**第8章 软件工程管理**

**术语和缩写**

|  |  |  |
| --- | --- | --- |
| *PMBOK*® *Guide* | *Guide to the Project Management Body of Knowledge* | 项目管理知识体系指南 |
| SDLC | Software Development Life Cycle | 软件开发生命周期 |
| SEM | Software Engineering Management | 软件工程管理 |
| SQA | Software Quality Assurance | 软件质量保证 |
| *SWX* | *Software Extension to the PMBOK*® *Guide* | 项目管理知识体系指南扩展软件 |
| WBS | Work Breakdown Structure | 工作分解结构 |

**引言**

软件工程管理定义为应用管理活动—计划、协调、度量、监控、控制和报表[[1]](#footnote-1)—以保证软件产品及软件工程服务是高效率、高成效、符合相关方的利益。管理相关学科是所有知识领域中的要素，且它比其他知识领域相关性更强。同时，度量也是所有知识域中的一个重要方面，度量程序将在此进行讨论。

某种意义上，软件工程项目的管理，与其他复杂工程过程的管理类似。然而，软件产品和软件生命周期过程有其特定的方面，使得有效的管理变得复杂，这些方面包括：

* 客户往往不了解他们需要的是什么，或什么是可行的。
* 客户通常对软件工程中所蕴含的复杂度缺乏足够的认知，尤其是对于不断变化的需求所带来的影响。
* 对项目更强的理解以及变化的条件，将可能产生新需求或现有软件需求的变更。
* 变化的需求将使软件开发以迭代的过程进行，而非以一系列封闭式的任务进行。
* 软件工程往往融入了创造性和原则性两个方面，有时困难在于二者之间维持合理的平衡。
* 软件的创新性和复杂性通常很高。
* 所用到的技术，往往变化速度很快。

软件工程管理活动在三个层次上进行：组织和内部基础架构管理、项目管理、度量程序管理。本章中将详细描述后两个层次。当然，这并不意味着组织和基础架构管理问题并不重要。普遍认为软件工程经理应熟悉本章所谈及的项目管理以及软件度量方面的知识。同时他们还应具备一些目标领域的知识。类似地，在那些软件作为系统架构组成部分的复杂项目或程序中，如果经理能够了解软件过程的存在对项目管理、项目度量等工作所带来的改变，这同样会对项目开展有所帮助。

组织管理中的其他方面对软件工程具有巨大的影响（例如，组织中为软件工程项目的开展提供架构的政策和规程等）。这些政策和规程可能需要根据需求进行调整，从而保证软件开发和维护的有效性。此外，根据软件工程的特定需要，一系列特定的政策的贯彻落实，能够使得软件工程在组织层面更加有效。例如，常常有必要建立覆盖组织范围的、特定的软件工程任务过程或规程（如设计、搭建、估算、监视和报表）。这些策略对于有效的软件工程长期组织管理至关重要（例如，建立一个持续的过往项目性能度量基准，以实现不断改进）。

组织管理的另一个重要方面是人员管理在人员雇佣、培训、指导等方面上的政策与规程。这些政策与规程不仅仅存在于项目层面，还需考量个人职业发展以及组织在远期上的成功。软件工程人员的培训与管理可能产生独特的挑战（例如，在业界所运用的技术不断迅猛发展的背景下，如何保持人才的竞争力）。

组织中的信息沟通常常被忽略，但却非常重要。有效的沟通，在需要准确理解用户需要、软件需求、软件设计的情境下，决定了个人工作的效能。与此同时，组织需要建立项目名录管理，以便对正在开发、计划开发、正在使用的各类项目或程序，形成一个全局的认识。此外，软件复用也是保持并改善生产力和竞争力的关键因素。有效的复用依靠准确判断优势劣势的较强策略眼光。

除了理解软件项目对管理工作产生的独特影响之外，软件工程师还需对本章内容所讨论的管理问题，具有更广泛的知识，即使是在学习完本指南后刚刚参加工作的几年里也应如此。

组织文化与组织行为的性质，加上企业其他职能领域的管理，都对软件工程过程具有直接或间接的影响。

关于软件项目管理的延伸信息，可以在指南及其扩展软件中找到[1][2]。这两份内容分别包含十项知识领域：项目集成管理、项目范围管理、项目时间管理、项目成本管理、项目质量管理、项目人力资源管理、项目通信管理、项目风险管理、项目采购管理，以及项目利益相关方管理。每项主题都与软件工程管理知识域直接有关。

更多信息还可在本章内容的其他参考资料和拓展阅读中找到。

本章软件工程管理的知识领域，由图7.1软件项目管理过程中的前五项主题（启动与范围定义、软件项目计划制定、软件项目实施、评审与评价、关闭），以及第六项主题软件工程度量、第七项主题软件工程管理工具等内容构成。虽然项目管理与度量管理常常被分开讨论，它们两者也分别具有一些独特的属性，不过其紧密的联系使得本章将两者结合起来看待。

不幸的是，软件产业的一个普遍认知是软件产品时常延迟交付、预算超支、质量低下、功能不完善。度量信息支撑的管理——任何真正意义上的工程学科的基本原则（参见工程基础知识域中的度量章节）——可以帮助人们转变这种认知。本质上，缺乏定性或定量度量的管理，反映了条理原则的缺失；而没有管理的度量，则是目的或语境的缺失。有效的管理需要度量和经验的充分结合。

这里采用下列工作定义：

* 管理：一套为实现组织设定的战略目标而设立的过程和控制体系。
* 度量：分派给软件产品、过程、资源的数值量和标签，以及这些数值和标签衍生出的模型。这些模型可能是通过统计学或其他技术开发出来[3\*，c7，c8]。

