组，集中注意力于项目的活动，而其它的组，例如软件工程过程组，集中关注全组织的活动。

1. 该组或者是负责组织软件过程活动的组（例如软件工程过程组）的一部分，或者其活动与后者紧密地配合。

2 该组协调和帮助：

✍✍ 探查能应用新技术的潜在区域；

✍✍ 选择和规划新技术；

✍✍ 获得、安装和定制新技术；

✍✍ 在组织内部交流和协调有关的研究和开发活动；

日与技术供应商交流有关问题和改进意见。

能力 2

为负责组织技术改革管理活动的组的建立和人员配置提供足够的资源和投资。

1. 该组可以得到在专门领域有特长的有经验的人员，帮助评价、策划和支持有关技术改革管理的倡议。

专门领域的例子包括：

— 工作站，

— 计算机硬件，

— 软件应用，

— 计算机辅助软件工程（CASE）技术，

— 软件度量，

— 形式化方法，和

一编辑语言。

2. 使得支持技术改革管理的工具合用。

支持工具的例子包括：

一工作站，

一数据库程序，和

一对在线技术数据库的预约（subscription）。

能力 3

存在对于收集和分析那些评价技术改革所需数据的支持。

该支持包括以下几方面的能力：

1. 自动地记录选定的过程和产品的数据。

2. 支持数据分析。

3. 显示选定的数据。

以能恰当地传递信息内容的格式（例如图形显示）来描述数据分析的结果。

能力 4

可以得到有关软件过程和软件工作产品的合适的数据，以支持为评价和选择技术改革所作的分析。

过程和产品的数据的例子包括：

一按项目、过程阶段、所用的工具和方法、程序类别、程序修改程度等等列出的资源消耗和生产率；

一按项目、每个项目的过程阶段、程序类别、程序规模、程序修改程度等等列出的进度时间；

一同行评审数据，包括缺陷数据和评审有效性；

~ 缺陷数据，显示缺陷被发现的阶段、消除缺陷的阶段、缺陷的类别、原因、严重性及为修复缺陷所用的时间和工作量；

一更改活动，包括所生成的代码量、所生成的文档量、等等；

一有关修复缺陷的活动的的数据，包括缺陷的识别、完成缺陷修复的产品版本、以及对于在实施每个缺陷修复时又引入的缺陷的识别；和

一按项目、产品类型、特定产品、和特定于产品（例如程序模块）列出的缺陷密度。

能力 5

负责组织的技術改革管理活动的组的成员接受执行这些活动所要求的培训。

培训的例子包括：

一组织的标准软件过程，

一技术转换和改革管理，

一软件过程改进，

一组织所用的工具和方法，

一组织可得到的分析和支持设施，和

一统计质量控制原理。

参着培训大纲关键过程区域。

执行的活动

活动 1

组织制定和维护技术改革管理的计划。

该计划：

1. 包括所安排的职责和所要求的资源，包括人员和工具。
2. 确立长期的改进组织标准软件过程使之自动化以增强组织的市场准位的技术战略。
3. 确定在开展组织的技术改革管理活动时应遵循的规程。
4. 描述引进针对组织和项目的特定需要的新技术的方法。

✍✍ 鉴别可能需要技术改革的过程区域。

✍✍ 确定辨认技术改革机会的方法。

✍✍ 鉴别专门规划的成候选的技术。

✍✍ 在合适处对计划引进的技术的生命跨度——从引进到替代——进行估计。

✍✍ 对自行研制还是买进的权衡研究建立文档。

✍✍ 规定评估未经证明的候选技术的方法。

✍✍ 规定采购和安装的规程。

✍✍ 规定初始培训、后续培训和咨询支持。

5. 经受同行评审。

参看同行评审关键过程区域。

6. 由受影响的经理评审。

活动 2

负责组织技术改革管理活动的组和软件项目一起识别技术改革的区域。

该组：

1. 征集对于技术改革的建议。

2. 识别可以得到的、可能适合组织和项目需要的新技术。

✍✍ 进行定期的搜索以识别出市场上可得到的、能满足既定需要和预期需要的技术。

✍✍ 作出系统的努力以保持对领先的相关技术工作和新技术发展趋势的了解。

✍✍ 作出系统的努力去评审外界所用的技术并将其与组织内所用的对应技术作比较。

✍✍ 鉴别出新技术已成功地得到运用的领域，收集和评审使用这些技术的经验数据和文档。

3. 评价新技术以确定它们对组织和项目的当前和未来需要的适用性。

活动 3

不断向软件经理和技术人员通报新技术的情况。

1. 当合适时，散发有关新技术的信息。

2. 当合适时，散发有关在组织的某些部位已使用的先进技术的信息。

3. 当合适时，散发有关正被引进到组织中的技术的状态信息。

活动 4

负责组织技术改革管理的组系统地分析组织的标准软件过程，以便识别出需要新技术或能从新技术中获益的区域。

该组：

1. 分析组织的标准软件过程，以便确定新技术起作用最大的区域。

2. 识别有用的技术改革和确定这些改革的费效情况。

3. 确定所识别出的技术与组织标准软件过程的关系。

4. 当合适时，定性地和定量地确定技术改革的预期结果。

5. 确定对每项潜在技术改革进行先导性试验的必要性。

6. 确定候选新技术的优先级。

7. 将分析活动的结果写成文档。

活动 5

按照已文档化的规程，为组织和软件项目挑选和获取技术。

该规程一般规定：

1．将申请获得新技术的请求写入文档。

✍✍ 对干预计那些其化费超过预先规定水平的新技术，要求有管理者的批准。

2．对潜在的技术改革进行初步的费效分析。

3．采用预先确定且经批准的选择判据去识别最高的潜在效益。

4．确定对已选定的技术改革的要求和计划，并写成文档。

✍✍ 在可行处，估计该技术的预期生存期和替代 / 升级的计划。

✍✍ 在合适处，进行权衡研究，并对其进行评审和写成文档，以便确定该技术应自行开发还是从外部购买。

✍✍ 在合适处，该计划考虑在试点处安装新技术以便确定其有效性和经济效益。

✍✍ 由受影响的组和负责技术改革管理活动的组的经理评审这些要求和计划。

活动 6

在合适处，在将新技术引入标准实践之前，为改进技术作先导性试验。

1．进行先导性试验以确定未试用过的或先进的技术的可行性和经济效益。

2．将先导性试验工作计划写成文档。

✍✍ 该计划包括先导性试验的目标、评价准则和活动。

3．受影响的组的经理们评审和批准进行先导性试验的计划。

受影响的组的例子有：

一软件工程组（包括所有小组），

一软件估计组，

一系统工程组，

一系统测试组，

一软件质量保证组，

一软件配置管理组，

一合同管理组，和

一文档支持组。

4．负责技术改革管理活动的组对实施先导性试验的项目提供咨询和帮助。

5．先导性试验在与开发和维护环境有关的环境中进行。

6．收集、分析先导性试验的结果，并写成文档。

✍✍ 将先导性试验期间得到的经验教训和遇到的问题写成文档。

✍✍ 估计新技术在组织中广泛应用的得益和影响。评估这些估计中的不确定性。

✍✍ 作出决定，是否终止先导性试验继而广泛实施该技术，还是重新计划和继续先导性试验。

活动 7

按照已文档化的规程，将合适的新技术纳入组织的标准软件过程。

参看组织过程定义关键过程区域的活动 1 和过程更改管理关键过程区域的活动 5，以便找到包括组织标准软件过程更改的实践。

活动 8

按照巴文档化的规程，将合适的新技术纳入项目定义软件过程。

参看集成软件管理关键过程区域的活动 2，以便找到包括项目定义软件过程修订的实践。

测量和分析

测量 1

进行测量并将测量结果用于确定组织的技术改革管理活动的状态。

测量的例子有：

- 整个技术改革活动，包括改革的数目、类型和规模。
- 实施技术改革的效果，和目标相比较。

验证实施

验证 1

高级管理者定期参与评审那些组织的技术改革管理活动。

高级管理者定期评审的主要目的是在合适的抽象层次上并以及时的方式了解和洞察软件过程活动。评审间隔应满足组织的需要，只要已存在报告例外情况的合适机制，间隔可以长。

这些评审：

- 1．总结技术改革管理活动。
- 2．识别需要的战略变化。
- 3．导致问题的解决。
- 4．导致在合适时批准对技术改革管理计划的修订。

验证 2

软件质量保证组评审和（或）审计技术改革管理的活动和工作产品，并报告其结果。

参考软件质坦保证关键过程区域。

至少，评审和（或）审计查证：

- 1．技术改革管理的计划。
- 2．选择、购买和安装新技术的过程。

过程更改管理

等级 5（优化的）的一个关键过程区域

过程更改管理的目的是本着改进软件质量、提高生产率和缩短产品开发周期的宗旨不断改进组织中所用的软件过程。

过程更改管理包括定义过程改进目标，并且在高级管理者的支持下，在连续的基础上前瞻地（proactively）和系统地识别、评价和实施对组织标准软件过程和项目定义软件过程的改进。

制定培训和鼓励性计划，使得组织中的每个人能够并鼓励他们参加过程改进活动。识别改进的机会并评价其对组织的可能回报。在将过程更改纳入标准实践之前，进行先导性试验以评估这些过程更改。

当软件过程改进经批准用于标准实践时，在合适时修订组织标准软件过程和项目定义软件过程、在组织过程定义关键过程区域中找到修订组织标准软件过程的实践，在集成软件管理关键过程区域中找到修订项目定义软件过程的实践。

目标

目标 1

连续的过程改进是有计划的。

目标 2

组织的软件过程改进活动的参与面是全组织范围的。

目标 3

组织标准软件过程和项目定义软件过程被连续不断地改进。

执行约定

约定 1

组织遵循书面的实施过程改进的方针。

该方针一般规定：

- 1．组织有定量的、可测的软件过程改进目标，并对照这些目标跟踪性能。
- 2．组织过程改进的方向是改善产品质量、提高生产率和缩短产品开发周期。
- 3．要求组织的全部职员和经理均参与改进软件过程的工作。

承认有技能的和有积极性的人是主要的过程改进资源。

约定 2

高级管理者主办组织的软件过程改进活动。

高级管理者：

- 1．制定组织关于过程改进的长期目标和计划。
- 2．安排过程改进活动所需的资源。

3. 与软件管理者协调，保证他们有合理的但又是进取的过程改进目标和旨在实现这些目标的有效的过程改进计划。
4. 对照目标监控过程改进行为。
5. 在面临产品危机时，对过程改进保持一致的优先关注。
6. 保证过程改进问题得到及时解决。
7. 奖励参加过程改进活动的雇员。

执行能力

能力 1

提供足够的用于软件过程改进活动的资源和投资。

1. 为下列活动分配资源：

✍ 领导、指导和支持过程改进活动；

✍ 保持过程改进的记录；

✍ 开发、控制和传播过程更改；和

✍ 建立和运用行政管理职能和人类资源的职能，以便进行那些保持雇员更积极地参与所必须的通信、激励和表彰活动。

2. 可以得到在定义和分析软件过程方面有专门知识的有经验的个人，用来在组织的过程改进活动中帮助组织。

3. 使得支持过程改进的工具合用。

支持工具的例子有：

—统计分析工具，

—数据库系统，

—过程自动化工具，和

—过程建模工具。

能力 2

软件经理接受所要求的在软件过程改进方面的培训。

培训的例子有：

—对技术上和组织上的更改进行管理，

—群组的建立，和

—适用于连续过程改进的群组工作技能。

参阅培训大纲关键过程区域。

能力 3

软件工程组和其它软件一有关组的经理和技术人员接受所要求的在软件过程改进方面的培训。

软件一有关组的例子包括：

—软件质量保证组，

—软件配置管理组，和

—文档支持组。

培训的例子有：

—质量和过程改进原理，和

—提议过程改进的规程。

参阅培训大纲关键过程区域。

能力 4

高级管理者接受所要求的在软件过程改进方面的培训。

培训的例子有：

- 基准测定（benchmarking）和比较性评价，
- 过程改进原理，
- 过程改进目标的设置和跟踪，和
- 在连续的过程改进的环境中动力和群组的建立。

执行的活动

活动 1

制定软件过程改进大纲，该大纲授权组织的成员改进组织的过程。

活动 2

负责组织软件过程活动的组（例如，软件工程组）协调软件过程改进活动。

参着组织过程焦点关键过程区域以便找到包括负责组织软件过程改进活动的组的实践。
这个组：

- 1．确定组织的有关软件过程性能的目标和测量计划。
- 2．与高级管理者一起评审组织的过程性能目标以便得到他们的承认。
- 3．参加确定组织对过程改进的培训需求，支持编写与介绍培训课程材料。

参着培训大纲关键过程区域。

- 4．规定和维护处理过程改进建议的规程。
- 5．评审软件过程改进建议和协调有关这些建议的行动。
- 6．跟踪过程改进活动的状态，成就和参与情况，定期地向高级管理者报告结果。
- 7．协调和跟踪对组织标准软件过程的更改。
- 8．定义、建立和保持过程改进记录。

活动 3

组织按照巴文档化的规程制定和维护软件过程改进计划。

参阅组织过程焦点关键过程区域的活动 2，以便找到包括组织软件过程改进计划的实践

。

这个规程一般规定：

- 1．软件过程改进计划基干：

- ✍ 组织的经营和战略运作计划，和
- ✍ 顾客满意程度的标记。

- 2．软件过程改进计划经历同行评审。

参看同行评审关键过程区域。

- 3．受影响的经理评审软件过程改进计划。
- 4．对软件过程改进计划进行管理和控制。

“进行管理和控制”意味着在给定时间（过去或现在）使用的工作产品的版本是已知的（即版本控制），而且以受控的方式引进更改（即更改控制）。

如果希望有比“进行管理和控制”所蕴含的更高程度的控制，则工作产品可置于配置管理的完备纪律之下，正如在软件配置管理关键过程区域中所描述的。

活动 4

依照软件过程改进计划进行软件过程改进活动。

该计划包括：

- 1．所要求的资源，包括人员和工具。

2. 有最高优先组的过程改进区域。
3. 有关软件过程性能和改进的可测的短期目标和长期目标。
4. 致力于特定过程区域改进的群组和其任务。

群组的例子有；

- 工作组，
- 过程行动群组，和
- 技术委员会。

5. 关于以下各项的规程：

- ✍ 高级管理者监督软件过程改进活动；
 - ✍ 软件经理策划和协调软件过程改进活动；
 - ✍ 个人和群组识别、评价和引进合适的软件过程改进；和
 - ✍ 群组展开对规定过程区域的软件过程改进。
6. 为保持连续过程改进所要求的行政管理计划和支持计划。
 - ✍ 包括适当的旨在鼓励参与和促进软件过程改进活动的行政管理规模。
 - ✍ 在监督和评审软件过程改进活动中包括行政管理人员。
 - ✍ 认识到雇员对连续过程改进的作用和贡献。

活动 5

按照已文档化的规程处理软件过程改进建议。

该规程一般规定：

1. 提交软件过程改进建议。

软件过程改进建议可在任何时间提交，并可阐述软件过程的任何区域。

软件过程改进建议的来源的例子有：

- 软件过程评估的发现和和建议，
 - 组织的软件过程改进目标，
 - 对有关顾客问题和顾客满意度的数据的分析，
 - 对项目性能与软件质量目标 and 生产率目标相比较的数据的分析，
 - 过程基准测定的结果，
 - 过程 / 作业自动化的潜力，
 - 对有关缺陷原因的数据的分析，
 - 测量到的软件过程活动的有效性，
 - 一些被采用后获得成功的软件过程改进建议的例子，和
 - 当合适时，对以前提交的软件过程改进建议的反馈意见。
2. 评价每项软件过程改进建议；对是否实施该建议作出决定，并将决策依据写成文档。
 3. 确定每项软件过程改进建议的预期效益。

预期得益区域的例子包括：

- 生产率，
 - 质量，
 - 周期时间，
 - 顾客或最终用户满意度的其它标记，和
 - 任何其它的内部因素。
4. 确定所选出的软件过程改进建议实施的优先级。
 - ✍ 将注意力集中于具有高优先级的软件过程改进建议。
 5. 安排和计划从建议中形成的软件过程改进行动的实施。
 6. 将要求实质性工作的软件过程改进行动安排给负责实施的群组。

