

软件质量管理（下）

参考：

- Software Engineering - A Practitioner's Approach (软件工程——实践者的研究方法) 第一、第二部分
- ISO 9126 (软件产品质量) 和 ISO 14598 (软件产品评价)
- ISO 9000:2000
- CMM(SEI提出的CMM 的叁篇重要文献。“软件能力成熟度模型1.1版”；“能力成熟度模型的关键实践1.1 版”；“软件过程成熟度提问单”)
- ISO 15504 (SPICE, 软件过程改进和能力确定)
- ISO 12207 (软件生存期过程)

软件质量管理

- 质量管理的基本概念（质量管理体系评价）
- 软件质量管理的主要活动
- 认证和评估简介
- 计算机软件文档编制规范GB/T 8567-2006
- 软件质量管理实践
- 建立软件测试管理体系



质量管理的基本概念

- **体系(系统) system**

相互关联或相互作用的一组要素

- **组织 organization**

职责、权限和相互关系得到安排的一组人员及设施。

- **管理体系 management system**

建立方针和目标并实现这些目标的体系

注：一个组织的管理体系可包括若干个不同的管理体系，如质量管理体系、财务管理体系或环境管理体系。



质量管理的基本概念

○ 质量管理体系 quality management system

指导和控制组织的关于质量的管理体系。

为加深理解，看GB/T 6583(ISO 8402)中对质量管理体系的定义：
为实施质量管理所需的组织结构，程序、过程和资源。

- n 组织结构：组织为行使其职能按某种方式建立的职责、权限及其相互关系。
- n 程序：为进行某项活动所规定的途径。
- n 过程：将输入转化为输出的一组彼此相关的资源和活动。
- n 资源：可包括人员、资金、设施、设备、技术和方法。

质量管理的基本概念

对于质量管理体系：

- 1 质量体系的内容应以满足质量目标的需要为准。
- 2 一个组织的质量体系主要是为满足该组织内部管理的需要而设计的。它比特定顾客的要求要广泛。顾客仅仅评价质量体系中的有关部分。
- 3 为了合同或强制性质量评价的目的，可要求对已确定的质量体系要求的实施进行证实。



质量管理的基本概念

○ 质量方针 quality policy

由组织的最高管理者正式发布的该组织总的质量宗旨和质量方向。

注1：通常质量方针与组织的总方针相一致并为制定质量目标提供框架。

○ 质量目标 quality objective

关于质量的所追求的目的

注1：质量目标通常建立在组织的质量方针基础上。

注2：通常对组织的各相关职能和层次分别规定质量目标。

质量管理的基本概念

○ 质量策划 quality planning

质量管理的一部分，致力于制定质量目标并规定必要作业过程和相
关资源以实现质量目标

注：编制质量计划可以是质量策划的一部分。

GB/T6583-ISO 8402中的定义是：

确定质量以及采用质量体系要素的目标和要求的活动。

质量策划包括：

- a) 产品策划：对质量特性进行识别、分类和比较，并建立其目标、质量要求和约束条件。
 - b) 管理和作业策划：为实施质量体系进行准备，包括组织和安排。
 - c) 编制质量计划和作出质量改进的规定。
-

质量管理的基本概念

○ 质量控制 **quality control**

质量管理的一部分，致力于满足质量要求。

GB/T6583-ISO 8402中的定义是：

为达到质量要求所采取的作业技术和活动。

- 1 质量控制包括作业技术和活动，其目的在于监视过程并排除质量环中所有阶段中导致不满意的原因，以取得经济效益。
- 2 质量控制和质量保证的某些活动是互相关联的。