本章中的软件工程项目管理部分，会广泛涉及软件工程度量部分的内容。

本章内容与指南中的其他章节也密切相关，阅读下面章节描述材料将有很大帮助：

* **工程基础知识域**描述了一些可以直接适用于软件工程度量章节的一般性概念。此外，在工程基础知识统计分析部分呈现的概念与技巧，也可直接应用于本章节的许多问题中。
* **软件需求知识域**描述了项目启动和范围定义阶段需要完成的一些活动。
* **软件配置管理知识域**探讨软件配置标识、配置控制、配置状态报告、配置审查，以及软件发布管理和交付、软件配置管理工具。
* **软件工程过程知识域**描述软件生命周期模型，以及过程与产品之间的关系。
* **软件质量知识域**强调质量是管理工作，以及其他众多软件工程活动的目标。
* **软件工程经济学知识域**讨论如何在商业情境下进行软件相关的决策。

**软件工程管理的主题分解结构**

由于大多数软件开发生命周期模型，都是相似的活动以不同方式执行的结果，因此主题分解按照活动来展开。该分解结构见图7.1。该图中最上层分解出的元素，是软件开发项目管理中常进行的工作，且不受特定项目所遵循的软件生命周期模型（参见软件工程过程章节中的软件生命周期模型部分）的影响。这一分解结构并不会推荐某种特定的生命周期模型。分解结构仅描述需要进行的工作，而不涉及何时、以何方式、以怎样的频率来执行不同的活动。七个主题包括：

* 启动和范围定义：关于软件工程项目启动阶段的一系列决定。
* 软件项目计划：从管理的角度，为软件工程项目的成功所作的准备工作。
* 软件项目实施：关于软件工程项目执行期间的常见软件工程管理活动。
* 评审和评价：关于保证技术、进度、成本、质量工程等满足要求的工作。
* 关闭：关于软件工程项目完成时的活动。
* 软件工程度量：关于软件工程组织中有效地开发和实施度量程序的工作。
* 软件工程管理工具：阐述了关于为软件工程项目的管理选用工具的问题。

**1.启动和范围定义**

这些活动的重点在于通过各种导引方法来有效地确定软件需求，并从不同的视点来评估项目的可行性。一旦可行性建立后，这一阶段余下的任务就是需求的描述以及需求更新、审阅流程的选择。

*1.1 需求的决定和协商*

[3\*，c3]

需求的决定与协商过程，为即将执行的任务确定了一个可见的边界（参见软件需求知识域）。相关活动包括需求引导、分析、描述，以及验证。过程中应根据各个相关方的情况选取应用相应的方法或技巧。最终决定了项目的范围，从而满足相关目标和约束条件。

*1.2 可行性分析*

[4\*，c4]

可行性分析的目标是建立项目目标的清晰描述，并评估不同的开发途径，从而确定在考虑到包括技术、资源、财务、社会/政治等因素下，所提出的方案是否为最佳选择。此过程中需要准备初始项目、产品范围陈述、项目交付物、项目周期约束条件，以及预期资源投入等。

资源投入包括数量充足的满足所需技能的人员、设备、设施，以及内外部支持。可行性分析通常需要通过合适的方法对所需投入的资源和成本进行大致的估计（参见2.3节，资源、进度、成本估计）。

*1.3 需求评审和修订过程*

缘于不可避免的各种变化，相关方需要对需求更新和范围变更的过程达成一致（例如，改变管理流程、周期迭代回顾）。显然，这表明范围和需求不应一成不变，而应随着项目的展开，在预先设定好的节点进行回顾和评定（例如，当任务积压，或在里程碑评审节点上）。如果变更得到接受，则须进行可追踪性分析和风险分析，来确认这些变更所带来的影响（参见2.5节，软件配置管理知识域中的风险管理，软件配置控制部分）。

基于项目进行过程中发生的各种变化，项目可以采用变更管理的方法，在项目进度节点或整个项目完成时，对成果进行评估（参见第五个主题，项目关闭）。

**2.软件项目计划制订**

软件项目计划的第一步是选取合适的软件开发生命周期模型，并根据项目范围、软件需求、风险评估进行定制化调整。同时，在制定计划时还要考量诸如应用领域的性质、功能与技术复杂度、软件质量需求等因素（参见软件质量知识域中的软件质量需求部分）。

任何软件开发生命周期中，初始项目计划阶段都包含了风险评估环节，同时，团队需要对风险概要进行讨论，并获得所有相关方的接受。在进行计划时，还应确定软件质量管理过程（参见软件质量知识域中的软件质量管理过程部分），并形成软件质量保证、验证、认证、评估、审查的流程和责任（参见软件质量知识域）。对于正在进行的项目计划审阅和修订的过程和责任，同样需要定义清楚并达成一致。

*2.1 制定过程计划*

[3\*，c3，c4，c5] [5\*，c1]

软件开发生命周期（SDLC）模型在预测性和适应性之间连续变化（参见软件工程过程知识域中的软件生命周期模型部分）。预测性模型的特点是其更加注重详细的软件需求、细致的项目计划，以及不同开发阶段迭代过程最小化的计划等。适应性模型则适应不断出现的软件需求和计划的迭代调整。一个高度预测性的模型线性地执行图7.1中所示的前五个过程，并仅当必须时对前一个阶段进行修订。适应性模型的特征时迭代式的开发周期。而对于兼具预测性和适应性的模型，按照实现计划的进度（预测性），或随着不断更新的开发周期（适应性），在功能方面不断增长。