实质性工作的例子包括要求进行新技术先导性试验的或其它重大改变的改进。

建立关注特定软件过程区域的群组。

协调适于作先导性试验的行动。

群组的例子包括：

—工作组，

—过程行动群组，和

—技术委员会。

7．跟踪每项软件过程改进建议的状态。

8．识别那些对其响应的时间异乎寻常地长的软件过程改进建议，并作出反应，

9．经判断某些软件过程对产品质量或生产率有重大影响或能显著地改变顾客和最终用户的满意度。这些软件过程更改之前，必须经过适当的管理者的评审和批准。

10．已完成的软件过程改进行动在结束之前，必须得到评审、验证和批准。

11．软件过程改进建议的提议者收到：

✍✍ 对其建议的迅捷承认，和

✍✍ 对其建议的处置的通知。

活动6

组织的成员积极参加群组，对所规定的过程区域展开软件过程改进。

1．对每个这样的过程改进群组提供资金，并且对其活动加以规划和安排进度。

2．对每项过程改进工作建立目标；在可能处，这些目标是定量地规定的。

3．受影响的组和定义及维护那些受到影响的组描述的组的经理们批准该计划。

受影响组的例子包括：

—软件工程组〔包括所有小组，例如软件设计小组，还包括软件作业领导〕，

—软件估计组，

—系统工程组，

—系统测试组，

—软件质量保证组，

—软件配置管理组，

—合同管理组，和

—文档支持组。

活动了在合适时，在将软件过程改进纳入标准实践之前，在先导性试验基地设置软件过程改进，以便确定其效益和有效性。

1．在先导性试验中为了优化所建立的过程改进的实施工作，对过程改进建议进行调整，并用文档记载这些调整。

2．将经验教训和遇到的问题写入文档。

3．估计在组织中广泛使用过程改进的效益、风险和影响，并评估在这些估计中的不确定性。

4．作出决策是终止先导性试验，继之在大范围内实施改进，还是重新计划和继续先导性试验。

活动8

当决定将软件过程改进引入到标准实践时，依照已文档化的规程实施改进工作。

该规程一般规定：

1．确定为支持软件过程的主要更改所需要的资源，并给它们提供资金。

2．将测量和跟踪软件过程性能更改的数据采集策略写成文档、进行评审，并得到一致的认可。

✍✍ 由负责实施受该更改影响的软件过程的个人认同这个策略。

✍✍ 当合适时，配备支持工具去自动地记录所希望的数据。

3．更新培训课程以反映当前的软件过程，在设置过程更改供广泛使用之前提供培训。

参着培训大纲关键过程区域。

4．在设置过程更改供广泛使用之前建立适合预期需求的咨询支持，当需要时继续支持。

5．将合适的过程更改纳入组织的标准软件过程。

参着组织过程定义关键过程区域的活动 1 以便找到包括组织标准软件过程的实践。

6．将合适的过程更改纳入项目定义软件过程。

参阅集成软件管理关键过程区域的活动 2 以便找到包括项目定义软件过程的实践。

活动 9

保持软件过程改进活动的记录。

1．保持有关软件过程的改进建议的倡议、状态和实施的信息。

2．提供简便的存取软件过程改进记录的手段。

3．保持历史数游，生成有关软件过程改进的报告。

记录和报告的例子包括：

一项目的生产率、质量和进度性能；

一程序的缺陷历史；

一组织的软件质量和生产率趋势；和

一软件过程开发和改进的成本、进度和生产率。

参阅组织过程定义关键过程区域的活动 5 以便找到包括组织软件过程数据库的实践，它是保持过程改进记录的可能机制中的一种。

活动 10

在事件一驱动的基础上，软件经理和技术人员收到有关软件过程改进活动的状态和结果的反馈信息。

该反馈信息提供；

1．主要软件过程改进活动的概要。

2．为了进行软件过程改进所采取的重大创新和行动。

3．有关已提交的、待解决的和已完成的软件过程改进建议的概括的状态。

提供反馈的方法的例子有；

一电子布告栏，

一业务通信，和

一信息流会议。

测量和分析

测量 1

进行测量并将测量结果用于确定软件过程改进活动的状态。

测量的例子包括：

一对每个过程区域提交的和实施的软件过程改进建议的数目；

一每个项目、组和部门提交的软件过程改进建议的数目。

一每个项目、组和部门所受到的奖励和表彰的数目和类型；

一用于处理软件过程改进建议的响应时间；

一每个报告周期所接收的软件过程改进建议的百分数；

一全部的更改活动，包括更改的数目、类型和规模；

- 一实施每项过程改进的效果与其规定目标相比较；
- 一组织的和项目的总体性能，包括有效性、质量和生产率，与其规定目标相比较；
- 一每个项目的总的生产率和软件质量的趋势；和
- 一与顾客满意度的指标有关的过程测量。

验证实施

验证 1

高级管理者定期参加评审软件过程改进的活动。

高级管理者定期评审的主要目的是在合适的抽象层次上并以及时的方式了解和报软件过程活动。评审间隔应该满足组织的需要，只要已存在报告例外情况的合适制，间隔可以长。

进行这些评审是为了：

- 1．总结参加过程改进活动的情况。
- 2．评估过程性能。
- 3．识别所需要的目标更改。
- 4．解决问题。
- 5．当合适时，批准对软件过程改进计划的修订。

验证 2

软件质量保证组评审和（或）审计软件过程改进的活动和工作产品，并报告其结果。

参阅软件质量保证关键过程区域。

至少，评审和（或）审计要查证：

- 1．组织的软件过程改进计划的准备工作。
- 2．软件过程改进建议的倡议、提交、评审、批准和策划实施的过程。
- 3．过程测量符合软件过程描述的程度和反映实际性能的程度。
- 4．对组织标准软件过程和项目定义软件过程的更改建立文档、评审、批准、控制和传播的过程。
- 5．软件过程改进活动被一致地测量和跟踪的程度。
- 6．实际软件过程改进性能实现计划和目标的程度。

附录

附录 A 参考文献附录 B 术语汇编

执行能力 (ability to perform) —— (见共同特点)

验收准则 (acceptance criteria) —— 为了被用户、顾客或其它指定实体接收, 一个系统或成分必须满足的准则。IEEE - STD 610I

验收测试 (acceptance testing) —— 为确定一个系统是否满足其验收准则和为使得用户能确定是否接收该系统而进行的正式测试。(IEEE - STD - 610)

活动 (activity) —— 任何 ~ 个为实现某个目的而采取的步骤或所执行的功能, 既可是脑力的, 也可是体力的。活动包括经理和技术人员为完成项目和组织的作业所进行的全部工作。(对照参看作业)

执行的活动 (activities performed) —— (见共同特点)

行动项 (action item) —— (1) 已安排给某个个人或小组处理的事务清单中的一个单元。
(2) 一项已被采纳的行动建议。

行动建议 (action Proposal) —— 作为缺陷预防活动的结果, 一项已文档化的对过程或过程一有关项的更改建议, 它将预防将来再出现已识别出的缺陷。(也可参看软件过程改进建议)

分配需求 (allocated requirements) —— (见分配给软件的系统需求)。

应用领域 (application domain) —— 一组有界限的相关系统 (即处理一个特定类型问题的那些系统)。在一个应用领域内的开发和维护工作通常要求专门的技能和 (或) 资源。例如工资和人员系统、指挥和控制系统、编译程序和专家系统。

评估 (assessment) —— (见软件过程评估)

审计 (audit) —— 对一个成一组工作产品所作的独立的考察, 评估它们与规格说明、标准、合同协议或其它判据的符合性。IEEE - STD - 610J

基线 (baseline) —— 已经过正式评审和认可, 作为以后进一步开发的基础, 并且只有通过正式的更改控制规程才能进行更改的规格说明或产品。【IEEE - STD - 610J】

基线配置管理 (baseline configuration management) —— 建立经正式评审和认可并作为进一步开发工作的基础的基线。某些诸如软件设计和代码这样的软件工作产品应该有在预先确

定点上建立的基线, 并且对这些项应该施加严格的更改控制过程。当与顾客打交道时, 这些

基线提供控制和稳定性。(也可参看基线管理。)

基线管理 (baseline management) —— 在配置管理中, 运用技术上的和行政上的管理去指

定一些文档和对这些文档的更改, 这些文档在一个配置项的生存期内的某些特定时刻, 正式

标识出和建立起基线。IEEE - STD 610J

测定基准 (benchmark) —— 一项标准, 对照它能进行测量或比较。【IEEE - STD - 610J】

译注: measurement 的翻译有两种, 当与“软件”或“质量”相这时用度量, 与“硬件”或“过程”相这时用测量。

投标者 (bidder) —— 一个个人、合作伙伴、公司或协会, 他已提交一项提议, 并且是待签署的设计、开发和 (或) 制造一个或多个产品的合同的候选人。

能力成熟度模型 (capability maturity model) —— 对软件组织进化阶段的描述, 随着软件组织定义、实施、测量、控制和改进其软件过程, 它们经过这些阶段逐步前进。该模型使得

确定当前过程能力的工作和识别对软件质量和过程改进最关键的问题变得容易,从而对选择过程改进战略提供指导。

原因分析 (causal analysis) ——为确定缺陷的根源所作的缺陷分析。

原因分析会议 (causal analysis meeting) ——在完成一特定任务后为分析任务完成期间所暴露的缺陷而举行的会议。

CMM ——能力成熟度模型的首字母缩略词。

约定 (commitment) ——自由地接受的、可视的、期待各方遵守的协议。

执行约定 (commitment to perform) —— (见共同特点。)

缺陷的一段原因 (common cause of a defect) ——一种缺陷原因,这种缺陷原因是一个过程或系统固有的。一般原因影响到过程的每个输出和工作在此过程中的每个人。(对照参着特殊原因。)

共同特点 (common features) ——用以进一步细分 CMM 关键过程区域的类别。共同特点是指示一个关键过程区域的实施和规范化是否是有效的、可重复的、和持久的一些属性。CMM 的共同特点如下;

- ✍ 执行约定 ——组织必须采取的保证过程得以建立和持续下去的行动。执行约定一般涉及到组织方针的建立和高级管理者的支持。
- ✍ 执行能力 ——为了适当地实施软件过程,项目或组织中必须存在的先决条件。执行能力一般涉及资源、组织机构和培训。”
- ✍ 执行的活动 ——对于那些为实施一个关键过程区域所必须的角色和规程的描述。执行的活动一般涉及制定计划和规程,进行工作,跟踪它和必要时采取改正行动。
- ✍ 测量 和分析 ——对过程进行测量和对测量结果进行分析的需求的描述。测量和分析一般包括可能采取的旨在确定执行的活动的状态和有效性的测量例子。
- ✍ 验证实施 ——保证活动按照已建立的过程进行的措施。进行验证一般包括管理者和软件质量保证组作的评审和审计。

配置 (configuration) ——在配置管理中,软件或硬件所具有的正如在技术文档中所陈述的或产品所实现的那些功能特征和物理特征、[IEEE - STD-610]

配置控制 (configuration control) ——配置管理的一个构成部分,它包括在正式建立配置项的配置标识之后对它们所作更改的评价、协调、批准或拒绝和实施。[IEEE - STD-610]

配置标识 (configuration identification) ——配置管理的一个构成部分,它包括为系统选择配置项,并在技术文档中记录其功能特征和物理特征。[IEEE - STD-610]

配置项 (configuration item) ——指定为配置管理对象且在配置管理过程中作为单个实体处理的硬件集合、软件集合或硬软件集合。[IEEE - STD-610]

配置管理 (configuration management) ——对以下各项运用技术上和行政上的管理和监视的一种学科:对一个配置项的功能特征和物理特征进行标识并写成文档;对这些特征的更改进行控制;对重政处理过程和实施状态进行记录和报告;以及对规定需求的符合性进行验证。

[IEEE - STD-610]

配置管理库系统 (configuration management library system) ——存取软件基线库内容的工具和规程。

配置单元 (configuration unit) ——可放入配置管理库系统和从库中检索的一个配置项或成分的最低层次的实体。

一致性 (consistency) ——在系统或其成分的文档或部件中,一致、标准化和无矛盾的程度。[IEEE - STD - 610]

偶然性因素 (contingency factor) ——对规模、成本或进度计划的调整 (增加),以便计

及由于不完全的规格说明、在估计应用领域上缺乏经验等等所导致的很可能的对这些参数的过低估计。

合同条款和条件 (contract terms and conditions) ——合同规定的法律、金融和行政管理方面。

关键计算机资源 (critical computer resource) ——据认为将是项目风险源的计算机资源的参数，因为对这些资源的潜在需求有可能超出可得到的量。例子包括目标计算机的存储容量和行主计算机的磁盘空间。

关键路径 (critical path) ——为保持整个项目按进度进行必须按计划完成的项目的一系列的相互依赖的作业。。

顾客 (customer) ——负责接收产品和批准付款给开发组织的个人或组织。

缺陷(defect) ——系统或系统成分中的能造成它们无法实现其被要求有的功能的缺点。如果在执行过程中遇到缺陷，它可能导致系统的失效。

缺陷密度 (defectdensity) ——在一个产品中识别出的缺陷数目除以产品成分的规模 (用该产品的标准度量术语表示)。

缺陷预防 (defectprevention) ——与识别缺陷或潜在缺陷并预防它们被引入到产品有关的活动。

缺陷根源 (defectrootcause) ——使缺陷得以引入的基础原因 (例如过程不健全)。

已定义等级 (definedlevel) —— (见成熟度等级。)

已定义的软件过程 (definedsoftwareprocess) —— (见项目定义软件过程。)

相关项 (dependencyitem) ——一个个人或组必须向第二个个人或组提供的产品、行动或信息，使得第二个个人或组能完成所计划的任务。

开发配置管理 (developmentalconfigurationmanagemt) ——运用技术上的和行政上的管理去指定或控制那些软件和其相关的技术文档，它们定义一个软件工作产品在开发期间内不断进化的配置。开发配置管理处处在开发者的直接控制之下。置于开发配置管理之下的配置项不是基线，虽然在其开发的某些点上，它们可能被基线化并置于基线配置管理之下。

偏差 (deviation) ——相对于适当的规范、计划、标准、规程或受评审的变量来说是显著的或明显的偏离。

已文档化的规程 (documentedprocedure) —— (见规程)。

有效过程 (effectiveprocess) ——可特征化为实用的、巴文档化的、已实施的、经培训的、已测量的和能改进的过程。(也可参见妥善一定义的过程。)

最终用户 (enduser) ——当系统安装在运行环境中时，那些将按照其预先设计的运行用途而使用该系统的个人或组。

最终用户代表 (enduserrepresentatives) ——挑选出的能代表最终用户整体的最终用户样本。

工程组 (engineeringgroup) ——代表一门工程学科的个人 (包括经理和技术人员) 的集合。工程学科的例子包括系统工程、硬件工程、系统测试、软件工程、软件配置管理和软件质量保证。

评价 (evaluation) —— (见软件能力评价)。

事件一驱动的评审或活动 (event - drivenreview / activity) ——基于项目中某个事件的出现而执行的评审或活动 (例如一项正式评审或一个软件生存周期阶段的完成) (参看定期评审成活动以作对照。)

发现 (findings) ——标识出调查范围内最重要的议题、问题或机会的一项评估、评价、审计或评审所作出的结论。

一线软件经理 (first - line software manager) ——对由软件工程师和其它相关职员构成

的一个单独的机构单位（例如一个部门或项目组）的人员配置和活动负有直接管理责任（包括提供技术指导和对人员和工资进行行政管理等功能）的经理。

正式评审（formal review）——一次正式的会议，在此会议上将产品提交给最终用户、顾客或其它有兴趣的各方，以得到评论和批准。它也可以是对项目的管理和技术活动及项目进程的评审。

功能（function）——为实现～组目的或终止一件事，个人或工具所进行的一组特别安排的或特别适宜于其角色的相关的行动。

目标（goals）——对一个关键过程区域中关键实践的概括，能用于确定一个组织或项目是否已经有效地实施该关键过程区域。目标表示每个关键过程区域的范围、边界和意图。

组（group）——负责一组作业或活动的部门、经理和个人的集会。组的规模可以变化；从一个受指派的以部分时间工作的单个个人，到几个从不同部门安排来的非全日制的个人，到几个全部时间专注此工作的个人。