质量管理的基本概念

○ 质量保证 quality assurance

质量管理的一部分，致力于提供能满足质量要求的信任。

GB/T6583-ISO 8402中的定义是：

为了提供足够的信任表明实体能够满足质量要求，而在质量体系中实施并根据需要进行证实的全部有计划和有系统的活动。

1 质量保证有内部和外部两种目的。

a) 内部质量保证：在组织内部，质量保证向管理者提供信任。

b) 外部质量保证：在合同或其他情况下，质量保证向顾客或他方提供信任。

2 质量控制和质量保证的某些活动是相互关联的。

3 只有质量要求全面反映了用户的要求，质量保证才能提供足够的信任。

质量管理的基本概念

○ 质量改进 quality improvement

质量管理的一部分，致力于增强满足质量要求的能力

注：要求可以是有关任何方面的，如有效性、效率或可追溯性。

n 有效性 effectiveness

完成策划的活动并达到策划的结果的程度。

n 效率 efficiency

得到的结果与所使用的资源之间的关系。

n 可追溯性 traceability

追溯所考虑对象的历史、应用情况或所处场所的能力

注：当考虑产品时，可追溯性可涉及到：

--原材料和零部件的来源； --加工过程的历史；

--产品交付后的分布和场所。

质量管理的基本概念

○ 质量管理 quality management

指导和控制组织的关于质量的相互协调的活动

注：关于质量的指导和控制活动通常包括制定质量方针和质量目标以及质量策划、质量控制、质量保证和质量改进。

GB/T6583-ISO 8402中的定义是：

确定质量方针、目标和职责并在质量体系中通过诸如质量策划、质量控制、质量保证和质量改进使其实施的全部管理职能的所有活动。

- 1 质量管理是各级管理者的职责，但必须由最高管理者领导。质量管理的实施涉及到组织中的所有成员。
- 2 在质量管理中要考虑到经济性因素。

质量管理的基本概念

○ 质量管理的3个阶段

质量管理从出现到现在，大体经历了3个阶段

n 产品质量检验阶段

是在成品中挑出废品，以保证出厂产品质量。但这种事后检验把关，无法在生产过程中起到预防、控制的作用。

n 统计质量管理阶段

运用数理统计原理，在发现有废品生产的先兆时就进行分析改进，从而预防废品的产生。

n 全面质量管理阶段

执行质量职能是公司全体人员的责任。把质量问题作为一个有机整体加以综合分析研究，实施全员、全过程、全企业的管理。

质量管理的基本概念

○ 全面质量管理 total quality management

一个组织以质量为中心，以全员参与为基础，目的在于通过让顾客满意和本组织所有成员及社会受益而达到长期成功的管理途径。

- 1 “全员 ” 指该组织结构中所有部门和所有层次的人员。
- 2 最高管理者强有力和持续的领导以及该组织内所有成员的教育和培训是这种管理途径取得成功所必不可少的。
- 3 在全面质量管理中，质量这个概念和全部管理目标的实现有关。
- 4 “社会受益 ” 意味着在需要时满足 “ 社会需求 ” 。



软件质量管理

- 质量管理的基本概念（质量管理体系评价）
- 软件质量管理的主要活动
- 认证和评估简介
- 计算机软件文档编制规范GB/T 8567-2006
- 软件质量管理实践
- 建立软件测试管理体系

软件质量管理

- 软件质量管理的主要活动
 - n 软件质量策划
 - 1) 软件质量策划的内容：
 - 2) 软件组织的质量过程
 - n 软件质量控制与保证
 - 质量控制的主要内容
 - 软件配置管理
 - 软件过程流管理
 - 软件质量保证
 - n 软件质量的度量和验证
 - n 软件质量改进

软件质量管理的主要活动

1、软件质量策划

1) 软件质量策划的内容：

- 确定软件组织，适应其生产特点的组织结构，以及人员的安排和职责的分配。
- 确定组织的质量管理体系目标，根据组织的商业需要和产品市场，确定选择ISO9000或CMM作为其质量管理体系的符合性标准或模型。
- 标识和定义组织的质量过程，即对组织的质量过程进行策划，确定过程的资源、主要影响因素、作用程序和规程、过程启动条件和过程执行结果规范等。
- 识别产品的质量特性，进行分类和比较，建立其目标、质量要求和约束条件。
- 策划质量改进的计划、方法和途径。



软件质量管理的主要活动

2) 软件组织的质量过程

通常包含两类：软件工程过程和组织支持过程

n 软件工程过程

就是通常所说的软件生命周期中的活动，一般包括需求分析、软件设计、编码、测试、交付、安装和维护。

CMM中，一个组织的软件过程策划一般包括两个阶段：组织标准生产过程的策划和项目产品策划。CMM中定义了三个关键过程域来实现这两级的过程策划：组织过程定义，软件项目策划，软件产品工程。

软件质量管理的主要活动

n 组织支持过程

是软件组织为了保证软件工程过程的实施和检查而建立的一组公共支持过程。主要包括：

- 管理过程：包括评审、检查、文档管理、不合格品管理、配置管理、内部质量审核和管理评审。
- 支持过程：包括合同评审、子合同评审、采购、培训、进货检验、设备检验、度量和服务。
- 在CMM中，有一些对应的关键过程区域：需求管理、软件子合同管理、软件质量保证、软件配置管理、培训程序、同行评审