广为人知的软件开发生命周期模型包括瀑布模型、增长式模型、螺旋模型，以及各类敏捷软件开发模式[2] [3\*，c2]。

在制定计划时需要选取相关的方法（参见软件工程模型与方法知识域）和工具。在项目过程中采用的自动化工具也需要进行规划和获取。工具的用途包括项目进度制定、软件需求、软件设计、软件搭建、软件维护、软件配置管理、软件工程过程、软件质量等。而其中很多工具的选择，需主要基于其他章节中讨论的技术考量，有些则与本章中讨论的管理方面的因素密切相关。

*2.2 确定可交付制品*

[3\*，c4，c5，c6]

项目中每项活动的工作对象产品（例如软件架构设计文档、检查报告、接受测试的软件等），需要进行识别定义并具体描述其特征。复用以往项目软件组件或应用现有软件产品的可能性应得到评估。软件的采购以及借助第三方开发交付物等行为，需要实现计划并对供应商进行遴选（参见3.2节，软件获取和供应商合约管理）。

*2.3 工作量、进度和成本估算*

[3\*，c6]

整个项目或局部任务预期所需工作量的范围，可以基于历史规模及工作量数据（如果有相应记录的话），或采用诸如专家评判与类比等相关方法，以标定预估模型进行计算。任务的依赖关系、并行或串行任务的机会等，可以通过诸如甘特图等方法进行识别和制定。对于预测性生命周期的项目来说，项目计划阶段将制定任务的预期进度时间表，包括任务开始时间、持续时间、结束时间等。对于适应性生命周期的项目，整体工作量及进度表的预估通常根据项目需求的初始理解来完成。另一种方式下，总体工作量及进度表的约束条件，详细列出并用作决定迭代周期的数目以及每个周期所分配的工作量和资源。

资源需求（例如人力资源及工具等），可以换算成预计的成本。对工作量、进度表、成本的初始估计，是需要各个利益相关方不断迭代进行讨论磋商和修订来制定的，直到各方项目在完成所能获得的资源和时间上达成共识。

*2.4 资源分配*

[3\*，c5，c10，c11]

设备、设施、人员要根据相应的任务进行分配，包括完成项目各个任务以及整体任务的责任的分派等。可以利用一个矩阵来描述项目中的种种任务，应该向谁协调、问责、咨询、通知。

资源的分配，需要根据并受限于不同资源的可用性、最佳用途、人员问题等（例如，个人生产力、团队生产力、团队协作、团队架构等）。

*2.5 风险管理*

[3\*，c9] [5\*，c5]

风险和不确定性是相关但不同的两个概念。不确定性来自信息的缺乏，而风险则描述了一个事件产生负面影响的可能性及其对项目负面影响的特征。风险常常是由不确定性引发的，却也能够带来机会，即一些事件还可能带来一些正面影响。

风险管理的主要内容包括风险因素识别、每个因素可能性及潜在影响的分析，以及当一个因素可能带来问题时，如何降低可能性、消减影响等风险消弭策略的开发。各类风险评估方法（例如专家评判、历史数据、决策树、过程仿真等）可以用于识别和评价风险因素。

项目废弃条件，也可以在风险评估环节，通过与所有利益相关方探讨后来决定。软件具有独特的风险因素，例如软件工程师们趋于向产品中加入并不需要的特色功能，或是由于软件非实物特质所带来的风险。这些因素会影响到软件工程的风险管理。另外需要格外注意的是，软件质量需求相关的风险管理，例如安全与保障等问题（参见软件质量知识域）。风险管理不仅应在项目起始阶段进行，还应在整个项目生命周期中阶段性、周期性的进行。

*2.6 质量管理*

[3\*，c4] [4\*，c24]

软件工程和相关产品的软件质量需求，需要从定性和定量两个方面来识别制定。质量度量标准应根据相关方的需求和预期，针对每项质量需求设定。质量计划阶段，还要定义好正在进行中的软件质量保障（Software Quality Assurance，SQA）、项目开发过程中的质量改善，以及软件产品交付物的认证与验证等工作的有关流程（例如，技术审核、检查、已完成项目的演示等；参见软件质量知识域）。

*2.7 计划管理*

[3\*，c4]

软件项目中，常常会需要进行变更，因此需要对计划进行管理。对项目规划的管理也需要进行计划。软件开发工作所选用的计划和流程需要进行系统性的监视、评审、汇报，并在适当条件下进行修改。支撑流程的相关计划（例如，文档、软件配置管理、问题解决方案等）也需要进行管理。项目的汇报、监视和控制应满足项目所采用的软件开发生命周期模型，以及项目的现实情况。同时相关计划还需考虑到管理项目所需要的各类用品。

**3.软件项目实施**

软件工程实施阶段，相应项目计划将得到执行，计划中的相关流程也付诸实施。整个项目过程中，应该关注遵循软件开发生命周期相关过程进行执行，同时坚信遵循计划可以实现利益相关方的需求成功地得到满足，并完成项目目标。项目实施的基本工作包括持续的监视、控制、汇报等管理活动。

*3.1 计划的执行*

[4\*，c2]

项目活动需要按照项目计划和支撑性计划来开展。期间对资源（例如人力、技术、资金等）进行使用，并产出可交付物（例如软件设计、软件代码、软件测试案例等）。

*3.2 软件获取与供应商合同管理*

[3\*，c3，c4]

软件获取与供应商合同管理主要讨论两类问题：与软件开发机构的客户之间的合约，他们从软件开发机构获取可交付的产品；以及与软件开发机构的供应商之间的合约，他们向软件开发机构提供产品或服务。

这之间涉及到为合约选取合适的条件，例如固定价格、时间与材料、成本加固定费用，或成本加绩效。与客户或供应商的协议通常需要定义工作范围及交付物等，并包含各类条款，诸如延迟交付或未交付违约的惩罚措施，以及定义供应商提交物以及接收方支付物，加之双方交接及归属的知识产权协议。对于供应商开发等软件（包括对内及对外的），协议往往会定义好满足接收方顺利接受的软件质量的需求。