宿生计算机（hostcomputer）——用于开发软件的计算机。（参照目标计算机。）

初始等级（initiallevel）——（见成熟度等级。）

制度化（institutionalization）——建立那些支持方法、实践和规程的基础设施和公司文化，使得这些方法、实践和规程成为经营活动的进行方式，甚至在最初定义它们的人员离去之后。

集成软件管理(integratedsoftwaremanagement)-----将软件工程活动和管理活动统一并集成为一个协调的、基于组织标准软件过程和相关过程财富的、已定义的软件过程。

集成（integration）——（见软件集成。）

关键实践（keypractices）——对一个关键过程区域的有效实施和制度化贡献最大的基础设施和活动。

关键过程区域（keyprocessarea）-----一组相关的活动，当这些活动共同完成时，能实现据信双方建立过程能力是至关重要的一组目标。每个关键过程区域已经定义在单个成熟度等级上。SEI 确定它们是～些主要构成单元用来帮助确定一个组织的软件过程能力和了解为向更

高成熟度等级前进所必须作的改进。CMM 中等级 2 的关键过程区域是需由管理、软件项目策划、软件项目跟踪和监督、软件子合同管理、软件质量保证、和软件配置管理。CMM 中等级 3 的关键过程区域是组织过程焦点、组织过程定义、培训大纲、集成软件管理、软件产品工程、组间协调和同行评审。等级 4 的关键过程区域是定量过程管理和软件质量管理。等级 5 的关键过程区域是缺陷预防、技术改革哲理部过程行政管理。

生存周期（lifecycle）——（见软件生存周期。）

维护(maintenance)——在交付以后对一个软件系统或成分进行修改的过程，目的在于

改正差错、改善性能或其它属性、或者对其调整以适应改变后的环境。[IEEE - STD-610]

进行管理和控制（managedandcontrolled）——标识和定义软件工作产品的过程，这些软件工作产品不是基线的～部分，因此并不置于配置管理之下，但是为使项目以有纪律的方式开展工作，它们必须受控。“进行管理和控制”意味着在给定时间（过去或现在）使用的工作产品的版本是已知的（即版本控制），而且以受控的方式引进更改(即更改控制)。

已普理级（managedlevel）——（见成熟度等级。）

经理（manager）——一个角色，他对在经理的职责范围内进行作业或活动的个人提供技术上的和行政上的管理和控制。经理的传统职能包括对其职责范围内的工作进行策划、提供资源、进行组织，加以指导和控制。

成熟度等级（maturitylevel）——一个妥善定义的、朝着实现成熟软件过程进化的平台。

SEI 的能力成熟度模型中的五个成熟度等级是：

动抬级——软件过程被特征化为无序的、偶尔甚至是混乱的。几乎没有什么过程是

已定义的，成功依赖于个人的努力。

- ✍✍ 可重复级——已建立起基本的项目管理过程去跟踪成本、进度和功能性。为了在具有类似应用问题的项目上能重复以前的成功，必要的过程纪律已到位。
- ✍✍ 已定义级——在管理活动和工程活动两方面软件过程均已文档化、标准化，并且集成到组织的标准软件过程。所有的项目均采用经批准的、组织标准软件过程的剪裁版本去开发和维护软件。
- ✍✍ 已管理级——已采集软件过程和产品质量的详程度量。软件过程和产品均得到定量的理解和控制。
- ✍✍ 优化级——通过利用来自过程和对创新思想及技术的先导性试验的定量反馈信息，使得连续过程改进或为可能。

成熟度提问单 (maturityquestionnaire)——一组关于软件过程的提问，它抽样检查 CMM 的每个关键过程区域中的关键实践。它被用作评价一个组织或项目可靠地执行软件过程的能力的出发点。

度量单位 (measure)——度量的一个单位 (例如源代码行或者设计的文件页数)。

度量 (measurent)——某物的线数、容量、数量或总量 (例如。300 源代码行或 7 个设计文件页)。

方法 (method)——规则和判据的 ~ 个适度完整的集合，这些规则和判据建立起一种精确的和可重复的进行作业并达到所希望结果的方式。

方法论 (methodology)——一组方法、规程和标准的集合，它确定一套集成的开发某产品的综合工程方法。

里程碑 (milestone)——一个在预定时间发生的事件，某个人应对其负责，并且能用它来测量进程。

非技术性要求 (notechanicalrequirements)——影响和限定软件项目管理活动的协议、条件和 (或) 合同条款。

运行软件 (operationalsoftware)——打算交付给顾客，配置在预定环境中在某个系统中运行的软件。

优化级 (optimizininglevel)—— (见成熟度等级。)

组织 (organization)——一个公司或其它实体 (例如政府机关或军种) 中的一个单位，在其中对许多项目作为一个整体加以管理。一个组织内的所有项目共有一个共同的顶层经理和共同的方针。

组织的测量大纲 (organizatlon ' smasurementprogram)——一组相关的用以阐述组织的测量需求的要素。它包括对全组织范围适用的测量的定义、收集组织测量数据的方法和实践、分析组织测量数据的方法和实践、以及组织的测量目标。

组织的软件过程财富 (organization ' ssoftwareprecessassets)——一组实体的集合，它由组织维护，供项目在制定、剪裁、维护和实施其软件过程时使用。这些软件过程财富一般包括：

- ✍✍ 组织标准软件过程，
- ✍✍ 对批准使用的软件生存周期的描述，
- ✍✍ 剪裁组织标准软件过程的指南和准则，
- ✍✍ 组织的软件过程数据库，和
- ✍✍ 软件过程一有关文档库。

任何一项组织认为在进行过程定义和维护活动方面有用的实体均可作为过程财富被包括在内。

组织的软件过程数据库 (organization ' ssoftwaredatabase)——为收集有关软件过程及

其生成的软件工作产品的数据并使其可供使用而建立的数据库,特别当数据与组织标准软件过程有关时。该数据库包含或者索引实际的测量数据换和为理解测量数据及评估其合理性与适用性所需要的相关信息。过程和工作产品的数据的例子包括:对软件规模、工作量和成本的估计值;软件规模、工作量和成本和实际数据;生产率数据;同行评审的范围和效率;以及在软件代码中所发现的缺陷的数目及严重性。

组织标准软件过程 (organization's standard software process) ——对于基本过程的可操作的定义,基本过程指导在一个组织内的所有项目上建立一个共同的软件过程。它描述预期每个软件项目的已定义软件过程均会包含的基本的软件过程元素。它还描述这些软件过程元素间的关系(例如:排序和界面)。

定向培训 (orientation) ——向对在某专题领域内负责开展工作的个人进行监督或与这些个人打交通的那些人进行的有关该专题的概述或介绍。(对照参看培训。)

Pareto 分析 (Pareto analysis) ——通过将缺陷原因按从最重要到最不重要加以排序的方法对缺陷进行分析。Pareto 分析基于以 19 世纪经济学家 Vilfredo Pareto 命名的原理,即绝大多数后果起源于相对少的原因,也就是 80% 后果是由可能原因中的 20% 所遗成的。

同行评审 (peer review) ——由一个软件工作产品生产者的同行遵循已定义的规程对产品作的评审,目的在于识别出缺陷和改进之处。

同行评审领导 (Peer review leader) ——一个经过专门培训并取得规划、组织和领导同行评审资格的个人。

定期评审成活动 (periodic review / activity) ——在规定的固定时间间隔处出现的评审或活动。(对照参看事件一驱动评审成活动)

方针 (policy) ——一种指导原则,一般由高级管理者制定,组织或项目采用它去影响和决定决策。

主承包商 (prime contractor) ——一个在行政上管理有关一个或多个产品的设计、开发、和(或)维护工作的子合同的个人、合作伙伴、公司或协会。

规程 (procedure) ——对为完成一项给定作业将采取的行动步骤的书面描述。[IEEE - STD - 610]

过程 (Process) ——为了某个给定目的所执行的 ~ 系列步骤;例如,软件开发过程。[IEEE - STD-610]

过程能力 (process capability) ——通过遵循一个过程能实现预期结果的程度。(对照参过程体验。)

过程能力基线 (process capability baseline) ——用文档记载的对在典型环境下由于遵循某特定过程通常所能实现预期结果的范围的特性描述。一般在组织层上建立过程能力基线。(参过程性能基线以作对照。)

过程数据库 (process database) ——(参见组织的软件过程数据库。)

过程描述 (process description) ——对过程主要成分的可操作的定义。即以一种全面的、精确的、可验证的方式规定 ~ 个过程的要求、设计、行为或其它特征的文档、它也可能包括一些用以确定是否已满足上述特征的规程。在作业层、项目层或组织层上均可能有过程描述。过程制定 (process development) ——定义和描述一个过程的行为。它可能包括策划、构造、设计、实施和确认。

过程测量 (process measurement) ——一组定义、方法和活动,用来对一个过程及其产生的产品进行测量以便对该过程进行特征描述和理解它。

过程性能 (process performance) ——对遵循某过程所实现的实际结果的度量。(对照参过程能力。)

过程性能基线 (process performance baseline) ——用文档记载的对遵循一个过程所实现

的实际结果的特征描述，用来作为将实际过程性能与期望的过程性能进行比较的基准。一般在项目层上建立过程性能基线，虽然最初的过模控能基线一般从过程能力基线上导出。（参见过程能力基线以作对照。）

过程现我（processtailoring）——籍助于对～个过程的过程元素的细节或其它不完全的规格说明进行提炼、调整或补充的方法来生成一个过程描述的活动。一般在过程剪裁期间阐述项目的特殊业务需求。

产品（Product）——修见软件产品和软件工作产品。）

剖面（Profile）——计划或预测结果与实际结果的比较，一般是以图形的形式表示一段时间内随时间而变的比较结果。

项目（project）——一项要求协同一致努力的任务，它关注开发和（或）维护一个具体的产品。产品可以包括硬件、软件或其它成分。一般，一个项目有其自己的投资、成本核算和交付时间表。

项目定义软件过程（project'sdefinedsoftwareprocess）——对项目所用软件过程的可操作的定义。项目定义软件过程是一个已很好特征化的和已理解的软件过程，用软件标准、规程、工具和方法予以描述。通过剪裁组织的标准软件过程以适合项目的具体特征的方法来制定它。

（也可参见组织的标准软件过程、有效过程、和妥善定义的过程。）

项目经理（projectmanager）——对整个项目负有全面经营职责的角色，即指导、控制、行政管理和调整一个构造软件或硬件/软件系统的项目的个人。项目经理是最终对顾客负责的个人。

项目软件经理（Projectsoftwaremanager）——对一个项目的全部软件活动负有全面职责的角色。项目软件经理是项目经理按照软件约定与之打交道的个人，也是控制一个项目的全部软件资源的个人。

质量（quality）——（1）一个系统、成分或过程满足规定需求的程度。（2）一个系统、成分或过程满足顾客或用户的需要或期望的程度。[IEEE - STD-610]

质量保证（qualityassurance）——（参见软件质量保证。）

定全控制（quantitativecontrol）——任何一种定量的或基于统计方法的适于以下工作的技术：分析软件过程、识别软件过程性能变化的特殊原因和使得软件过程性能处于妥善定义的界限之内。

可重复级（repeatablelevel）——（参见成熟度等级。）

要求的培训（requiretraining）——组织指定的为执行一个特定角色所要求的培训。

风险（risk）——遭受损失（loss）的可能性。

风险管理（riskmanagement）——一种问题分析的手段，它采用风险概率去估计某情况下的风险以达到对所涉及风险更为精确的了解。风险管理包括风险识别、分析、优先级排序和控制。风险管理计划（riskmanagementplan）——描述一个项目应进行的风险管理活动的若干

计划的总集合。

角色（role）——已定义职责的一个单元，可以由一个或多个个人承担。

SCE 一软件能力评价的首字母缩略词。

SCM 一软件配置管理的首字母缩略词。

高级经理（seniormanager）——在组织的足够高层次上的一个管理角色，他主要关注组织的长期生命力，而不是短期的项目和合同所关心的事和压力。一般讲，一个工程方面的高级经理负责多个项目。

软件作系结构（softwarearchitecture）——软件或模块的组织结构。[IEEE - STD-610]

软件基线审计 (softwarebaselineaudit)——对于软件基线库的结构、内容和设施的考查,以便查证基线是否符合描述基线的文档。

软件基线库 (softwarebaselinelibrary)——存储配置项及相连记录的仓库的内容。

软件构造 (softwarebuild)——软件系统或成分的一个可运行的版本,它包含最终软件系统或成分将提供的能力的规定于集。

软件能力评价 (softwarecapabilityevaluation)——由一个经培训的专业人员群组所作的鉴定,旨在识别合格的从事某项软件工作的承包商或监控一项现有软件工作所用软件过程的状态。

软件配置控制委员会 (softwareconfigurationcontrolboard)——负责评价和批准(或不批准)对配置项所提出的更改,并负责保证那些已批准的更改能得到实施的组。

软件开发计划 (softwaredevelopmentplan)——描述软件项目应进行活动的那些计划的集合。它指导对一个软件项目的软件工程组所进行活动的管理。它并不受限于任何一个特定的用于制定计划的标准(例如 DOD - STD - 2167A 和 IEEE - STD - 1058)的范围,但可能采用类似的术语。

软件工程组 (softwareengineeringgroup)——负责一个项目的软件开发和维护活动(即需求分析、设计、编码和测试)的个人(经理和技术人员)的集合。进行软件一有关工作的组,例如软件质量保证组,软件配置管理组和软件工程过程组,则不如软件工程组之内。

软件工程过程组 (Softwareengineeringprocessgroup)——促进组织所用软件过程的定义、维护和改进的专家组。在关键实践中,这个组一般指的是“负责组织的软件过程活动的组”。

软件工程职员 (softwareengineeringstaff)——进行项目的软件开发和维护活动的技术人员(例如:分析员、程序员和工程师),包括软件作业领导,但不是经理。

软件集成 (softwareintegration)——将选定的软件成分组装在一起的过程,旨在提供最终软件系统将具备的能力或者指定的部分能力。

软件生存周期 (softwarelifecycle)——从设想一软件产品开始到软件不再供使用时为止的时间间隔。软件生存周期一般包括概念阶段、需求阶段、设计阶段、实现阶段、测试阶段、安装和调整阶段、运行和维护阶段、有时还有退役阶段。[IEEE . sTD - 610]

软件经理 (softwaremanager)——在项目或组织层上任何一个对软件开发和(或)维护负有直接责任的经理。

软件计划 (softwareplans)——用于表达将如何开展软件开发和(或)维护活动的那些计划的集合,既可是正式计划又可是非正式计划。这样的计划的例子有:软件开发计划、软件质量保证计划、软件配置管理计划、软件测试计划、风险管理计划和过程改进计划。

软件过程 (softwareprocess)——人们用以开发和维护软件及其相连产品(例如:项目计划、设计文档、代码、测试用例和用户手册)的一组活动、方法、实践和变换(transformation)。

软件过程评估 (softwareprocessassessment)——一个经培训的软件专业人员的群组所作的一项鉴定、旨在确定一个组织当前软件过程的状态以及组织所面临的具有高优先级的软件过程'有关的问题,和获得组织对软件过程改进的支持。

软件过程财富 (softwareprocessassets)——(见组织的软件过程财富。)

软件过程能力 (softwareprocesscapability)——(见过程能力。)

软件过程描述 (softwareprocessdescription)——对于确定为项目定义软件过程或组织标准软件过程中的主要软件过程成分的可操作的定义。它以一种全面的、精确的、可验证的方式用文档记载一个软件过程的要求、设计、行为或其它特征。(也可参见过程描述。)

软件过程元素 (softwareprbeesselement)——软件过程描述的一个构成元素。每个过程元素包括一个妥善定义的、有界的、紧密相关的作业集合(例如软件估计元素、软件设计元素、编码元素和同行评审元素)。过程元素的描述可以是待填入的样板、持党成的片段、待

精炼的抽象概念、或者待修订的或已使用而无需修订的完整描述。

软件过程改进计划 (software process improvement plan) —— 一个从软件过程评估的建议中导出的计划，它标识出将采取的旨在改进软件过程的具体行动，并且略述用以实施这些行动的计划。有时称其为行动计划。

软件过程改进建设 (software process improvement proposal) —— 一个已文档化的、对一个过程或过程有关条款所作的将提高软件过程能力和性能的更改建议。(也可参见行动建议。)