软件质量管理的主要活动

2、软件质量控制与保证

软件质量控制的主要目标就是按照质量策划的要求，对质量过程进行监督和控制。质量控制的主要内容有：

- 1) 组织中与质量活动有关的所有人员，按照职责分工进行质量活动。
- 2) 所有质量活动按照已经策划的方法、途径、相互关系和时间，有序地进行。
- 3) 对关键过程和特殊过程，实施适当的过程控制技术以保证过程的稳定性，并在受控的情况下，提高过程的能力。
- 4) 所有质量活动的记录都被完整、真实地保存下来，以供统计分析使用。

软件质量管理的主要活动

实施软件质量控制通常涉及的技术：

- n 软件配置管理
 - 概述
 - 术语
 - 软件配置管理（SCM）的任务
- n 软件过程流管理
- n 软件质量保证



软件质量管理的主要活动--软件配置管理

○ 软件配置管理（简称SCM）

n 概述

软件配置管理：对软件生产过程中的所有有意义的中间产品进行管理。

变更不可避免，SCM协调软件开发使得混乱最小。

软件配置管理既不对变化需求的产生负责，也不对变化的实现负责，它只提供控制变化过程的机制。

n 几个术语：

被控制的项称为**软件配置项**(SCI，详见下页)，在实现SCM时，把SCI组织成**配置对象**(见后页)，评审通过的SCI成为**基线**(见后页)。

软件质量管理的主要活动--软件配置管理

○ 软件配置项 SCI

就是软件配置管理的对象。包括：

- *系统规格说明
 - *软件项目实施计划
 - * 软件需求说明
 - * 可执行的原型
 - * 初步的用户手册
 - * 设计规格说明
 - *源代码清单
 - * 操作和安装手册
 - * 测试计划和过程、测试用例和测试结果记录
 - * 可执行程序（可执行程序模块、连接模块）
 - * 数据库描述（模式和文件结构、初始内容）
 - * 正式的用户手册
- (接下页)



软件质量管理的主要活动--软件配置管理

○ 软件配置项 SCI (续)

*维护文档 (软件问题报告、维护请求、工程变更次序)