在签订协议后，需要对项目执行进行管理，以符合协议条款（本章更多信息请参见扩展软件第12章软件采购管理部分内容[2]）。

*3.3 度量过程的实施*

[3\*，c7]

在软件项目中实施的度量过程应保证收集到相关的、有用的数据（参见6.2节计划度量过程，及6.3节执行度量过程）。

*3.4 监控过程*

[3\*，c8]

项目执行是否遵循计划应持续按照预先设定的间隔进行，并对每项任务的过程产出和完成情况进行评价。可交付物需要根据其需求特征（例如通过检验或通过功能演示）来进行评价。需要对工作量消耗、进度执行情况、已产生的成本等进行分析，并检验资源使用情况。应对项目风险情况（参见2.5节风险管理）进行回顾，并评价软件质量需求的合规情况（参见软件质量知识域中的软件质量需求部分）。

对度量数据进行分析（参见工程基础知识域中的统计分析部分）。应针对预期产出与实际产出之间的偏差进行差异分析。其中包括成本超支、进度落后、或其他类似指标。应识别异常值，并对质量及其他度量数据进行分析（如缺陷分析；参见软件质量知识域中的软件质量度量部分）。项目所面临的风险需要再度进行计算（参见2.5节风险管理）。这些活动通过判断各项指标是否超出，来帮助发现问题与异常。当超标情况出现时，应根据需要报告结果。

*3.5 控制过程*

[3\*，c7，c8]

项目监视活动的产出，为下一步的决策制定提供了依据。当条件适宜，且风险可能性及风险因素影响得到充分认识后，可以对项目进行变更。变更可通过修正措施的形式执行（例如，对软件组件进行重新测试）；也会涉及到额外的措施（例如，决定通过原型产品来辅助校验软件需求；参见软件需求知识域中的原型产品部分）；和/或导致项目计划及其他项目文档（如项目需求规格）的修订，从而适应未能预知的事件及其影响。

某些情况下，对项目进行管控的结果是项目废弃。任何情况下，软件规格控制和软件规格管理规程应遵循所有相关方（参见软件规格管理知识域），决策需与所有相关方沟通并进行记录，必要时对计划进行回顾和修订，相关数据需进行记录（参见6.3节度量过程实施）。

*3.6 制作报表*

[3\*，c11]

在定义好并得到共同认可的时间点，当前进度需要在组织内部（例如项目指导委员会）以及对外部利益相关方（例如客户或用户）进行汇报。报表的重点应根据听众关注点进行调整，对于组织内部，更多汇报详细的开发进度状态，对于外部则在目标听众所需要的信息。

**4.评审和评价**

在根据需要事先定义好的时间点，应对达到设定目标及满足利益相关方需求（用户及客户）的整体进度进行评估。相似地，针对软件过程有效性、参与人员、使用工具及方法等方面的评价，也应定期进行或根据条件来执行。

*4.1 决定需求的满意度*

[4\*，c8]

由于软件工程经理的首要目标既是达到利益相关方对项目的满意，因此达到这一目标的过程，应定期进行评价。 过程评价应基于项目主要里程碑的达成情况（例如完成软件设计架构、软件技术评审），或基于产品增长的迭代开发周期的完成情况。偏离软件需求的情况应及时得到发现并采取适当措施。

在上述控制过程的活动中（参见3.5节控制过程），应遵循软件配置控制及软件配置管理规程（参见软件配置管理知识域），决策需与全部相关方沟通并存档，需要时对计划进行评审和修订，相关数据进行记录（参见6.3节，开展度量过程）。

*4.2 绩效的评审和评价*

[3\*，c8，c10]

对项目人员的绩效定期进行评审，能够帮助洞察按计划和过程执行的可能性，以及可能存在的难点（例如团队成员之间的冲突）。所应用的各类方法、工具、技巧，也应进行有效性和适用性的评估；项目所采用的过程，也应进行系统性和周期性的评价，考察其在项目情境下的相关性、实用性和功效性。合适的情况下，应执行并管理变更。

**5.关闭**

当全部计划和过程皆已执行并完成时，整个项目、项目主要阶段，或是一个迭代开发周期即可关闭。项目、阶段、周期所相应的成功标准应进行评估。当执行关闭时，可执行存档、回顾、过程改进等活动。

*5.1 确定关闭*

[1，s3.7，s4.6]

当项目、阶段、迭代周期规定任务完成，确认满足相应完成标准时，项目可以进入关闭程序。软件需求可以确认为是否满足，并决定目标的满足程度。关闭程序应让相关方参与，且相关方接受的结果做好相应文件存档。此外任何问题也应进行存档。

*5.2 关闭活动*

[2，s3.7，s4.8]

项目确认可以关闭后，项目材料的归档，应按照相关方达成共识的方法、位置、期限来完成——这可能包括敏感信息、软件、副本存储介质的销毁等。组织度量的数据库应由相关项目数据进行更新。项目、阶段、迭代周期的回顾分析中，应针对遇到的议题、问题、风险、机会等进行（参见主题4，回顾与评估）。项目应得出相关经验，并加入组织学习和改进的活动之中。

**6.软件工程度量**

度量的重要性及其对于良好管理和工程实践所发挥的作用，已被广泛认同（参见工程基础知识域中度量部分）。有效的度量，已经成为组织成熟度的基石。度量可应用于组织、项目、过程，以及工作产品。本节将重点关注度量活动在项目、过程，以及工作产品层面的应用。