软件过程成熟度 (software process maturity) —— 一个具体过程被明确地定义、管理、测量、控制而且有效的程度。成熟度暗示着能力上的增长潜力，并指示一个组织的软件过程的丰富性和组织中所有项目在运用该过程上的一致性。

软件过程性能 (software process performance) —— (参见过程性能)

软件过程一有关文档 (software process - related documentation) —— 预计未来的项目在剪裁组织标准软件过程时可能用到的一些实例文档 (example documents) 和文档片段。实例可以涉及以下论题：例如项目定义软件过程、标准、规程、软件开发计划、测量计划和过程培训材料。

软件产品 (software product) —— 指定要交付给顾客或最终用户的一个完整的计算机程序、规程及相连文档和数据的集合，或者该集合中的任何一个单独的项。[IEEE - STD-610] (对照参见软件工作产品。)

软件项目 (software project) —— 一项要求协同一致努力的任务，它关注一个系统的软件成份及其相连文档的分析、详细说明、设计、开发、测试和(或)维护等工作。一个软件项目可以是构造一个硬件/软件系统的项目的~部分。

软件质量保证 (software quality assurance) —— (1) 为了对一个软件工作产品符合所制定的技术要求提供足够的量信度，必须采取的全部行动的一个有计划的和系统的模式、(2) 为了评价开发和(或)维护软件工作产品的过程而设计的一组活动。

软件质量目标 (software quality goal) —— 为一个软件工作产品确定的定量质量目标。

软件质量管理 (software quality management) —— 以下工作的过程：确定一个软件产品的质量目标；制订实现这些目标的计划；以及为了满足顾客和最终用户的需要和希望而监控和调整这些软件计划、软件工作产品、活动与质量目标。

软件一有关组 (software - related group) —— 代表一门支持但不直接负责软件开发和(或)维护的软件工程科目的个人(包括经理和技术人员)的集合。软件工程科目的例子包括软件质量保证和软件配置管理。

软件需求 (software requirement) —— 用户为解决一个问题或实现一个目标要求软件一定得满足的条件或能力。[IEEE-STD-610]

软件工作产品 (software work product) —— 作为定义、维护或使用软件过程一部分所生成的任何人工制品，它包括过程描述、计划、规程、计算机程序、和相连的文档，这些可能打算也可能不打算交付给顾客或最终用户。(参见软件产品以作对照。)

SPA — 软件过程评估的首字母缩略词。

特殊原因 (一个缺陷的) (special cause (of a defect)) —— 那种在某种暂态情况下所特有的而不是一个过程的固有部分的缺陷原因。特殊原因在过程性能上造成随机的变化(噪音)。(参见一般原因以作对照。)

SQA — 软件质量保证的首字母缩略词。

职员 (staff) —— 负责完成一项指派的职能——例如软件开发或软件配置管理——的个人，包括作业领导但不是经理。

阶段 (stage) —— 软件工作的一种划分，它具有可管理的规模，并且表示项目所进行

的相关作业的一个有意义的和可测的集合。通常认为阶段是对软件生存周期的细分，常常以在下一阶段开始前的一次正式评审作为结束。

标准 (standard) —— 为了规定一种有纪律的、一致的软件开发方法而采用的和强制实施的强制性要求。

标准软件过程 (standardsoftwareprocess) —— (参见组织的标准软件过程。)

工作陈述 (statementofwork) —— 对为一个项目所要求的全部工作的描述，一般由顾客提出。

子合同经理 (subcontractmanager) —— 主承包商的组织中直接负责控制和管理一个或多个子承包的经理。

子承包商 (subcontractor) —— 与一个组织 (即主承包商) 签订合同进行一个或多个产品的设计、开发和 (或) 制造工作的个人、合作伙伴、公司或协会。

系统 (system) —— 为完成一个特定功能或一组功能而组织在一起的一些成分的集会。
[IEEE - STD-610]

系统工程组 (systemengineeringgroup) —— 负责以下工作的个人的集会 (包括经理和技术人员): 规定系统需求; 将系统需求分配给硬件、软件和其它成分; 规定硬件、软件和其它成分间的界面; 以及监控这些成分的设计和开发工作以保证符合其规格说明。

系统需求 (systemrequirement) —— 一个系统或系统成分为实现用户在解决一个问题时所需要的条件或能力而必须满足或具有的条件或能力。[IEEE - STD - 610]

分配给软件的系统需求 (systemrequirementsallocatedtosoftware) —— 拟由系统的软件成分实现的系统需求的子集。分配需求是软件开发计划主要输入。软件需求分析精心推敲和提炼分配需求，产生用文档记载的软件需求。

剪裁 (tailor) —— 为更好地匹配过程要求或产品需求而修改过程、标准或规程。

目标计算机 (targetcomputer) —— 打算在它上面运行交付软件的计算机。(对照参见宿生计算机。)

作业 (task) —— (1) 作为工作基本单元加以处理的一序列指令。[IEEE - STD-610]
(2) 软件过程中的一个妥善定义的工作单元，它能提供管理者一个可视的对项目状态的检验点。作业有准备就绪准则 (前提条件) 和完成准则 (后绪条件)。(对照参见活动。)

作业启动会议 (taskkick-offmeeting) —— 在项目的一个作业开始时举行的会议，目的在于使有关的个人作好准备能有效地进行该作业的活动。

作业领导 (taskleader) —— 从事一特定作业的技术群组的领导，他负技术责任并对工作在该作业的职员进行技术指导。

群组 (team) —— 一组人，常来自不同的但相关的组，受指派去执行组织或项目的一个妥善定义的功能。群组成员可以是群组的非全日制的参加者并有其它主要职责。

可测试性 (testability) —— (1) 一个系统或者成分在建立测试准则和进行测试以便确定测试准则是否已得到满足方面提供方便的程度。(2) 从允许建立测试准则和进行测试以便确定那些准则是否已得到满足角度看，阐述需求的程度。[IEEE-STD-610]

技术要求 (technicalrequirements) —— 描述软件必须作的事及其运行限制的那些要求。技术要求的例子包括功能、性能、界面和质量要求。

技术 (technology) —— 在实现某项特定的结果方面科学和 (或) 工程学的运用。

可追踪性 (traceability) —— 开发过程中的两个或多个产品，特别是彼此间具有继承关系或主从关系的那些产品间能建立相互关系的程度。[IEEE - STD-610]

培训 (train) —— 造就掌握专门化的指南和实践的能手。(也可参见定向培训。)

培训组 (traininggroup) —— 负责为某个组织协调和安排培训活动的个人的集合 (包括经理和职员)。这个组一般准备和讲授大多数的培训课程并协调其它培训载体的使用。

培训大纲 (trainingprogram) ——集中阐述组织培训需求的那些相关元素的集合。它包括组织的培训计划、培训教材、培训的开发、培训的执行、培训设施、对培训的评价、以及培训记录的维护。

免除培训证明书 (traninwaiver) ——一个书面批文批准某个人免于接受业已指定为某特定角色所要求的培训。因为已经客观地断定这个人已具有担当该角色所需要的技能,因此,免除得到认可。

单元 (unit) ——(1) 在计算机软件成分的设计中具体说明的一个独立的可测试的元素。(2) 计算机程序的一个逻辑上独立的部分。(3) 不再能细分为其它成分的软件成分。[IEEE - STD-610]

用户 (user) ——(见最终用户。)

确认 (validation) ——在开发过程期间或结束时为了确定软件是否满足规定的需求而对软件进行评价的过程。[IEEE - STD-610]

验证 (verificatio) ——为了确定某个给定开发阶段的产品是否满足该阶段开始时所施加的条件而对软件进行评价的过程。[IEEE - STD - 610]

验证实施 (verifyingimplementation) ——(见共同特点。)

免修证明书 (waiver) ——(见免除培训证明书。)

妥善定义的过程 (well-definedprocess) ——包括以下各项的过程;准备就绪准则、输入、进行工作的标准和规程、验证机制(例如同行评审)、输出和完成准则。(也可参见有较过程。)

附录 C 关键实践的简略版本

本附录给出关键实践的简略版本,提供 SEI 对每个关键过程区域所描述的主要活动的高层次概况。利用它能“快速查看”每个关键过程区域。但是它既不提供这些关键实践的具体活动也不包括全部的关键实践。制作它的目的仅仅为了提供信息,而不是为了确定对关键实践的符合性或策划过程改进工作。

本简略本包含对关键过程区域、它的目标和关键过程区域的执行的活动这共同特点中的关键实践陈述的简短描述。这些条款是从详细的关键实践文本中一字不变抽出的。

本附录中不包括在其它共同特点(即执行约定、执行能力、测量和分析、及验证实施)下所规定的其它关键实践。但这些其它关键实践必须到位以保证这里列出的关键实践得到恰当地和有效地实施,保证它们扎扎实实地建立起来,将得到维护和不随时间的变迁而性能变坏,并且保证能将它们有效地运用到新的工作。为了恰当地建立一个关键过程区域,应该使用全部关键实践。

执行约定一般涉及制定组织方针和规定高级管理者的支持保证。执行能力一般与资源、组织机构和培训等有关。测量和分析一般包括可能采取的、旨在确定执行的活动的状态和有效性的测量的例子。验证实施一般包含管理者和软件质量保证组所作的评审和审计。

等级 2: 需求管理

需求管理的目的是在顾客和将处理顾客需求的软件项目之间建立对顾客需求的共同理解。

需求管理包括和顾客一起建立和维护有关软件项目需求的协议。该协议称作“分配给软件的系统需求”。“顾客”可解释为系统工程组、销售组、另一个内部组织、或者一个外部顾客。协议既包括技术需求、又包括非技术需求(例如交付日期)。该协议形成估计、策划和跟踪整个软件生存周期内软件项目活动的基础。

将系统需求分配给软件、硬件、和其它系统成分的工作可能由软件工程组之外的组(例如系统工程组)完成,软件工程组可能对此分配无直接控制。在项目约束范围内,软件工程

组采取恰当步骤以保证对分配给软件的需求建立文档、并加以控制，该组负责阐述分配给软件的系统需求。

为实现此控制，软件工程组评审初始的和经修改的分配给软件的系统需求，以便在它们被纳入软件项目之前使问题得以解决。每当改变分配给软件的系统需求时，都要调整受到影响的软件计划、工作产品和活动，使其与更新后的需求保持一致。

需求管理的目标是：

- 1.分配给软件的系统需求是受控的，建立供软件工程和管理使用的基线。
2. 软件计划、产品和活动与分配给软件的系统需求保持一致。

需求管理所执行的顶层活动是：

1. 在分配需求被纳入软件项目之前，软件工程组评审它们。
2. 软件工程组采用分配需求作为软件计划、工作产品和活动的基础。
3. 评审对分配需求的更改，将其纳入软件项目。

等级2：软件项目策划

软件项目策划的目的是为完成软件工程和管理软件项目制定合理的计划。

软件项目策划包含估计待完成的工作，建立必要的约定和确定进行该工作的计划。

软件策划首先作出有关待完成的工作和其它定义及界定软件项目的约束和目标(由需求管理关键过程区域的实践所建立的)的陈述。软件策划过程包括以下步骤：估计软件工作产品的规模及所需的资源，制定时间表，鉴别和评估软件风险和协商约定。为了执行软件计划(即软件开发计划)，可能需要重反侧面过这些步骤。

该计划提供完成和管理软件项目活动的基础，并按照软件项目的资源、约束和能力，阐述对软件项目的顾客作的约定。

软件项目策划的目标是：

1. 对供策划和跟踪软件项目用的软件估计建立文档。
2. 软件项目的活动和约定是有计划的并建立文档。
3. 受影响的组和个人同意他们关于软件项目的约定。

软件项目策划所执行的顶层活动是：

二. 软件工程组参加项目建议群组。

2. 在整个项目策划的早期阶段启动软件项目策划，此两项策划平行进行。
3. 在项目的整个生存期内，软件工程组和其它受影响的组一起参加整个项目的策划。
4. 高级管理者参加按照已文档化的规模评审对组织外部的个人和组所作的软件项目约定。
5. 识别或确定具有可管理规模的预先规定阶段的软件生存周期。
6. 按照已文档化的规程制定项目的软件开发计划。
7. 对有关软件项目的计划建立文档。
8. 识别为建立和保持对软件项目的控制所必须的工作产品。
9. 按照已文档化的规程导出对软件工作产品规模(或对软件工作产品规模的更改)的估计。
10. 按照已文档化的规程导出对软件项目的工作量及成本的估计。
11. 按照已文档化的规程导出对项目的关键计算机资源的估计。
12. 按照已文档化的规程导出项目的软件进度表。
13. 对与项目的成本、资源、进度和技术方面相联系的软件风险进行鉴别、评估和建立文档。
14. 制定关于项目软件工程设施和支持工具的计划。
15. 记录软件策划数据。

等级 2：软件项目跟踪和监督

软件项目跟踪和监督的目的是建立对实际进展的适当的可视性,使管理者能在软件项目性能明显偏离软件计划时采取有效措施。

软件项目跟踪和监督包括对照已文档化的估计、约定、和计划评审和跟踪软件完成情况和结果,基于实际的完成情况和结果调整这些计划。

软件项目的已文档化的计划即软件开发计划,正如在软件项目策划关键过程区域中所描述的)用作跟踪软件活动、传送状态、和修订计划的基础。管理者监控软件活动。主要通过在所选出的软件工作产品完成时和在所选择的里程碑处,将实际的软件规模、工作量、成本和时间表与计划相比较,来确定进展情况。当确定未实现软件项目计划时,采取纠正措施。

这些措施可以包括修订软件开发计划以反映实际的完成情况和重新策划遗留的工作或者采取改进性能的措施。

软件项目跟踪和监督的目标是:

- 1、对照软件计划,跟踪实际结果和性能。
- 2、当实际结果和性能明显偏离软件计划时,采取纠正措施并加以管理直到结束。
- 3、对软件的约定的三改得到受到影响的组 and 个人的认可。

软件项目跟踪和监督所执行的顶层活动是;

1. 将已文档化的软件开发计划用于跟踪软件活动和传送状态。
2. 按照已文档化的规程修订项目的软件开发计划。
3. 高级管理者参与按照已文档化的规程评审那些对组织外部的个人和组所作的软件项目约定和约定的更改。
4. 将经批准的、影响软件项目约定的更改传达给软件工程组和其它软件一有关组的成员。
5. 跟踪软件工作产品的规模(或者软件工作产品更改的规模),必要时采取纠正措施。
6. 跟踪项目的软件工作量和成本,必要时采取纠正措施。
7. 跟踪项目的关键计算机资源,必要时采取纠正措施。
8. 跟踪项目的软件进度,必要时采取纠正措施。
9. 跟踪软件工程技术活动,必要时采取纠正措施。
10. 跟踪与项目的成本、资源、进度及技术方面有关的软件风险。
11. 记录软件项目的实际测量数据和重新策划的数据。
12. 软件工程组进行定期的内部评审以便对照软件开发计划跟踪技术进度、计划、性能和问题。
13. 按照已文档化的规程在所选择的项目里程碑处进行正式评审以评价软件项目的完成情况和结果。

等级 2：软件子合同管理

软件子合同管理的目的是选择合格的软件子承包商并有效地管理他们。

软件子合同管理包括选择软件子承包商、建立和子承包商的约定,及跟踪和评审子承包商的性能和结果。这些实践包括对纯软件子合同的管理,也包括对于合同的软件成分的管理,后者含有软件、硬件和可能有的其它系统成分。

基于子承包商完成工作的能力选择子承包商。许多因素对将承包商工作的一部分签为子合同的决策产生影响。选择子承包商可以基于战略经营同盟及技术上的考虑。这个关键过程区域的实践阐述与将工作的一个确定部分签子合同给另一个组织相联系的传统采购过程。

当签子合同时,建立一个包括技术和非技术(例如交付日期)需求的巴文档化的协议并将其用作管理于合同的基础。对将由于承包商完成的工作和关于该工作的计划建立文档。子承包商将遵循的标准和主承包商的标准一致。

子承包商的软件的策划、跟踪和监督活动由于承包商完成。主承包商确保恰当地完成这些策划、跟踪、和监督活动并且子承包商交付的软件产品满足其验收准则。主承包商和子承包商一起工作去管理他们的产品和过程界面。