*软件工程标准

*项目开发总结

除以上SCI外，许多软件工程组织还把配置控制下的软件工具列入其中，即：编辑程序、编译程序、其它CASE工具的特定版本。

软件质量管理的主要活动--软件配置管理

○ 配置对象（1/2）

在实现SCM时，把SCI组织成配置对象，在项目数据库中用一个单一的名字来组织它们。

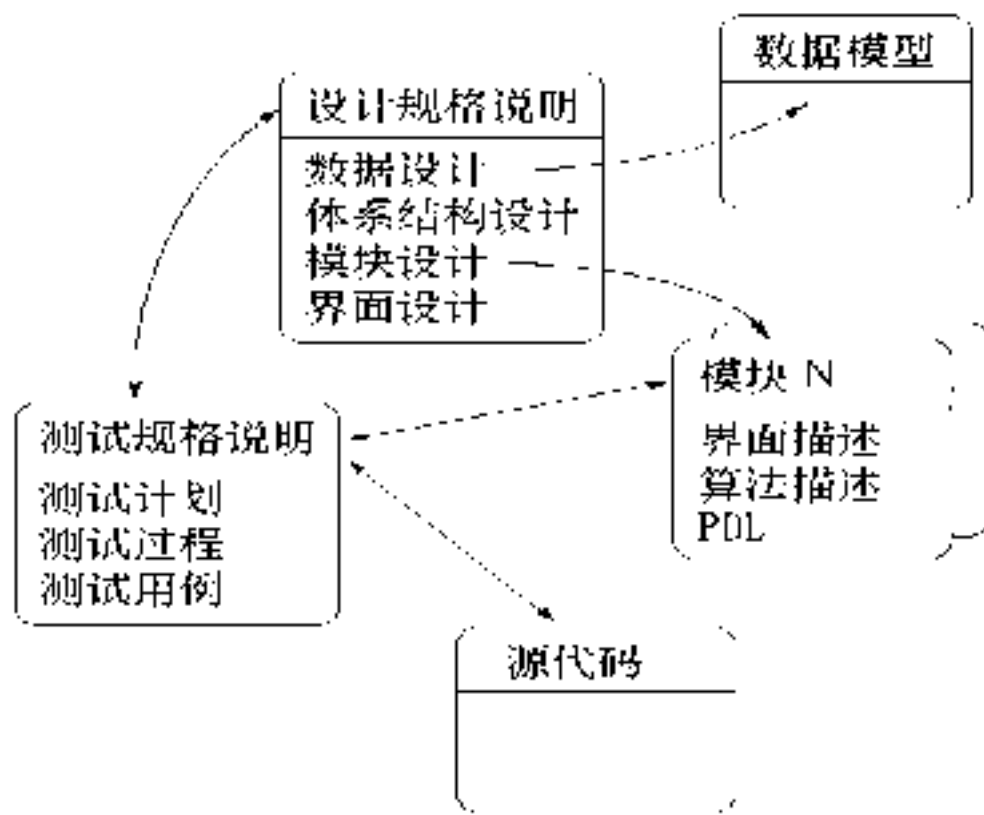
一个配置对象有一个名字和一组属性，并通过某些联系“连接”到其它对象。

“连接”用箭头表示。箭头指明了一种构造关系。双向箭头则表明一种相互关系。如：

软件质量管理的主要活动--软件配置管理

○ 配置对象 (2/2)

如果对某对象作了一个变更，软件工程师就可以根据这种相互关系确定，其它哪些对象（和SCI）可能受到影响。



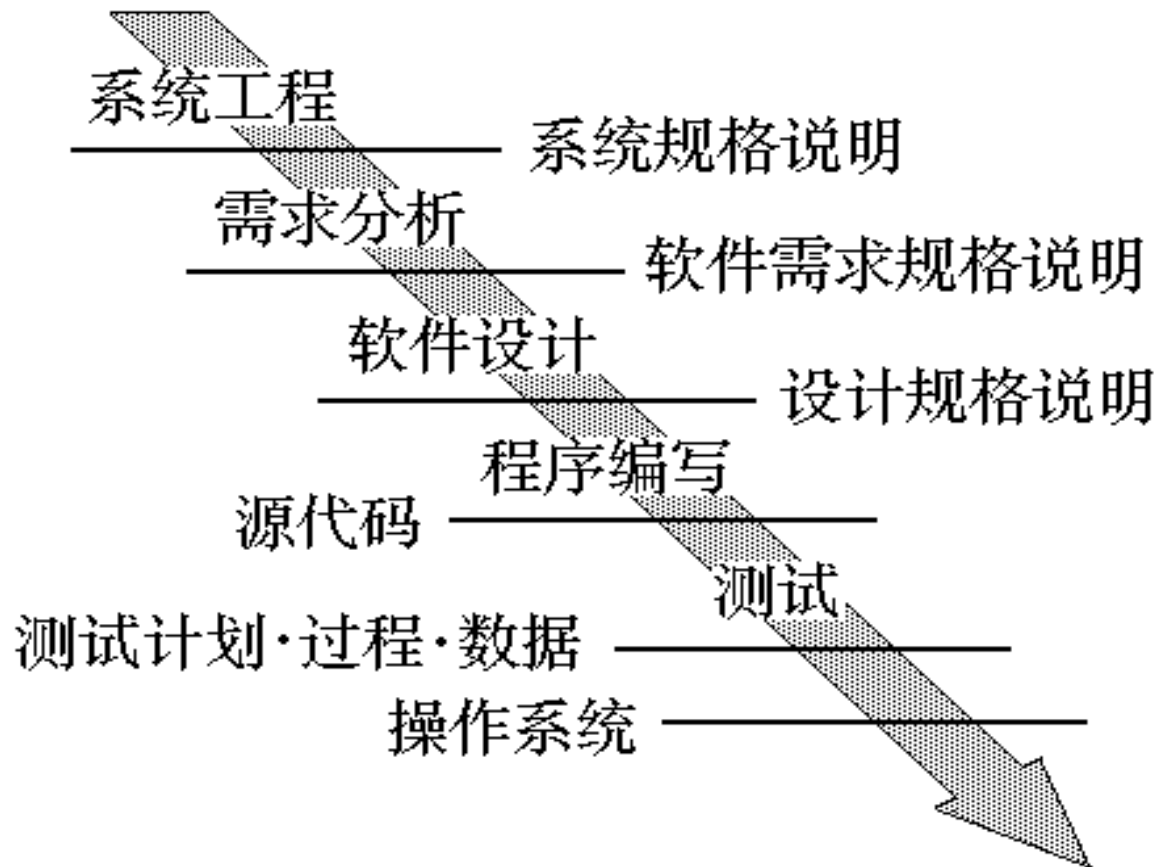
软件质量管理的主要活动--软件配置管理

○ 基线 (Baseline)