本节按照IEEE 15939:2008[6]标准编写，该标准描述了实施软件度量过程所需的活动与任务的定义的过程。该标准还包含了度量信息模型。

*6.1 建立和维持度量工作*

[7\*，c1，c2]2

* 度量的需求：每项度量工作都应由组织目标指导，并由一套组织或项目建立的度量需求来驱动（例如一个组织目标是“率先向市场投放新产品”）。
* 度量的范围：定义每项度量需求所规定的组织范围。这一范围可能是单项功能、单个项目、单个网站，或整个企业。度量工作的时间范围也应纳入考虑范围，因为有些度量变量需要时间序列数据，例如，测量估计模型（参见2.3节，工作量、进度、成本估计）。
* 度量工作的委派：委派工作需要正式执行、沟通，并由各类资源支撑（参见下一条目）。
* 度量的资源：组织对度量工作的投入程度，是项目成功的关键因素，这体现在度量过程的资源分配情况上。资源指派包括不同度量过程任务（例如分析师和管理员）的责任分配。同时，还应为度量过程的执行，提供合适的资金、培训、工具、支持等。

*6.2 度量过程的计划*

[7\*，c1，c2]

* 定义组织的单位：组织单位是度量工作执行的环境，因此组织背景应明确定义，包括组织对度量过程的约束条件等。组织单元可从组织过程、应用范围、技术、组织界面、组织架构等方面进行定义。
* 识别信息需求。信息需求基于组织单位的目标、约束条件、风险、问题等进行制定。其可从业务、组织、规章，和/或产品目标演变而来。这些需要应得到定义并排出优先级。随后，需制定的目标子集，即可由利益相关方进行选取、记录、交流、评审。
* 选取指标：先选择出与信息需求具有明确联系的候选指标。指标的选取需要根据信息需求的优先级及其他标准，诸如采集成本、采集过程中对过程的中断程度，获取精确一致数据的难度，分析和汇报的难度等。由于内部质量特征（参见软件质量知识域中的模型与质量特征部分）往往不包含于合约规定的软件需求之中，因此很重要的一个工作是考虑度量软件的内部质量，从而为可能影响外部利益相关方的潜在问题提供早期的指示。
* 定义数据采集、分析、汇报的流程：包括数据的采集流程、进度、存储、验证、分析、汇报、配置管理等。
* 选取评价信息产品的标准：评价标准受到组织单位技术与业务目标的影响。信息产品包括与在制产品相关的，以及与在用的项目管理与度量过程相关的产品。

\_\_\_\_\_\_\_\_\_\_\_\_本次新版翻译至此\_\_\_\_\_\_\_\_\_\_\_\_\_

定义数据的收集、分析和报表制作流程：这包括数据的收集流程和进度、存储、验证、分析、报表制作和配置管理[ISO15939-02:5.2.4].

定义评价信息产品的准则：评价的准则受组织单元的技术和业务目标的影响，信息产品包括与正在生产的产品相关的，以及与用来管理和度量项目的过程相关的[ISO15939-02: 5.2.5 and Appendices D, E]。

评审、批准和为度量任务提供资源：（1）必须评审度量计划，并由适当的干系人批准，这包括所有的数据收集流程、存储、分析和报表制作流程、评价准则、进度和责任。评审这些制品的准则应该在组织单元或更高的层次上建立，并用作评审的基础。这些准则应该考虑以前的经验、资源的可用性、当建议对目前的实践进行变更时对项目的破坏程度。批准表示了对度量过程的承诺[ISO15939-02: 5.2.6.1 and Appendix F]。（2）应该让资源对于实现列入计划并被批准的度量任务是可用的，资源可用性可能在变更被控制在广泛部署之前就准备好。应该考虑为成功部署新流程或度量所必须的资源[ISO15939-02: 5.2.6.2]。

获取和部署支持技术：这包括评价可用的支持技术、选择最适当的技术、获取这些技术和部署这些技术[ISO15939-02:5.2.7].

*6.3 完成度量过程*

将度量流程与相关的过程集成：诸如数据收集等度量流程，必须集成到它们度量的过程中，这可能涉及改变目前的过程以包容数据收集或产生活动，也可能涉及分析目前的过程，将对雇员的影响的附加的工作量和评价最小化，以保证度量流程被接受。需要考虑士气问题和其它人类因素问题。此外，必须与提供数据的人员交流度量流程，可能需要提供培训，还要提供支持。通常，数据分析和报表制作流程必须用类似的方式集成到组织和/或项目过程[ISO15939-02: 5.3.1]。

收集数据：必须收集、验证和存储数据[ISO15939-02 :5.3.2]。 分析数据和开发信息产品：作为分析过程的一部分，使用适合于数据本质和信息要求的某种严密程度，数据被聚集、变换或重新编码。分析的结果通常是一些指标，如图形、数值或其它必须解释的指标，得到要提交给干系人的初步结论。必须使用组织定义的过程（可能是正式或非正式的），来评审结果和结论。数据提供者和度量用户应参与数据评审，以保证数据有意义和精确，并能导致合理的行动[ISO15939-02: 5.3.3 and Appendix G]。.