软件予合同管理的目标是：

- 1．主承包商选择合格的软件于承包商。
- 2．主承包商和软件子承包商认同他们相互的约定。
- 3．主承包商和软件子承包商保持不断的通信。
- 4．主承包商对照约定跟踪软件子承包商的实际结果和性能。

软件予合同管理所执行的顶层活动是：

- 1．按照已文档化的规程定义和规划待签子合同的工作。
- 2．按照已文档化的规模，在评价子合同投标者完成该工作的能力的基础上选择软件予承包商。
- 3．将主承包商和软件于承包商间的合同协议用作管理干合同的基础。
- 4．主承包商评审和批准已文档化的子承包商软件开发计划。
- 5．将已文档化的且经批准的子承包商软件开发计划用于跟踪软件活动和通信状态。
- 6．按照已文档化的规程判定对软件子承包商的工作陈述、子合同条款和条件、以及其它约定的更改。
- 7．主承包商的管理者和软件予承包商的管理者一起执行定期的状态或协调评审。
- 8．软件子承包商参与定期技术评审和交流。
- 9．按照已文档化的规程在所选择的里程碑处进行正式评审，评价子承包商的软件工程完成情况和结果。
- 10．主承包商的软件质量保证组按照已文档化的规程监控于承包商的软件质量保证活动。
- 11．主承包商的软件配置管理组按照已文档化的规程监控于承包商的软件配置管理活动。
- 12．主承包商按照已文档化的规程进行验收测试，这是子承包商软件产品交付的一部分。
- 13．定期评价软件子承包商的性能，并与子承包商一起评审该评价工作。

等级 2:软件质量保证

软件质量保证（SQA）的目的是向管理者提供适当的对软件项目正使用的过程和正构造产品的可视性。

软件质量保证包括评审和审计软件产品和活动以验证它们符合适用的规程和标准，给项目和其它有关的经理提供这些评审和审计的结果。

在软件项目的早期阶段，软件质量保证组与软件项目一起工作制定计划、标准和规程等，这些计划、标准和规程将增加软件项目的价值并将满足项目和组织方针上的限制。通过参与制定计划、标准和规程，软件质量保证组帮助确保它们适合项目的需要，并且帮助验证它们对完成整个软件生存周期中的评审和审计将是适用的。软件质量保证组在整个生存周期评审项目活动，审计软件工作产品，并就软件项目是否正遵守已制定的计划、标准、和规程等给管理者提供可视性。

首先在软件项目内部处理符合性问题，如可能的话就地解决它。对于那些无法在软件项目内部解决的问题，软件质量保证组逐级上递该问题到管理者的恰当层次以求得解决。

这个关键过程区域只包括该组履行软件质量保证功能的实践。而识别软件质量保证组要评审和（或）审计的具体的活动和工作产品的实践一般包含在其它关键过程区域的验证实施共同特点中。

软件质量保证的目标是；

1. 软件质量保证活动是有计划的。
2. 软件产品和活动遵守适用的标准、规程和需求的情况得到客观的查证。
3. 受影响的组和个人接到软件质量保证活动和结果的通知。
4. 高级管理者处理在软件项目内部不能解决的不符合问题。

软件质量保证所执行的顶层活动是：

1. 按照已建档的规程为软件项目制定 SQA 计划。
2. 按照 SQA 计划进行 SQA 组的活动。
3. SQA 组参与准备和评审项目的软件开发计划、标准、和规程。
4. SQA 组评审软件工程活动以验证符合性。
5. SQA 组审计指定的软件工作产品以验证符合性。
6. SQA 组定期向软件工程组报告其活动的结果。
7. 按照已文档化的规程对在软件活动和软件工作产品中所鉴别出的偏差建立文档并加以处理。
8. 当合适时，SQA 组与顾客的 SQA 人员一起对它的活动和发现进行定期评审。

等级 2：软件配置管理

软件配置管理 SCM 的目的是建立和维护在项目的整个软件生存周期中软件项目产品的完整性。

软件配置管理包括标识在给定时间点上软件的配置(即选定的软件工作产品及其描述)，系统地控制对配置的更改，并维护在整个软件生存周期中配置的完整性和可跟踪性、置于软件配置管理之下的工作产品包括交付给顾客的软件产品(例如软件需求文档和代码)，以及在这些软件产品等同的产品项或生成这些软件产品所要求的产品项(例如编译程序)。

建立一个软件基线库，当软件基线形成时就将它们纳入该库。通过软件配置管理的更改控制和配置审核功能，系统地控制基线的更改和那些利用软件基线库所构造成的软件产品的发行。

这个关键过程区域仅包括实施软件配置管理功能的实践。而标识具体的配置项或单元的实践则包含在描述每个配置项或单元的开发和维护的关键过程区域中。

软件配置管理的目标是：

1. 软件配置管理活动是有计划的。
2. 所选定的软件工作产品是经标识的、受控的和适用的。
3. 对已标识的软件工作产品的更改是受控的。
4. 受影响的组和个人得到软件基线的状态和内容的通知。

软件配置管理所执行的顶层活动是：

1. 按照已文档化的规程对每个项目准备一份 SCM 计划。
2. 用已文档化的经批准的 SCM 计划作为进行 SCM 活动的基础。
3. 建立一个配置管理库系统作为软件基线的仓库。
4. 标识将置于配置管理之下的软件工作产品。
5. 按照已文档化的规程，启动、记录、评审、批准和跟踪对所有配置项或单元的更改请求和问题报告。
6. 按照已文档化的规程控制对基线的更改。
7. 按照已文档化的规模生成由软件基线库制造的产品并控制它们的发行。
8. 按照已文档化的规程记录配置项或单元的状态。
9. 编制用文档记载 SCM 活动和软件基线内容的标准报告，并使受影响的组和个人可以使用它。
10. 按照已文档化的规程进行软件基线审计。

等级3：组织过程焦点

组织过程焦点的目的是建立组织对软件过程活动的责任,这些活动能改进组织的整体软件过程能力。

组织过程焦点包括增进和保持对组织的和项目的软件过程的了解,协调那些评估、制定、维护和改进这些过程的活动。

组织提供长期的保证和资源,通过诸如软件工程过程组那样的组来协调那些横跨当前和未来软件项目的软件过程的制定和维护工作。这个组负责组织的软件过程活动,具体负责制定和维护组织标准软件过程和有关的过程财富(正如在组织过程定义关键过程区域中所描述的),同时它和软件项目一起协调过程活动。

组织过程焦点的目标是：

1. 在整个组织中,过程的制定和改进活动是协调的。
2. 标识出所使用的软件过程与一个过程标准相比的强处和弱处。
3. 组织层上过程的制定和改进活动是有计划的。

组织过程焦点所执行的顶层活动是：

1. 定期地评估软件过程,制定处理评估发现的行动计划。
2. 组织制订和维护用于其软件过程制定和改进活动的计划。
3. 在组织层上协调组织和项目制定及改进其软件过程的活动。
4. 在组织层上协调组织的软件过程数据库的使用。
5. 对组织中限制使用的新的过程、方法和工具进行监控和评价,并当合适时,将其推广到组织的其它部分。
6. 在整个组织内协调有关组织和项目的软件过程的培训。
7. 将组织和项目的有关软件过程制定和改进的活动通知与实施软件过程有关的组。

等级3：组织过程定义

组织过程定义的目的是开发和维护一组便于使用的软件过程财富,这些财富改进横跨各项目的过程性能,并为组织积累性的长期得益打下基础。

组织过程定义包括制定和维护组织的标准软件过程,以及相关的过程财富,例如:软件生存周期的描述,过程剪裁指南和准则,组织软件过程数据库和软件过程一有关文档库。

这些财富能以多种方式汇集,取决于对组织过程定义的组织实施。例如软件生存周期的描述可以是组织标准软件过程的一个必备部分,或者软件过程一有关文档库的某些部分可以存储在组织软件过程数据库中。

在开发、实施和维护项目定义软件过程中,可以使用组织软件过程财富。(与项目定义软件过程的制定和维护有关的实践在集成软件管理关键过程区域中加以描述。)

组织过程定义的目标是：

1. 制定和维护组织标准软件过程。
2. 收集和评审有关软件项目使用组织标准软件过程的信息,并使其合用。

组织过程定义所执行的顶层活动是：

1. 按照已文档化的规程制定和维护组织标准软件过程。
2. 按照已制定的组织标准对组织标准软件过程建立文档。
3. 对经批准项目使用的软件生存周期的描述建立文档并进行维护。
4. 制定和维护项目剪裁组织标准软件过程的指南和准则。
5. 建立和维护组织的软件过程数据库。
6. 建立和维护软件过程一有关文档库。

等级3：培训大纲

培训大纲关键过程区域的目的是培育个人的技能和知识,使他们能有效地和效率高地履

行其职责。

培训大纲首先分别识别出组织、项目、和个人所需要的培训，然后展开或设法获得培训以满足所识别的需求。

每个软件项目 B 评价它当前的和未来的技能需求，并决定如何才能获得这些技能。某些技能可有效地和效率高地通过非正式的载体传递（例如在职培训和非正式指导），而其它的技能则需要较正式的培训载体（例如课堂培训和受指导的自学）才能有效地和效率高地传递。必

须选择和使用恰当的载体。

这个关键过程区域仅包括培训组完成培训功能的实践。而识别特定培训专题（即所需要的知识或技能）的实践则包含在各个有关的关键过程区域的执行能力共同特点中。

培训大纲的目标是：

- 1．培训活动是有计划的。
- 2．提供用以培育为履行软件管理和技术角色所需要的技能和知识的培训。
- 3．软件工程组和软件一有关组中的个人受到履行其角色所必须的培训。

培训大纲所执行的顶层活动是：

- 1．每个软件项目制定和维护一个规定其培训需要的培训计划。
- 2．按照已文档化的规程制定和修正组织的培训计划。
- 3．按照组织的培训计划进行组织的培训。
- 4．按照组织标准开发和维护在组织层上准备的培训课程。
- 5．制定免修所要求的培训应遵循的规程，并将其用于确定是否那些个人已具有履行其岗位职责所要求的知识和技能。
- 6．保持培训记录。

等级 3：集成软件管理

集成软件管理的目的是将软件工程活动和管理活动集成为一个协调的、已定义的软件过程，该软件过程是从组织标准软件过程和有关的过程财富剪裁而得到的，这在组织过程定义中加以描述。

集成软件管理包括制定项目定义软件过程并采用此已定义的软件过程去管理软件项目。

项目定义软件过程是组织标准软件过程经剪裁的版本，剪裁是为了针对项目的具体特征。

软件开发计划是基于项目定义软件过程并描述将如何实施和管理项目定义软件过程的活动。软件项目的规模、工作量、成本、进度、人员配置和其它资源等与项目定义软件过程的作业紧密相连。

由于项目定义软件过程全部是由组织标准软件过程剪裁而得到的，因此软件项目能共享过程数据和所吸取的经验教训。

关于估计、策划和跟踪软件项目的基本实践均在软件项目策划和软件项目跟踪和监督等关键过程区域中描述。当问题出现时它们集中注意力于识别问题及调整计划和（或）性能以便解决这些问题。本关键过程区域的实践建立在那两个关键过程区域的实践之上，是对它们的补充。集成软件管理的重点转移到采取预防措施防止问题发生，避免这些问题的后果或使之影响最小。

集成软件管理的目标是；

- 1．项目定义软件过程是组织标准软件过程经剪裁的版本。
- 2.按照项目定义软件过程对项目进行计划和管理。

集成软件管理所执行的顶层活动是：

- 1．按照已文档化的规程剪裁组织标准软件过程来制定项目定义软件过程。

2. 按照已文档化的规程修订每个项目定义软件过程。

3. 按照已文档化的规程制定和修改项目的软件开发计划，该计划描述项目定义软件过程的使用。

4. 按照项目定义软件过程管理软件项目。
5. 将组织的软件过程数据库用于软件策划和估计。
6. 按照已文档化的规程管理软件工作产品的规模（或软件工作产品更改的规模）。
7. 按照已文档化的规程管理项目的软件工作量和成本。
8. 按照已文档化的规程管理项目的关键计算机资源。
9. 按照已文档化的规程管理项目软件进度的关键依赖关系和关键路径。
10. 按照已文档化的规程对项目的软件风险进行识别、评估、建立文档和管理。
11. 定期执行对软件项目的评审以便确定那些必须的行动，它们使得软件项目的性能和结果与经营的、顾客和最终用户的当前需求和设想需求相一致（当合适的时候）。

等级3：软件产品工程

软件产品工程的目的是—致地执行一个妥善定义的工程过程，该过程集成全部软件工程活动，以便有效地且高效率地生产正确的、一致的软件产品。

软件产品工程包括采用项目定义软件过程（在集成软件管理关键过程区域中加以描述）和适当的方法及工具去进行构造与维护软件的工程作业。

软件工程作业包括；分析分配给软件的系统需求（这些系统需求在需求管理关键过程区域中加以描述）、制定软件需求、开发软件的体系结构、设计软件、用代码实现软件、集成软件成分和测试软件以验证它满足所规定的需求（即分配给软件的系统需求和软件需求）。

制定并评审为进行软件工程作业所必须的文档（即软件需求文档、软件设计文档、测试计划和测试规程）以保证每个作业都针对其前面作业的结果，而它所产生的结果又适合于后续的作业（包括运行软件和维护软件的作业）。当更改被批准时，受影响的软件工作产品、计划、约定、过程和活动也要修改以反映该被批准的更改。

软件产品工程的目标是：

1. 定义和集成软件工程作业，并一致地执行它们以生产软件。
2. 软件工作产品彼此间保持一致。

软件产品工程所执行的顶层活动是：

1. 将合适的软件工程方法和工具集成到项目定义软件过程中去。
2. 按照项目定义软件过程，通过系统地分析分配需求来制定软件需求并对其加以维护、建立文档和验证。
3. 按照项目定义软件过程，开发、维护、用文档记载和验证软件设计，以适应软件需求和形成编码的框架。
4. 按照项目定义软件过程，对软件代码进行开发、维护、建立文档和验证，以实现软件需求和软件设计。
5. 按照项目定义软件过程进行软件测试。
6. 按照项目定义软件过程，计划和执行软件集成测试。
7. 计划和执行软件的系统测试和验收测试以显示软件满足其需求。
8. 按照项目定义软件过程编写和维护将用于运行和维护软件的文档。
9. 按照项目定义软件过程收集和分析有关同行评审和测试中所鉴别出的缺陷的数据。
10. 维护软件工作产品间的一致性，工作产品包括；软件计划、过程描述、分配需求、软件需求、软件设计、代码、测试计划和测试规程。

等级3：组间协调

组间协调的目的是建立软件工程组与其它工程组一起积极参与的方式以使得项目更能

够有效地和效率高地满足顾客的需要。

组间协调包括软件工程组和其它项目工程组一起参与阐述系统层的需求、对象和问题。项目工程组的代表，当合适时与顾客和最终用户一起工作，参与建立系统层的需求、对象和计划。这些需求、对象和计划成为全部工程活动的基础。

对组间的技术工作界面和相互作用加以计划和管理以保证整个系统的质量和统一性。各项目工程组的代表参与定期的技术评审和内部交流，以保证所有工程组都清楚各组的状态和计划，并保证系统和组间的问题受到恰当的关注。

和这些工程作业有关的软件一专门实践在需求管理和软件产品工程关键过程区域中描述。

组间协调同目标是：

- 1．顾客需求得到全部受影响的组的认同。
- 2．工程组之间的约定得到受影响的组的认同。
- 3．工程组识别、跟踪和解决组间问题。

组间协调所执行的顶层活动是：

- 1．软件工程组和其它工程组，在合适时与顾客和最终用户一起参与建立系统需求。
- 2．项目软件工程组的代表和其它工程组的代表一起工作去监控和协调技术活动，以及解决技术问题。
- 3．将已文档化的计划用于交流组间约定，协调和跟踪所进行的工作。
- 4．按照已文档化的规程识别、协调和跟踪工程组之间的关键依赖关系。
- 5．所生产的作为其它工程组的输入的工作产品由接收组的代表评审以保证该产品满足他们的需要。
- 6．对项目工程组的个别代表不能解决的组间问题按已文档化的规程加以处理。
- 7．项目工程组的代表进行定期的技术评审和交流。

等级3：同行评审

同行评审的目的是为了及早地和高效率地从软件工作产品中清除缺陷。一个重要的伴随结果是对软件工作产品及可防止的缺陷得到更好的了解。

同行评审就是生产者的同行对软件工作产品进行系统地考察，以便识别缺陷和需作更改的区域。将经受同行评审的具体产品在项目定义软件过程中加以标识，并作为软件项目策划活动的一部分来安排进度，正如在集成软件管理中所描述的。

这个关键过程区域仅包括执行同行评审的实践。而识别需经受同行评审的具体软件工作产品的实践则包含在描述每个软件工作产品的开发和维护的关键过程区域中。

同行评审的目标是：

- 1．同行评审活动是有计划的。
- 2．识别和消除在软件工作产品中的缺陷。

同行评审所执行的顶层活动是：

- 1．计划同行评审，并将计划写成文档。
- 2．按照已文档化的规程进行同行评审。
- 3．记录有关同行评审的执行情况和结果的数据。

等级4：定量过程管理

定量过程管理的目的是定量地控制软件项目的过程性能。软件过程性能表示遵循一软件过程所得到的实际结果。

定量过程管理包括建立集成软件管理关键过程区域所描述的项目定义软件过程的性能目标、测定过程性能、分析这些测量结果并作出调整以保持过程性能在可接受的范围内。当在可接受范围内过程性能是稳定时，项目定义软件过程、相关连的测量和测量的可接受范围

就作为基线建立起来，并被用来定量地控制过程性能。

组织从软件项目采集过程性能数据，并使用这些数据对组织过程定义关键过程区域所描述的组织标准软件过程的过程能力（即一个新项目能预计获得的过程性能）进行特征描述。过

程能力描述从遵循一个软件过程所得到预期结果的范围（即，预计组织承担下一个软件项目时最可能的输出）。软件项目采用这些过程能力数据来建立和修订其过程性能目标和分析项目定义软件过程的性能。

定量过程管理的目标是：

- 1．定量过程管理活动是有计划的。
- 2．项目定义软件过程的过程性能受到定量地控制。
- 3．组织标准软件过程的过程能力是定量地已知的。

定量过程管理所执行的顶层活动是：

- 1．按照已文档化的规程制定软件项目的定量过程管理计划。
- 2．按照项目的定量过程管理计划执行软件项目的定量过程管理活动。
- 3．在项目定义软件过程的基础上确定数据采集的策略和应作的定量分析。
- 4．按照已文档化的规程，采集用于定量地控制项目定义软件过程的测量数据。
- 5．按照已文档化的规程分析项目定义软件过程，并将其置于定量控制之下。
- 6．准备和散发对软件项目的定量过程管理活动结果的书面报告。
- 7．按照已文档化的规程建立和维护组织标准软件过程的过程能力基线。

等级 4：软件质量管理

软件质量管理的目的是建立对项目的软件产品质量的定量了解和实现特定的质量目标。

软件质量管理包含确定软件产品的质量目标，制定实现这些目标的计划，并监控及调整软件计划、软件工作产品、活动和质量目标，以满足顾客和最终用户对高质量产品的需要及愿望。

软件质量管理的实践基于三个关键过程区域，前二个是建立和实施项目定义软件过程的关键过程区域——集成软件管理和软件产品工程，而后一个是定量过程管理关键过程区域，它建立对项目定义软件过程实现所希望结果的能力的定量了解。

基于组织、顾客和最终用户的需要来建立软件产品的质量目标，这样这些目标才能实现。为实现质量目标，组织制定战略和计划，项目则具体调整其已定义的软件过程。

软件质量管理的目标是：

- 1．项目的软件质量管理活动是有计划的。
- 2．软件产品质量的可测目标及其优先级是确定的。
- 3．实现软件产品的质量目标的实际进程是用数量表示的和受到管理的。

软件质量管理所执行的顶层活动是；

- 1．按照已文档化的规程制定和维护项目的软件质量计划。
- 2．项目的软件质量计划是项目软件质量管理活动的基础。
- 3．在整个软件生存周期，确定、监控和修订项目的软件产品的定量质量目标。
- 4．在事件驱动的基础上，对项目软件产品的质量进行测量、分析，并将其与产品的定量质量目标相比较。

- 5．将软件项目的产品定量质量目标恰当地分配给那些向项目交付软件产品的子承包商。

等级 5：缺陷预防

缺陷预防的目的是鉴别缺陷的原因并防止它们再出现。

缺陷预防包括分析过去曾遇到的缺陷和采取特别措施，以防止将来再出现此类缺陷。缺陷可能曾在其它项目中和在当前项目的早期阶段或作业中已经鉴别出来。缺陷预防活动也是

在项目间传播经验教训的一种机制。

为了跟踪已遇到过的缺陷类型和识别出多半会再出现的缺陷，必须对趋势进行分析。基于对项目定义软件过程和其实施方式(正如在集成软件管理和软件产品工程关键过程区域中所描述的) 的理解，确定缺陷的根本原因和对于未来活动缺陷的推断。

项目和组织均采取预防缺陷再出现的特别措施。某些组织的措施可以像在过程更改管理关键过程区域中所描述的那样处理。

缺陷预防的目标是：

1. 缺陷预防活动是有计划的。
2. 缺陷的一般原因已经找出和标识。
3. 缺陷的一般原因已按优先级排序和系统地消除。

缺陷预防所执行的顶层活动是：

1. 软件项目制定和维护一个有关其缺陷预防活动的计划。
2. 在一个软件作业开始时，承担此作业的群组的成员聚在一起准备该作业的活动和有关的缺陷预防活动。
3. 按照已文档化的规程进行原因分析会议。
4. 受指派协调缺陷预防活动的每个群组均定期相聚，评审和协调得自原因分析会议的措施建议的实施情况。
5. 在所有群组协调缺陷预防的活动中，均将缺陷预防数据记入文档并对其进行跟踪。
6. 按照已档化的规程采纳由缺陷预防措施所导致的对组织标准软件过程的修改。
7. 按照已文档化的规程采纳由缺陷预防措施所导致的对项目定义软件过程的修改。
8. 软件工程组和软件一有关组的成员定期收到有关组织的和项目的缺陷预防活动的状态和结果的反馈信息。

等级 5：技术改革管理

技术改革管理的目的是识别出新技术（即工具、方法和过程），并以有序的方式将其引进到组织中去。

技术改革管理包括鉴别、选择和评价新技术，并将有效的技术采用到组织中去。其目的是改进软件质量、提高生产率和缩短产品开发周期。

组织建立一个组（例如软件工程过程组或技术支持组），它与软件项目一起引进和评价新技术共管理对现有技术的改革。特别注意那些多华能改善组织标准软件过程能力的技术改革（有关标准软件过程的描述在组织过程定义关键过程区域中）。

通过保持对软件一有关的技术创新的了解和对它们进行系统的评价与实验，组织选择合适的技术用以改进其软件的质量和其软件活动的效率。在新的和未经证明的技术引入到标准实践中去之前，进行先导性试验以评估它们。当合适时，在组织管理者的恰当支持下，将选出的技术纳入组织的标准软件过程和当前的项目。

按照在过程更改管理关键过程区域中所描述的那样，处理由这些技术改革所造成的对组织标准软件过程（正如在组织过程定义关键过程区域中所描述的）和项目定义软件过程（正如在集成软件管理关键过程区域所描述的）的更改。

技术改革管理的目标是：

1. 技术改革的采纳是有计划的。
2. 评价新技术以确定他们对质量和生产率的作用。
3. 将合适的新技术引入到全组织的标准实践中。

技术改革管理所执行的顶层活动是：

1. 组织制定和维护技术改革管理的计划。
2. 负责组织技术改革管理活动的组和软件项目一起识别技术改革的区域。

3. 不断向软件经理和技术人员通报新技术的情况。
4. 负责组织技术改革管理的组系统地分析组织的标准软件过程，以便识别出需要新技术或能从新技术中获益的区域。
5. 按照已文档化的规程，为组织和软件项目挑选和获取技术。
6. 在合适处，在将新技术引入到标准实践之前，为改进技术作先导性试验。
7. 按照巴文档化的规程，将合适的新技术纳入组织标准软件过程。
8. 按照已文档化的规程，将合适的新技术纳入项目定义软件过程。

等级5：过程更改管理

过程更改管理的目的是本着改进软件质量、提高生产率和缩短产品开发周期的宗旨不断改进组织中所用的软件过程。

过程更改管理包括定义过程改进目标，并且在高级管理者的支持下，在连续的基础上，前摄地（Proactively）和系统地识别、评价和实施对组织标准软件过程和项目定义软件过程的改进。

制定培训和鼓励性计划，使得组织中的每个人能够并鼓励他们参加过程改进活动。识别改进的机会并评价其对组织的可能回报。在将过程更改纳入标准实践之前，进行先导性试验以评估这些过程更改。

当软件过程改进经批准用于标准实践时，在合适时修订组织标准软件过程和项目定义软件过程。在组织过程定义关键过程区域中找到修订组织标准软件过程的实践，在集成软件管理关键过程区域中找到修订项目定义软件过程的实践。

过程更改管理的目标是：

1. 连续的过程改进是有计划的。
2. 组织的软件过程改进活动的参与面是全组织范围的。
3. 组织标准软件过程和项目定义软件过程被连续不断地改进。

过程更改管理所执行的顶层活动是：

1. 制定软件过程改进大纲，该大纲授权组织的成员改进组织的过程。
2. 负责组织软件过程活动的组（例如，软件工程过程组）协调软件过程改进活动。
3. 组织按照巴文档化的规程制定和维护软件过程改进计划。
4. 依照软件过程改进计划进行软件过程改进活动。
5. 按照已文档化的规程处理软件过程改进建议。
6. 组织的成员积极参加群组，对所规定的过程区域展开软件过程改进。
7. 在合适处，在将软件过程改进纳入标准实践之前，在先导性试验基地设置软件过程改进，以便确定其效益和有效性。
8. 当决定将软件过程改进引入到标准实践时，依照已文档化的规程实施改进工作。
9. 保持软件过程改进活动的记录。
10. 在事件驱动的基础上，软件经理和技术人员收到有关软件过程改进活动的状态和结果的反馈信息。

第三篇式 软件过程成熟度提问单

软件过程成熟度提问单

软件工程研究所 卡内基-梅隆大学 匹兹堡, PA 15213 - 3890

软件过程成熟度提问单能力成熟度模型, 1.1 版 1994 年 4 月

本文包含有关对贵软件组织中重要软件实践的实施方面的一些提问。这些提问按关键过程区域——例如软件项目策划和软件配置管理——分组。在每组提问之前有一小段描述每个关键过程区域。除非掌管此提问单的人另有指示, 请基于您当前项目中的知识和经验回答这些提问。

为帮助我们更好地理解那些对有关贵组织中软件过程提问的回答, 本文档以有关您本太软件工作背景的提问作为开始。

请阅读和回答所有提问。如果您想要对任何提问作评论, 或者证明您的回答, 请利用所提供的空白评论处。

鉴定组将对您的回答严格保密。在贵组织中不会鉴别出您的具体回答, 用任何其它方式也不可能鉴别出来。您的姓名仅被用于行政管理的目的; 在响应分析期间引导鉴定组和帮助他们在需要澄清时与您联系。

谢谢您的帮助。

软件工程研究所

卡内基-梅隆大学

匹兹堡, 宾夕法尼亚州。

填写您的回答

我们将用光学扫描技术输入您的回答, 因此请您在整个提问单中整洁地打印或书写。

- 如果您需要更多的空间书写回答或其评论, 您可随意使用空白处。
- 请将您的标记保持在查验框内。任何标记均需如此
- 您应使用蓝黑或黑色墨水的钢笔。

术语定义

本提问单所基于的能力成熟度模型采用一些术语, 在贵组织中他们可能有不同的用法。

· 组织方面的术语在蓝色挂图上加以定义。您可能希望现在就复习它, 并在您完或提问单过程中必要时随时参考它。

- 技术方面的术语在使用它们的那页上加以定义。

回答者身份

(请详细说明)

姓名 :

日期 :

项目名 :

工作电话 :

第一部分回答者背景

1. 哪一项能最好地描述您当前的地位? (请标出所有适用的框)

项目或群组领导

经理

技术成员

软件工程过程组 (SEPG) 成员

其它 (请具体说明)

2. 您当前从事哪项活动？（请标出所有适用框）

软件需求

软件质量保证

软件设计

配置管理

编码和单元测试

软件过程改进

测试和集成

其它（请具体说明）

3. 您是否接受过任何有关 CMM 的培训？ 否 是（情描述）

4. 您的软件经历在以下方面有多少年？（请对每一方面具体说明）

在您现在的组织中？..... 年

您的整个软件经历？..... 年

5. 您是否参加过以前形式的软件过程评估、软件能力评价和 / 或其它形式的软件过程鉴定？（请标出一个框）

否

是 多少次？（请对每类详细说明）

SPA（软件过程评估）的次数

SCE（软件能力评价）的次数

其它基于 SEI 方法的评估次数（清简要描述：例如，小型评估（mini - assess-ments）

或瞬时剖面（Instantprdiles）

基于非 SEI 方法的过程改进工作的次数（清简要描述；例如 ISO · 9000 / 9001 审计）

第二部分软件实践

答题须知

1. 在每个提问的右边，有相应于四种可能回答的框：是，否，不适用，和不知道。

选择是，当：

· 实践被很好地建立和一致地执行。

一实践应该几乎总是按照标准操作规程执行，才能被认为是很好建立的和一致执行的。

选择不是，当：

· 实践未被很好地建立或一致地执行。

一该实践可能有时被执行，或者甚至是经常被执行，但在困难情况下，它被忽略。

选择不适用，当；

· 您对该项目或组织以及所提问题具有所要求的知识，但您觉得此提问对该项目不适用。

一例如，如果您不和子承包商一起工作，那么“软件予合同管理”的整个章节就可能不适用于该项目。

选择不知道，当：

· 您不能确定该如何回答此提问。

2. 您对提问回答的任何详细说明或适用性证明，请利用评论空间。

3. 对每个提问选择其框中的一个。请回答所有的提问。

需求管理的目的是在顾客和软件项目之间建立对顾客需求的共同理解，顾客需求将由软件项目处理。需管理包括和顾客一起建立和维护有关软件项目需求的协议。协议既包括技术需求又包括非技术需求（例如交付日期）。该协构成估计、策划、执行、和跟踪整个软件生存周期内软件项目活动的基础。每当改变分配给软件的系统需求时，都要调整受到影响的图件计划、工作产品和活动，使其与更新后的需求保持一致。

分配需求（allocatedrequirements）（分配给软件的系统需求）——拟由系统的软部件实现的系统需求的子集。分配需求是软件开发计划的主要输入。软件需求分析详细阐述和精炼分配需求，产生文档化的软件需求。

方针 (policy) —— 一种指导原则 , 一般由高级管理者制定 , 组织或项目采用它去影响和确定决策。

软件计划 (softwareplans) —— 用于表达将如何进行软件开发和 / 或维护活动的计划的集合 , 既可是正式计划 , 也可是非正式计划。这样的计划的例子有 : 软件开发计划、软件质量保证计划、软件配置管理计划、软件测试计划、风险管理计划和过程改进计划。

软件质量保证 (softwarequalityassurance (SQA)) —— (1) 为使人充分相信一个软件工作产品符合规定的技术要求所必须采取的全部行动的一种有计划和有组织的模式。(2) 为评价软件工作产品的开发和 / 或维护过程而设计的一组活动。

软件工作产品 (softwareworkproduct) —— 作为定义、维护或使用一个软件过程的一部分所产生的任何人工制品 , 包括过程描述、计划、规程、计算机程序和相关的文档 , 无论是否打算将它们交付给顾客或最终用户。

是 否 不适用 不知道

1 用分配给软件的系统需求来建立供软件工程和管理用的基线 ?

评论 :

2 分配给软件的系统需求更改时 , 是否对软件计划、工作产品和活动作出必要的调整 ?

评论 :

3 项目是否遵循一个书面的组织对分配给软件的系统需求的管理方针 ?

评论 :

4 项目中负责管理分配需求的人员是否受到管理分配需求所用规程的培训 ?

评论

5 是否用测量量来确定管理分配需求所进行的活动的状态 (例如 , 所提议的、尚未解决的、已批准的和已纳入基线的需求更改的总数) ?