交流结果：信息产品必须记录到文档中，并与用户和干系人交流[ISO15939-02: 5.3.4]。

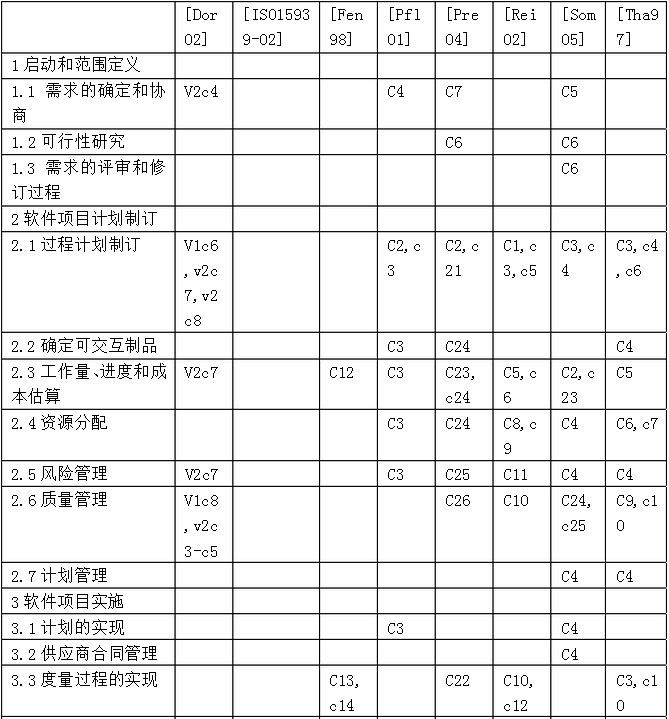
*6.4 评价度量*

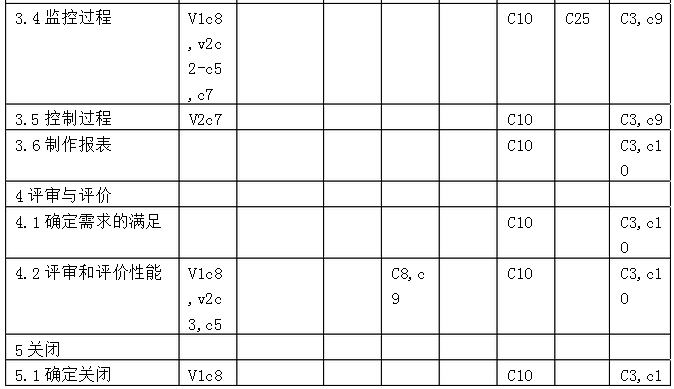
评价信息产品：根据规定的评价准则来评价信息产品，确定信息产品的优缺点。这可能由内部过程或外部审计来完成，并要包括来自度量用户的反馈。将经验和教训记录到适当的数据库中[ISO15939-02: 5.4.1 和Appendix D]。

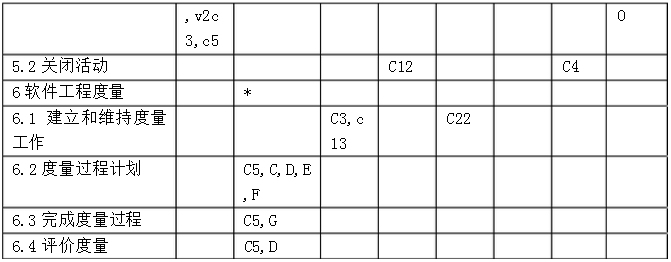
评价度量过程：根据规定的评价准则来评价度量过程，确定过程的优缺点。这可能由内部过程或外部审计来完成，并要包括来自度量用户的反馈。将经验和教训记录到适当的数据库中 [ISO15939-02: 5.4.1和Appendix D]。

标识潜在的改进：这类改进可能是指标格式的改变、度量单位的改变或重新分类。确定潜在改进的成本和效益，选择适当的改进行动。与度量过程所有者和干系人进行交流，以便评审和批准。如果分析没有标识出改进，也要指出缺乏潜在的改进[ISO15939-02: 5.4.2]。

**主题与参考文献矩阵**







**软件工程管理的推荐参考文献**

 [Dor02] M. Dorfman and R. H. Thayer, Eds., "Software Engineering." (Vol. 1 & vol. 2), IEEE Computer Society Press, 2002, Vol. 1, Chap. 6, 8, Vol. 2, Chap. 3, 4, 5, 7, 8.

[Fen98] N. E. Fenton and S. L. Pfleeger, "Software Metrics: A Rigorous & Practical Approach," Second ed: International Thomson Computer Press, 1998, Chap. 1-14.

[ISO15939-02] ISO/IEC 15939:2002, Software Engineering-Software Measurement Process: ISO and IEC, 2002.

[Pfl01] S. L. Pfleeger, "Software Engineering: Theory and Practice," Second ed: Prentice-Hall, 2001, Chap 2-4,8,9,12, 13.

[Pre04] R. S. Pressman, "Software Engineering: A Practitioner's Approach," Sixth ed: McGraw-Hill, 2004, Chap. 2, 6, 7, 22-26.

[Rei02] D. J. Reifer, Ed., "Software Management." IEEE Computer Society, 2002, Chap. 1-6, 7-12, 13.

[Som05] I. Sommerville, "Software Engineering," Seventh ed: Addison-Wesley, 2005, Chap. 3-6, 23-25.

[Tha97] R. H. Thayer, Ed., "Software Engineering Project Management." IEEE Computer Society, 1997, Chap. 1-10

**附录A 深入阅读的参考文献**

(Adl99) T. R. Adler, J. G. Leonard and R. K. Nordgren, "Improving Risk Management: Moving from Risk Elimination to Risk Avoidance," Information and Software Technology, vol. 41, 29-34, 1999

(Bai98) R. Baines, "Across Disciplines: Risk, Design, Method, Process, and Tools," IEEE Software, 61-64, July/August, 1998

(Bin97) R. V. Binder, "Can a Manufacturing Quality Model Work for Software?," IEEE Software, 101-102,105, September/October, 1997

(Boe97) B. W. Boehm and T. DeMarco, "Software Risk Management," IEEE Software, 17-19, May/June, 1997

(Bri96) L. C. Briand, S. Morasca and V. R. Basili, "Property-Based Software Engineering Measurement," IEEE Transactions on Software Engineering, vol. 22, iss. 1, 68-86, 1996

(Bri96a) L. Briand, K. E. Emam and S. Morasca, "On the Application of Measurement Theory in Software Engineering," Empirical Software Engineering, vol. 1, 61-88, 1996

(Bri97) L. C. Briand, S. Morasca and V. R. Basili, "Response to: Comments on 'Property-based Software Engineering Measurement: Refining the Addivity

Properties'," IEEE Transactions on Software Engineering, vol. 23, iss. 3, 196-197, 1997 (Bro87) F. P. J. Brooks, "No Silver Bullet: Essence and Accidents of Software Engineering," Computer, 10-19, Apr., 1987

(Cap96) J. Capers, Applied Software Measurement: Assuring Productivity and Quality, Second ed: McGraw-Hill, Inc., 1996.

(Car97) M. J. Carr, "Risk Management May Not Be For Everyone," IEEE Software, 21-24, May/June, 1997

(Cha96) R. N. Charette, "Large-Scale Project Management is Risk Management," IEEE Software, 110-117, July, 1996

(Cha97) R. N. Charette, K. M. Adams and M. B. White, "Managing Risk in Software Maintenance," IEEE Software, 43-50, May/June, 1997

(Col96) B. Collier, T. DeMarco and P. Fearey, "A Defined Process for Project Postmortem Review," IEEE Software, 65-72, July, 1996

(Con97) E. H. Conrow and P. S. Shishido, "Implementing Risk Management on Software Intensive Projects," IEEE Software, 83-89, May/June, 1997

(Dav98) A. M. Davis, "Predictions and Farewells," IEEE Software, 6-9, July/August, 1998

(Dem87) T. DeMarco and T. Lister, Peopleware: Productive Projects and Teams: Dorset House Publishing, 1987.

(Dem96) T. DeMarco and A. Miller, "Managing Large Software Projects," IEEE Software, 24-27, July, 1996

(Fav98) J. Favaro and S. L. Pfleeger, "Making Software Development Investment Decisions," ACM SIGSoft Software Engineering Notes, vol. 23, iss. 5, 69-74, 1998

(Fay96) M. E. Fayad and M. Cline, "Managing Object-Oriented Software Development," Computer, 26-31, September, 1996

(Fen98) N. E. Fenton and S. L. Pfleeger, Software Metrics: A Rigorous & Practical Approach, Second ed: International Thomson Computer Press, 1998.

(Fle99) R. Fleming, "A Fresh Perspective on Old Problems," IEEE Software, 106-113, January/February, 1999

(Fug98) A. Fuggetta, L. Lavazza, S. Morasca, S. Cinti, G. Oldano and E. Orazi, "Applying GQM in an Industrial Software Factory," ACM Transactions on Software Engineering and Methodology, vol. 7, iss. 4, 411-448, 1998

(Gar97) P. R. Garvey, D. J. Phair and J. A. Wilson, "An Information Architecture for Risk Assessment and Management," IEEE Software, 25-34, May/June, 1997

(Gem97) A. Gemmer, "Risk Management: Moving beyond Process," Computer, 33-43, May, 1997

(Gla97) R. L. Glass, "The Ups and Downs of Programmer Stress," Communications of the ACM, vol. 40, iss. 4, 17-19, 1997

(Gla98) R. L. Glass, "Short-Term and Long-Term Remedies for Runaway Projects," Communications of the ACM, vol. 41, iss. 7, 13-15, 1998

(Gla98a) R. L. Glass, "How Not to Prepare for a Consulting Assignment, and Other Ugly Consultancy Truths," Communications of the ACM, vol. 41, iss. 12, 11-13, 1998

(Gla99) R. L. Glass, "The Realities of Software Technology Payoffs," Communications of the ACM, vol. 42, iss. 2, 74-79, 1999

(Gra99) R. Grable, J. Jernigan, C. Pogue and D. Divis, "Metrics for Small Projects: Experiences at the SED," IEEE Software, 21-29, March/April, 1999

 (Gra87) R. B. Grady and D. L. Caswell, Software Metrics: Establishing A Company-Wide Program. Englewood Cliffs NJ, USA: Prentice-Hall, 1987.

(Hal97) T. Hall and N. Fenton, "Implementing Effective Software Metrics Programs," IEEE Software, 55-64, Mar/Apr, 1997

(Hen99) S. M. Henry and K. T. Stevens, "Using Belbin's Leadership Role to Improve Team Effectiveness: An Empirical Investigation," Journal of Systems and Software, vol. 44, 241-250, 1999

(Hoh99) L. Hohmann, "Coaching the Rookie Manager," IEEE Software, 16-19, January/February, 1999

(Hsi96) P. Hsia, "Making Software Development Visible," IEEE Software, 23-26, March, 1996

(Hum97) W. S. Humphrey, Managing Technical People: Innovation, Teamwork, and the Software Process: Addison-Wesley, 1997.

(IEEE12207.0-96) IEEE/EIA 12207.0-1996//ISO/IEC12207:1995, Industry Implementation of Int. Std. ISO/IEC 12207:95, Standard for Information Technology-Software Life Cycle Processes, vol. IEEE, 1996.