评论 :

6 管理该项目分配需求的活动是否受到 SQA 评审 ?

评论 :

软件项目策划的目的是为进行软件工程活动和管理软件项目制定合理的计划。软件项目策划包括估计待完成的工作 , 建立必要的约定和确定进行该工作的计划。

约定 (commitment) —— 自由地采用的、可视的和期待各方遵守的协议。

事件驱动的评审或活动 (event - drivenreview / activity) —— 根据项目中某事件的发生而进行的评审或活动 (例如 : 一项正式评审或一个软件生存期阶段的完成)。

方针 (policy) —— 一种指导原则 , 一般由高级管理者制定 , 组织或项目采用它去影响和确定决策。

软件计划 (software plans) —— 用于表达将如何进行软件开发和 (或) 维护活动的计划集合 , 既可是正式计划 , 也可是非正式计划。这样的计划的例子有 ; 软件开发计划、软件质量保证计划、软件配置管理计划、软件测试计划、风险管理计划和过程改进计划。

是 否 不适用 不知道

1 供策划和跟踪软件项目用的估计 (例如 : 规模、成本和进度估计) 是否已文档化 ?

评论 :

2 软件项目计划是否将拟进行的活动和对软件项目所作的约定文档化？

评论：

3 所有受影响的组和个人对他们有关软件项目的约定是否同意？

评论：

4 项目是否遵循一个书面的用于策划软件项目的组织方针？

评论：

5 是否为策划软件项目提供足够的资源（例如，资金和有经验的个人）？

评论：

6 是否用测量量来确定策划软件项目活动的状态（例如，项目策划活动里程碑的完成情况与计划相比较）？

评论：

7 项目经理是否既定期地又事件驱动地评审策划软件项目的活动？

评论：

软件项目跟踪和监督的目的是提供适当的对实际进展的可视性，使管理者能在软件项目性能明显偏离软件计划时采以纠正措施。纠正措施可以包括修订软件开发计划以反映实际的完成情况和重新策划遗留的工作或者采取改进性能的措施。软件项目跟踪和监督包括对照文档化的估计、约定和计划评审和跟踪软件完成情况和结果，以及在实际完成情况和结果的基础上调整这些计划。

约定（commitment）——自由地采用的、可视的和期待各方遵守的协议。

定期评审或活动（periodicreview / activity）——在规定的固定时间间隔处出现的评审或活动，而不是在主要事件完成处进行的。

方针（policy）——一种指导原则，一般由高级管理者制定，组织或项目采用它去影响和确定决策。

软件计划（software Plans）——用于表达将如何进行软件开发和（或）维护活动的计划集合，既可是正式计划，也可是非正式计划。这样的计划的例子有；软件开发计划、软件质量保证计划、软件配置管理计划、软件测试计划、风险管理计划和过程改进计划。

软件工作产品（softwarework product）——任何所产生的作为定义、维护或使用一个软件过程的一部分的人工制品，包括过程描述、计划、规程、计算机程序和相关的文档，无论是否打算将它们交付给顾客或终端用户。

是 否 不适用 不知道

1 项目的实际结果（例如，进度、规模和成本）是否与软件计划中的估计相比较？

评论：

2 当实际结果明显偏离项目软件计划时是否采取纠正措施？

评论：

3 所有受影响的组和个人是否同意对软件约定的更改？

评论：

4 项目是否遵循一书面的用于跟踪和控制其软件开发活动的组织方针？

评论：

5 是否安排项目中的某人专门负责跟踪软件工作产品和活动
(例如,工作量和、进度和预算)?

评论:

6 是否用测量量来确定软件跟踪和监督活动的状态(例如,
在进行跟踪和监督的活动中所化费的总工作量)?

评论:

7 高级管理者是否定期参与评审软件项目跟踪和监督的活动
(例如:项卧佳能、待解决的问题、风险、和行动条款)?

评论:

软件子合同管理的目的是选择合格的软件予承包商并有效地管理他们。软件予合同管理包括选择软件予承包商、建立和子承包商的约定,以及跟踪和评审子承包商的性能和结果。这些实践包括对纯软件予合同的管理,也包括对于合同的软件成分的管理,后者子合同中含有软件、硬件和可能有的其它系统成分。

文档化的规程(documented procedure)——对为了完成一给定作业将采取的行动步骤的书面描述。[IEEE - STD - 610 Glossary]

事件驱动的评审或活动(event-driven review / activity)——根据项目中某事件的发生而进 if 的评审或活动(例如:一项正式评审或一个生存周期阶段的完成)。

定措评审成活动(periodic review / activity)——在规定的固定时间间隔处出现的评审或活动而不是在主要事件完成时进行的。

方针(policy)——一种指导原则,一般由高级管理者制定,组织或项目采用它去影响和确定决策。

是 否 不适用 不知道

1 是否采用文档化的规程,根据于承包完成工作的能力挑选手
承包商?

评论:

2 子合同的更改是否得到主承包商和子承包商双方的同意?

评论:

3 是否和子承包商进行定期的技术交流?

评论:

4 是否针对其约定跟踪软件予承包商的结果和性能?

评论

5 项目是否遵循一书面的管理软件予合同的组织方针?

评论;

6 负责管理软件子合同的人员是否经过管理软件子合同方面的
培训?

评论;

7 是否用测量量来确定管理软件予合同活动的状态(例如,就
计划的交付日期来说的进度状态和管理子合同所化资的工作
量)?

评论;

8 项目经理是否既定期地又事件驱动地参与评审软件子合同的
活动?

评论:

软件质量保证的目的是向管理者提供对软件项目正使用的过程和正构造的产品的适当

的可视性。软件质量保证包括评审和审计软件产品和活动以验证它们符合适用的规程和标准，以及向软件项目经理和其它合适的经理提供这些评审和审计的结果。

审计 (audit)——对一个或一组工作产品所作的独立考察，评估它们与规格说明、标准、合同协议或其它准则的符合性。[IEEE - STD - 610Glossary]

定期评审或活动 (periodicreview / activity)——在规定的固定时间间隔处出现的评审或活动，而不是在主要事件完成处进行的。

方针 (policy)——一种指导原则，一般由高级管理者制定，组织或项目采用它去影响和确定决策。

规程 (procedure)——对为了完成给定作业将采取的行动步骤的书面描述。[IEEE - STD - 610Glossary]

软件质量保证 (SQA) —— (1) 为使人充分相信一个软件工作产品符合规定的技术要求，所必须采取的全部行动的一种有计划和有组织的模式。(2) 为评价软件工作产品的开发和 (或) 维护过程而设计的一组活动。

标准 (standard) ——为规定一种有纪律的、一致的软件开发方法而采用的强制实施的要求。

是 否 不适用 不知道

1SQA 活动是否计划好的？评论：

2SQA 是否就软件工作产品和活动符合适用的标准、规程和要求提供客观的验证？

评论：

3 是否将 SQA 评审和审计的结果提供给受影响的组和个人(例如，进行该项工作的人员和负责该项工作的人员) ？

评论：

4 在软件项目内不能解决的不符合 (不合格) 问题是否由高级管理者解决 (例如偏离适用的标准) ？

评论：

5 项目是否遵循一书面的实施 SQA 的组织方针？

评论：

6 是否为进行 SQA 活动提供足够的资源 (例如资金和一位指定的接收和处理不符合项 (不合格品) 的经理) ？

评论：

7 是否用测量量来确定进行 SQA 活动的成本和进度状态(例如，所完成的工作、所花费的工作量和资金与计划相比较) ？

评论：

8 高级管理者是否定期地参与评审 SQA 活动？

评论：

软件配置管理 (SCM) 的目的是建立和维护在项目的整个生存周期内软件项目产品的完整性。软件配置管理包括标识在给定时间点上软件的配置 (即选定的软件工作产品及其描述)，系统地控制对配置的更改，并在整个软件生存周期中维护配置的完整性和可跟踪性。置于软件配置管理之下的工作产品包括交付给顾客的软件产品，以及与这些产品一起标识的产品项或产生这些软件产品所要求的产品项。

审计 (audit)——对一个或一组工作产品所作的独立考察，评估它们与规格说明、标准、合同协议或其它准则的符合性。[IEEE - STD - 610 Glossary]

配置项 (configuration item) ——指定供配置管理用的且在配置管理过程中作为单个实

体处理的硬件集合、软件集合、或硬软件集合。[IEEE - STD - 610 Glossary]

文档化规程 (documented procedue) ——对为了完成给定作业将采取的行动步骤的书面描述。[IEEE - STD - 610 Glossary]

方针 (policy) ——一种指导原则，一般由高级管理者制定，组织或项目采用它去影响和确定决策。

软件基线 (software baseline) ——经过正式评审和认定的一组配置项 (软件文档和软部件)，此后它们作为进一步开发的基础，并且只有通过正式的更改控制规程才能被更改。

软件工作产品 (software workproduct) ——作为定义、维护或使用一个软件过程的一部分所产生的任何人工制品，包括过程描述、计划、规程、计算机程序和相关的文档，无论是否打算将它们交付给顾客或终端用户。

是 否 不适用 不知道

1 关于项目的软件配置管理活动是否计划好的？

评论；

2 通过运用配置管理，项目是否已经标识、控制软件工作产品，并使其可用？

评论：

3 项目是否遵循一文档化的规程来控制配置项或单元的更改？

评论：

4 是否将有关软件基线的标准报告 (例如，软件配置控制委员会会议记录和更改申请简报，以及状态报告) 分发给受影响的组和个人？

评论：

5 项目是否遵循一书面的实施软件配置管理活动的组织方针？

评论：

6 项目人员是否经过培训以便能完成他们所负责的软件配置管理活动？

评论：

7 是否用测量量来确定软件配置管理活动的状态 (例如，软件配置管理活动所化资的工作量和资金) ？

评论；

8 是否进行定期审计去验证软件基线与定义它们 (例如由 SCM 组定义) 的文档相符？

评论；

组织过程焦点的目的是建立组织对一些软件过程活动的责任，这些活动能改进组织整体软件过程能力。组织过程焦点包括增进和保持对组织的和项目的软件过程的了解，协调那些评估、制定、维护和改进这些过程的活动。组织提供长期的承诺和资源，通过诸如软件工程过程组那样的组来协调那些与当前和未来软件项目都有关的软件过程的制定和维护工作。这个组负来组织的软件过程活动。

定现评审或活动 (periodic review / activity) ——在规定的固定时间间隔处出现的评审或活动，而不是在主要事件完成时进行的。

软件过程 (software process) ——人们用以开发和维护软件及其相关产品 (例如：项目计划、设计文档、代码、测试用例和用户手册) 的一组活动、方法、实践、和变换。

软件过程评估 (software process assessment) ——由一个经培训的软件专业人员群组所作的一项鉴定，旨在确定一个组织当前软件过程的状态，确定组织所面临的具有高优先级的

软件过程有关的问题和获得组织对软件过程改进的支持。

是 否 不适用 不知道

1 在整个组织中，制定和改进组织及项目的软件过程的活动是否协调（例如，通过一个软件工程过程组）？

评论：

2 贵组织的软件过程是否受到定期的评估？

评论：

3 贵组织是否遵循一文档化的制定和改进组织软件过程的计划？

评论：

4 高级管理者是否提倡组织的软件过程制定和改进的活动（例如，通过建立长期规划，通过对资源和资金的承诺）

评论：

5 是否有一个或多个个人用全部时间或部分时间负责组织的软件过程活动（例如一个软件工程过程组）？

评论：

6 是否用测量量来确定为制定和改进组织的软件过程而进行的活动状态（例如，软件过程评估和改进所花费的工作量）？

评论：

7 高级管理者是否定期地参与评审为制定和改进软件过程所进行的活动？

评论：

组织过程定义的目的是开发和维护一组便于使用的软件过程财富，这些财富改进横跨各项目的过程性能，并为组织积累性的长期得益打下基础。组织过程定义包括制定和维护组织标准软件过程以及相关的过程财富，例如：软件生存周期的描述、过程剪裁指南和准则、组织软件过程数库和软件过程有关的文档库。

审计（audit）——对一个或一组工作产品所作的独立考察，评估它们与规格说纷、标准、合同协议或其它准则的符合性。

组织标准软件过程（organization's standard software process）——基本过程的可操作的定义，基本过程指导对一个组织内的所有项目建立一个共同的软件过程。它描述预期每个软件项目的项目定义软件过程均会包含的基本软件过程元素。它还描述这些软件过程元素间的关系（例如：排序和界面）。

方针（policy）——一种指导原则，一般由高级管理者制定，组织或项目采用它去影响和决定决策。

软件质量保证（SQA）——（1）为使人充分相信一个软件工作产品符合规定的技术要求所必须采取的全部行动的一种有计划和有组织的模式。（2）为评价软件工作产品的开发和/或维护过程而设计的一组活动。

是 否 不适用 不知道

1 贵组织是否已制定并维护一个标准软件过程？

评论：

2 贵组织是否收集和评审与使用组织标准软件过程有关的信息，并使其可用（例如有关软件规模、工作量和成本的估计值和实际数据；生产率数据；和质量度量）？

评论：

3 组织是否遵循一书面的用于制定和维护其标准软件过程和相关过程财富的组织方针（例如，对经批准的软件生存周期的描述）？

评论：

4 制定和维护组织标准软件过程的个人是否受到所要求的进行这些活动的培训？

评论：

5 是否用测量量来确定为定义和维护组织标准软件过程而进行的活动的状态（例如，进度里程碑的状态和过程定义活动的成本）？

评论：

6 制定和维护组织标准软件过程的活动和工作产品是否受到 SQA 的评审和审计？

评论：

培训大纲关键过程区域的目的是培育个人的技能和知识，使他们能有效地和效率高地履、行其职责。培训大纲包括：首先，识别组织、项目和个人的培训要求，然后，展开或获得培训以满足所识别的需求。某些技能可有效地和效率高地通过非正式载体传授（例如在职培训和非正式指导），而其它的技能则需要较正式的培训载体（例如课堂培训和受指导的自学）才能有效地和效率高地传授。选择和使用恰当的载体。

定期评审成活动（periodicreview / activity）——在规定的固定时间间隔处出现的评审或活动，而不是在主要事件完成时进行的。

方针（policy）——一种指导原则，一般由高级管理者制定，组织或项目采用它去影响和决定决策。

是 否 不适用 不知道

1 培训活动是否计划好的？

评论：

2 是否为培育履行软件管理和技术角色所需要的技能和知识提供培训？

评论：

3 软件工程组和其它软件有关组的成员是否受到履行其角色所必须的培训？

评论：

4 贵组织是否遵循一书面的组织方针去满足其培训需求？

评论：

5 是否为实施组织的培训大纲提供足够的资源（例如：资金、软件工具、适当的培训设施）？

评论：

6 是否用测量量来确定培训大纲的质量？

评论：

7 高级管理者是否定期地参与培训大纲活动的评审？

评论：

集成软件管理的目的是将软件工程活动和管理活动集成为一个协调的、已定义的软件过程，该软件过程是对组织标准软件过程和有关的过程财富剪裁而得到的。集成软件管理包括制定项目定义软件过程并采用此已定义的软件过程去管理软件项目。项目定义软件过程是组

组织标准软件过程经剪裁的版本，剪裁是为了针对项目的具体特征。软件开发计划基于项目定义软件过程并描述将如何实施和管理项目定义软件过程的活动。

审计 (audit)——对一个或一组工作产品所作的独立考察，评估它们与规格说明、标准、合同协议或其它准则的符合性。

组织标准软件过程 (Organization's standard software process)——基本过程的可操作的定义，基本过程指导对一个组织内的所有项目建立一个共同的软件过程。它描述预期每个软件项目的项目定义软件过程均会包含的基本软件过程元素。它还描述这些软件过程元素间的关系 (例如：排序和界面)。

方针 (policy)——一种指导原则，一般由高级管理者制定，组织或项目采用它去影响和确定决策。

项目定义软件过程 (project's defined software process)——项目所用软件过程的可操作的定义。项目定义软件过程是一个已很好特征化的和已理解的软件过程，用软件标准、规程、工具和方法予以描述。通过剪裁组织标准软件过程使之适合项目特定特征的方法来制定它。