(Jac98) M. Jackman, "Homeopathic Remedies for Team Toxicity," IEEE Software, 43-45, July/August, 1998

(Kan97) K. Kansala, "Integrating Risk Assessment with Cost Estimation," IEEE Software, 61-67, May/June, 1997

(Kar97) J. Karlsson and K. Ryan, "A Cost-Value Aproach for Prioritizing Requirements," IEEE Software, 87-74, September/October, 1997

 (Kar96) D. W. Karolak, "Software Engineering Risk Management," IEEE Computer Society, 1996

(Kau99) K. Kautz, "Making Sense of Measurement for Small Organizations," IEEE Software, 14-20, March/April, 1999

(Kei98) M. Keil, P. E. Cule, K. Lyytinen and R. C. Schmict, "A Framework for Identifying Software Project Risks," Communications of the ACM, vol. 41, iss. 11, 76-83, 1998

(Ker99) B. Kernighan and R. Pike, "Finding Performance Improvements," IEEE Software, 61-65, March/April, 1999

(Kit97) B. Kitchenham and S. Linkman, "Estimates, Uncertainty, and Risk," IEEE Software, 69-74, May/June, 1997

(Lat98) F. v. Latum, R. v. Solingen, M. Oivo, B. Hoisl, D.Rombach and G. Ruhe, "Adopting GQM-Based Measurement in an Industrial Environment," IEEE Software, 78-86, January-February, 1998

(Leu96) H. K. N. Leung, "A Risk Index for Software Producers," Software Maintenance:Research and Practice, vol. 8, 281-294, 1996

 (Lis97) T. Lister, "Risk Management is Project Management for Adults," IEEE Software, 20-22, May/June, 1997

(Mac96) K. Mackey, "Why Bad Things Happen to Good Projects," IEEE Software, 27-32, May, 1996

(Mac98) K. Mackey, "Beyond Dilbert: Creating Cultures that Work," IEEE Software, 48-49, January/February, 1998 (Mad97) R. J. Madachy, "Heuristic Risk Assessment Using Cost Factors," IEEE Software, 51-59, May/June, 1997

(McC96) S. C. McConell, Rapid Development: Taming Wild Software Schedules: Microsoft Press, 1996.

(McC97) S. C. McConell, Software Project Survival Guide: Microsoft Press, 1997.

(McC99) S. C. McConell, "Software Engineering Principles," IEEE Software, 6-8, March/April, 1999

(Moy97) T. Moynihan, "How Experienced Project Managers Assess Risk," IEEE Software, 35-41, May/June, 1997

(Ncs98) P. Ncsi, "Managing OO Projects Better," IEEE Software, 50-60, July/August, 1998

(Nol99) A. J. Nolan, "Learning From Success," IEEE Software, 97-105, January/February, 1999

(Off97) R. J. Offen and R. Jeffery, "Establishing Software Measurement Programs," IEEE Software, 45-53, Mar/Apr, 1997

(Par96) K. V. C. Parris, "Implementing Accountability," IEEE Software, 83-93, July, 1996 (Pfl97) S. L. Pfleeger, "Assessing Measurement (Guest Editor's Introduction)," IEEE Software, 25-26, March/April, 1997

(Pfl97a) S. L. Pfleeger, R. Jeffery, B. Curtis and B. Kitchenham, "Status Report on Software Measurement," IEEE Software, 33-43, March/April, 1997

(Put97) L. H. Putman and W. Myers, "Industrial Strength Software - Effective Management Using Measurement," Los Alamitos, CA, 1997

(Rob99) P. N. Robillard, "The Role of Knowledge in Software Development," Communications of the ACM, vol. 42, iss. 1, 87-92, 1999

(Rod97) A. G. Rodrigues and T. M. Williams, "System Dynamics in Software Project Management: Towards the Development of a Formal Integrated Framework," European Journal of Information Systems, vol. 6, 51-66, 1997

(Rop97) J. Ropponen and K. Lyytinen, "Can Software Risk Management Improve System Development: An Exploratory Study," European Journal of Information Systems, vol. 6, 41-50, 1997

(Sch99) C. Schmidt, P. Dart, L. Johnston, L. Sterling and P. Thorne, "Disincentives for Communicating Risk: A Risk Paradox," Information and Software Technology, vol. 41, 403-411, 1999

(Sco92) R. L. v. Scoy, "Software Development Risk: Opportunity, Not Problem," Software Engineering Institute, Carnegie Mellon University CMU/SEI-92-TR-30, 1992 (Sla98) S. A. Slaughter, D. E. Harter and M. S. Krishnan, "Evaluating the Cost of Software Quality," Communications of the ACM, vol. 41, iss. 8, 67-73, 1998

(Sol98) R. v. Solingen, R. Berghout and F. v. Latum, "Interrupts: just a minute never is," IEEE Software, 97-103, September/October, 1998

(Whi95) N. Whitten, Managing Software Development Projects: Formulas for Success: Wiley, 1995.

(Wil99) B. Wiley, Essential System Requirements: A Practical Guide to Event-Driven Methods: Addison-Wesley, 1999.

(Zel98) M. V. Zelkowitz and D. R. Wallace, "Experimental Models for Validating Technology," Computer, vol. 31, iss. 5, 23-31, 1998

**附录B 有关的标准**

(IEEE610.12-90) IEEE Std 610.12-1990 (R2002), IEEE Standard Glossary of Software Engineering Terminology: IEEE, 1990.

(IEEE12207.0-96) IEEE/EIA 12207.0-1996//ISO/IEC12207:1995, Industry

Implementation of Int. Std. ISO/IEC 12207:95, Standard for Information Technology-Software Life Cycle Processes, vol. IEEE, 1996.

(ISO15939-02) ISO/IEC 15939:2002, Software Engineering-Software Measurement Process: ISO and IEC, 2002.

(PMI00) Project Management Institute Standards Committee, A guide to the project management body of knowledge (PMBOK): Project Management Institute, 2000

///////////////////////////////////////////////////////////////

1. 项目管理知识体系指南及其扩展软件中，使用启动、计划、执行、监视、控制、关闭等术语描述过程组。 [↑](#footnote-ref-1)