软件质量保证 (SQA)——(1) 为使人充分相信一个软件工作产品符合规定的技术要求所必须采取的全部行动的一种有计划和有组织的模式。(2) 为评价软件工作产品的开发和 (或) 维护过程而设计的一组活动。

剪裁 (tailoring)——为更好地匹配过程要求或产品要求而修改过程、标准、或规程。

是 否 不适用 不知道

1 是否通过剪裁组织标准软件过程来制定项目定义软件过程？

评论：

2 是否按照项目定义软件过程对项目进行策划和管理？

评论：

3 项目是否遵循一书面的组织方针，该方针要求使用组织标准软件过程去策划和管理软件项目？

评论：

4 是否要求被分配做剪裁组织标准软件过程去定义新项目的软件过程工作的个人接受培训？

评论：

5 是否用测量量来确定集成软件管理活动的有效性 (例如：重新策划工作的频率、原因、和数量)？

评论：

6 用于管理软件项目的活动和工作产品是否受到SQA的评审和审计？

评论：

软件产品工程的目的是—致地执行一个妥善定义的工程过程，该过程集成全部软件工程活动，以便有效且高效率地生产正确的、—致的软件产品。软件产品工程包括采用项目定义软件过程和适当的方法及工具去构造与维护软件的工程作业。软件工程作业包括：分析分配给软件的系统需求、开发软件的体系结构、设计软件、用代码实现软件、测试软件以验证它们满足所规定的需求。

审计 (audit)——对一个或一组工作产品所作的独立考察，评估它们与规格说明、标准、合同协议或其它准则的符合性。

方针 (policy)——一种指导原则，一般由高级管理者制定，组织或项目采用它去影响和确定决策。

项目定义软件过程 (project ' sdefinedsoftwareprocess) ——项目所用软件过程的可操作的定义。项目定义软件过程是一个已很好特征化的和已理解的软件过程，用软件标准、规程、工具和方法予以描述。通过剪裁组织标准软件过程以适合项目特定特征的方法来制定它。

软件质量保证 (SQA) —— (1) 为使人充分相信一个软件工作产品符合规定的技术要求所必须采取的全部行动的一种有计划和有组织的模式。 (2) 为评价软件工作产品的开发和 (或) 维护过程而设计的一组活动。

软件工作产品 (software work product) ——作为定义、维护或使用一个软件过程的一部分所产生的任何人工制品，包括过程描述、计划、规程、计算机程序和相关的文档，无论是否打算将它们交付给顾客或终端用户。

是 否 不适用 不知道

1 是否按照项目定义软件过程生产软件工作产品？

评论：

2 各软件工作产品之间的一致性是否得到保持 (例如从软件需求、设计、代码一直到测试用例的文档是否始终保持跟踪分配需求) ？

评论：

3 项目是否遵循一书面的进行软件工程活动的组织方针 (例如：要求使用适当的方法和工具来构造和维护软件产品的方针) ？

评论：

4 是否为进行软件工程作业提供足够的资源 (例如，资金、有技能的个人、和适当的工具) ？

评论：

5 是否用测量量来确定软件产品的功能性和质量 (例如所鉴别出缺陷的数量、类型和严重性) ？

评论：

6 软件产品工程的活动和工作产品是否受到SQA的评审和审计 (例如，是否进行所要求的测试？从软件需求、设计、代码一直到测试用例是否始终跟踪分配需求？) ？

评论：

组间协调的目的是为软件工程组建立一种与其它工程组一起积极参与的方式以使得项目能更加有效地和效率高地满足顾客的需要。组间协调包括软件工程组和其它项目工程组一起参与阐述系统层的需求、目的和问题。合适时，项目各工程组的代表通过与顾客和终端用户一起工作，参与制定系统层的需求、目的和计划。这些需求、目的和计划成为全部工程活动的基础。

约定 (commitment) ——自由地采用的、可视的、期待各方遵守的协议。

事件驱动评审成活动 (event - driven review / activity) ——基于项目中某事件的发生而执行的评审或活动 (例如：一项正式评审或一个软件生存周期阶段的完成)。

定期评审或活动 (periodic review / activity) ——在规定的固定时间间隔处出现的评审或活动，而不是在主要事件完成时进行的。

方针 (poliCy) ——一种指导原则，一般由高级管理者制定，组织或项目采用它会影响和确定决策。

是 否 不适用 不知道

1 对项目来说，软件工程组和其它工程组是否与顾客合作去制定系统需求？

评论：

2 工程组是否认同在总的项目计划中提出的约定？

评论：

3 工程组是否识别、跟踪和解决组间问题（例如，不协调的进度、技术风险、或系统层的问题）？

评论：

4 是否存在一书面的指导建立跨学科工程群组的组织方针？

评论：

5 不同工程组所用的支持工具是否使有效的通讯和协调成为可能（例如：相兼容的字处理系统、数据库系统、和问题跟踪系统好）？

评论：

6 是否用测量量来确定组间协调活动的状态（例如：软件工程组在支持其它组上所花费的工作量）？

评论：

7 项目经理是否定期地和事件驱动地参与评审组间协调的活动？

评论：

同行评审的目的是为了及早地和高效率地从软件工作产品中消除缺陷。一个重要的伴随结果是对软件工作产品及可防止的缺陷得到更好的了解。同行评审就是生产者的同行为了识别缺陷和需作更改的区域对软件工作产品进行系统地考察。拟经受同行评审的具体产品在项目定义软件过程中加以标识，并且其进度被作为软件项目策划活动的一部分加以安排。

审计（audit）——对一个或一组工作产品所作的独立考察，评估它们与规格说明、标准、合同协议或其它准则的符合性。

同行评审（peerreview）——由软件工作产品生产者的同行遵循已定义的规程对该产品进行的评审，目的在于识别出缺陷和改进之处。

方针（policy）——一种指导原则，一般由高级管理者制定，组织或项目采用它去影响和确定决策。

软件质量保证（SQA）——（1）为使人们充分相信一个软件工作产品符合规定的技术要求所必须采取的全部行动的一种有计划和有组织的模式。（2）为评价软件工作产品的开发和（或）维护过程而设计的一组活动。

（软件工作产品（softwareworkproduct）——作为定义、维护或使用一个软件过程的一部分所产生的任何人工制品，包括过程描述、计划、规程、计算机程序和相关的文档，无论是否打算将它们交付给顾客或终端用户。

是 否 不适用 不知道

1 同行评审是否计划好的？

评论：

2 与同行评审中识别出的缺陷有关的行动是否受到跟踪直到它们被解决？

评论：

3 项目是否遵循一书面的进行同行评审的组织方针？

评论：

4 同行评审的参加者是否受到为履行其角色所要求的培训？

评论；

5 是否用测量量来确定同行评审活动的状态（例如；同行评审次数、有关同行评审所花费的工作量、受评审的工作产品数等与计划相比较）？

评论：

6 同行评审的活动和工作产品是否受到 SQA 的评审和审计（例如，是否进行了计划好的评审，是否跟踪了后续措施）

评论：

定量过程管理的目的是定量地控制软件项目的过程性能。定量过程管理包括测量过程性能，分析这些测量，并作出调整以保持过程性能在可接受的范围内。当过程性能稳定在可接受的范围内时，就将该项目定义软件过程、相关连的测量和测量的可接受范围作为基线建立起来，用来定量地控制过程性能。

—事件驱动评审成活动（event - driven review / activity）——基于项目中某事件的发生而进行的评审或活动（例如：一项正式评审或一个软件生存周期阶段的完成）。

组织标准软件过程（organization ' s standard software process）——基本过程的可操作的定义，基本过程指导对一个组织内所补项目建立一个共同的软件过程。它描述预期每个软件项目的项目定义软件过程均产包含的基本软件过程元素。它还描述这些软件过程元素间的关系（例如：排序和界面）。

定期评审或活动（periodic review / activity）——在规定的固定时间间隔处出现的评审或活动，而不是在主要事件完成时进行的。

方针（policy）——一种指导原则，一般由高级管理者制定，组织或项目采用它去影响和确定决策。

过程能力（process capability）——通过遵循一个过程能实现预期结果的程度。

项目定义软件过程（project ' s defingd software Process）——项目所用软件过程的可操作的定义。项目定义软件过程是一个已很好特征化的和已理解的软件过程，用软件标准、规模、工具和方法予以描述。通过剪裁组织标准软件过程以适合项目特定特征的方法来制定它。

是 否 不适用 不知道

1 项目是否遵循文档化的进行定量过程管理的计划？

评论：

2 是否定量地控制项目定义软件过程的性能（例如通过使用定量分析方法）？

评论：

3 组织标准软件过程的过程能力是否是定量地已知的？

评论：

4 项目是否遵循一书面的用以测量和控制项目定义软件过程性能的组织方针（例如；如何识别、分析和控制变化的特殊原因的项目计划）？

评论：

5 是否为定量过程管理活动提供足够的资源（例如：资金、软件支持工具、和组织测量大纲）？

评论：

6 是否用测量量来确定定量过程管理活动的状态（例如：定量过程管理活动的成本和定量过程管理活动的里程碑的完成情况）？

评论：

7 项目经理是否既定期地又事件驱动地参与评审定量过程管理的活动？

评论：

软件质量管理包括确定软件产品的质量目标，制定实现这些目标的计划、并监控及调整软件计划、软件工作产品、活动和质量目标以满足顾客和终端用户的需要和愿望。根据组织、顾客和终端用户对高质量产品的需求来建立定量的产品质量目标，这样，这些目标才能实现。

为了实现质量目标，组织制定战略和计划而项目则具体调整其已定义的软件过程。

定期评审活动（periodicreview / activity）——在规定的固定时间间隔处出现的评审或活动，而不是在主要事件完成时进行的。

方针（policy）——一种指导原则，一般由高级管理者制定，组织或项目采用它去影响和确定决策。

是 否 不适用 不知道

1 对于项目，管理软件质量的活动是否计划好的？

评论：

2 项目是否采用可测量的和排好优先次序的目标来管理其软件产品的质量（例如：功能性、可靠性、可维护性和易用性）？

评论：

3 是否将质量的测量值与软件产品的质量目标相比较以确定是否达到质量目标？

评论：

4 项目是否遵循一书面的管理软件质量的组织方针？

评论：

5 软件工程组和其它软件一有关组的成员是否受到在软件质量管理方面所要求的培训（例如，在采集测量数据及定量管理产品质量的好处等方面的培训）？

评论：

6 是否用测量量来确定管理软件质量活动的状态（例如：质量低劣所造成的成本）？

评论：

7 高级管理者是否定期地参与评审软件质量管理所进行的活动？

评论：

缺陷预防包括分析过去曾遇到的缺陷和采取具体措施以防止将来再出现此类缺陷。这些缺陷可能是在其它项目中也可可是在当前项目的早期阶段或作业中，被鉴别出来的。为了跟踪已遇到过的缺陷类型和识别出多半会再出现的缺陷，必须对趋势进行分析。项目和组织均应采取具体的预防缺陷再出现的措施。

审计（audit）——对一个或一组工作产品所作的独立考察，评估它们与规格说明、标准、合同协议或其它准则的符合性。

原因分析会议（causalanalysismeeting）——在完成一特定作业后，为分析作业完成期间所暴露的缺陷而举行的会议。

（缺陷）的一般原因（commoncause（ofadefect））——一个过程或系统的一部分所固有的缺陷原因。一般原因影响过程的每个输出和工作在此过程中的每个人。

方针（policy）——一种指导原则，一般由高级管理者制定，组织或项目采用它去影响

和确定决策。

软件质量保证 (SQA) —— (1) 为使人们充分相信一个软件工作产品符合规定的技术要求所必须采取的全部行动的一种有计划和有组织的模式。(2) 为评价软件工作产品的开发和 / 或维护过程而设计的一组活动。

是 否 不适用 不知道

1 缺陷预防活动是否计划好的？

评论：

2 项目是否举行鉴别缺陷的一般原因的原因分析会议？

评论：

3 一旦识别出缺陷的一般原因，是否将它们排序并系统地消除它们？

评论：

4 项目是否遵循一书面的有关缺陷预防活动的组织方针？

评论：

5 软件工程组和其它软件一有关组的成员是否受到在行其缺陷预防活动方面所要求的培训？

评论：

6 是否用测量量来确定缺陷预防活动的状态（例如，识别和更正缺陷所用的时间和成本以及所建议的、尚待进行的和已完成的行动项数）？

评论：

7 有关缺陷预防的活动和工作产品是否受到 SQA 的评审和审计？

评论：

技术改革管理包括鉴别、选择和评价新技术、并将有效的技术引进到组织中去。其目的是改进软件质量、提高生产率和缩短产品开发周期。组织建立一个组（例如软件工程组或技术支持组），它与软件项目一起工作以引进和评价新技术及管理对现有技术的改革。特别注意那些多半能改善组织标准软件过程能力的技术改革。在新的和未经证明的技术被纳入标准实践中去之前，进行先导性试验以评估这些技术。当合适时，在组织管理者的适当保证下，将选出的技术纳入组织标准软件过程和当前的项目。

文档化的规程 (Documented procedure) —— 对为了完成一给定作业将采取的行动步骤的书面描述。[IEEE - STD - 610 Glossary]

组织标准软件过程 (organization's standard software process) —— 基本过程的可操作的定义，基本过程指导对一个组织内的所有项目建立一个共同软件过程。它描述预期每个软件项目的项目定义软件过程均会包含的基本软件过程元素。它还描述这些软件过程元素间的关系（例如：排序和界面）。

定期评审或活动 (periodic review / activity) —— 在规定的固定时间间隔处出现的评审或活动。

是 否 不适用 不知道

1 组织是否遵循一个管理技术改革的计划？

评论：

2 是否评价新技术以确定他们对质量和生产率的功效？

评论：

3 组织是否遵循一个文档化的将新技术引入组织标准软件过程

的规程？

评论：

4 高级管理者是否主办组织对技术改革的管理活动（例如，通过制定长期计划和作出资金、人员配置、和其它资源上的承诺）？

评论：

5 是否存在过程数据来帮助选择新技术？

评论：

6 是否用测量量来确定组织在技术改革管理方面的活动的状态（例如，实施技术改革的效果）？

评论：

7 高级管理者是否定期地参与评审在技术改革管理方面组织上的活动？

评论：

过程更改管理包括确定过程改进目标，并在高级管理者的主办下，在连续的基础上系统地识别、评价和实施对组织标准软件过程和项目定义软件过程的改进。制定培训和鼓励大纲，使得组织中每个人能够参加过程改进活动并鼓励他们参加过程改进活动。识别改进的机会并评价其对组织的可能回报。在将过程更改纳入标准实践之前，进行先导性试验以评估过程更改。当软件过程更改经批准用于标准实践时，在合适时修订组织标准软件过程和项目定义软件过程。

文档化的规程（Documented procedure）——对为了完成一给定作业将采取的行动步骤的书面描述。 [IEEE - STD - 610 Glossary]

组织标准软件过程（Organization's standard software process）——基本过程的可操作的定义，基本过程指导对一个组织内的所有项目建立一个共同的软件过程。它描述预期每个软件项目的项目定义软件过程均会包含的基本软件过程元素。它还描述这些软件过程元素间的关系（例如；排序和界面）。

定期评审或活动（Periodic review / activity）——在规定的固定时间间隔处出现的评审或活动，而不是在主要事件完成时进行的。

方针（policy）——一种指导原则，一般由高级管理者制定，组织或项目采用它去影响和确定决策。

项目定义软件过程（Project's defined software process）——项目所用软件过程的可操作的定义。项目定义软件过程是一个已很好特征化的和已理解的软件过程，用软件标准、规程、工具和方法予以描述。通过剪裁组织标准软件过程以适合项目特定特征的方法来制定它。

是 否 不适用 不知道

1 组织是否遵循一文档化的用于制定和维护软件过程改进计划的规程？

评论：

2 是否全组织的人员都参加过程改进活动（例如成为开发软件过程改进的群组的成员）？

评论：

3 是否对组织标准软件过程和项目定义软件过程连续不断地进行改进？

评论：

4 组织是否遵循一书面的实施软件过程改进的方针？

评论：

5 是否对管理职员和技术职员都要求接受在软件过程改进方面的培训？

评论：

6 是否用测量量来确定软件过程改进活动的状态（例如，实施每项过程改进的效果与其规定的目标相比较广

评论：

7 高级管理者是否定期地参与评审软件过程改进的工作？

评论；