- n 基线是软件生存期中各开发阶段末尾的特定点，又称里程碑。
- n 由正式的技术评审而得到的SCI协议和软件配置的正式文本成为基线。
- n 基线的作用是把各阶段工作的划分更加明确化，以便于检验和肯定阶段成果。

软件质量管理的主要活动--软件配置管理

软件开发各阶段的基线



软件质量管理的主要活动--软件配置管理

○ 软件配置管理（SCM）的任务：

- n 标识SCI (配置标识、对象标识)
- n 版本控制
- n 控制变更
- n 审查软件配置
- n 报告所有加在配置上的变更。

软件质量管理的主要活动--软件配置管理

○ 配置标识

随着软件生存期的向前推进，SCI的数量不断增多。某一时刻的配置就是配置的一个片段。

为了方便对软件配置的各个片段（SCI）进行控制和管理，不致造成混乱，首先应给它们命名。

○ 对象类型

基本对象：在分析、设计、编码和测试时所建立的文本单元。如，基本对象可能是需求规格说明中的一节，一个模块的源程序清单、一组用来测试一个等价类的测试用例。

复合对象：是基本对象或其它复合对象的一个组合。

软件质量管理的主要活动--软件配置管理

- 对象标识：（名字、描述、资源、实现）
 - n 对象的名字明确地标识对象。
 - n 对象描述包括：SCI类型（如文档、程序、数据）、项目标识、变更和 / 或版本信息。
 - n 资源包括由对象产生的、处理的、引用的或其它需要的一些实体。
 - n 基本对象的实现是指向文本单元的指针，复合对象的实现为null。

软件质量管理的主要活动--软件配置管理

○ 命名对象之间的联系

n 对象的层次关系：

一个对象可以是一个复合对象的一个组成部分，用联系< is part of >标识。如：

data model < is part of > Design Specification;

只可以建立SCI的一个层次。

n 对象的相互关联关系：

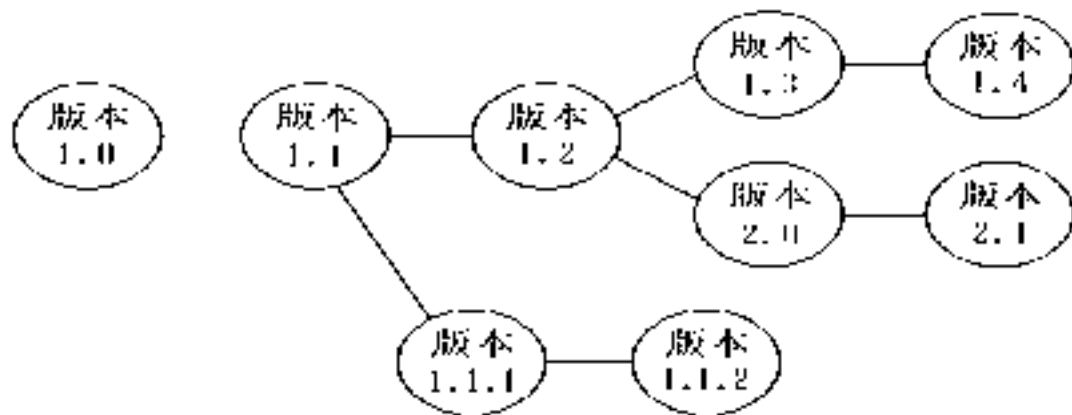
对象相互关联、交叉的结构联系表达方式如下：

data model <interrelated> data flow model;

软件质量管理的主要活动--软件配置管理

○ 演变图 (1/2)

- n 整个软件工程过程中所涉及的软件对象都必须加以标识。
- n 在对象成为基线以前可能要做多次变更，在成为基线之后也可能需要频繁地变更。
- n 对于每一配置对象都可以建立一个演变图，用演变图记叙对象的变更历史。



软件质量管理的主要活动--软件配置管理

○ 演变图（2/2）

在某些工具中，当前保持的只是最后版本的完全副本。

为了得到较早时期(文档或程序)的版本，可以从最后版本中“提取”出(由工具编目的)变更，使得当前配置直接可用，并使得其它版本也可用。

软件质量管理的主要活动--软件配置管理

○ 软件配置管理（SCM）的任务：

- n 标识SCI (配置标识、对象标识)
- n 版本控制
- n 控制变更
- n 审查软件配置
- n 报告所有加在配置上的变更。

软件质量管理的主要活动--软件配置管理

○ 版本控制（1/7）

n 概述

版本控制是SCM的基础，它管理并保护开发者的软件资源。

版本控制管理在软件工程过程中建立起配置对象的不同版本。

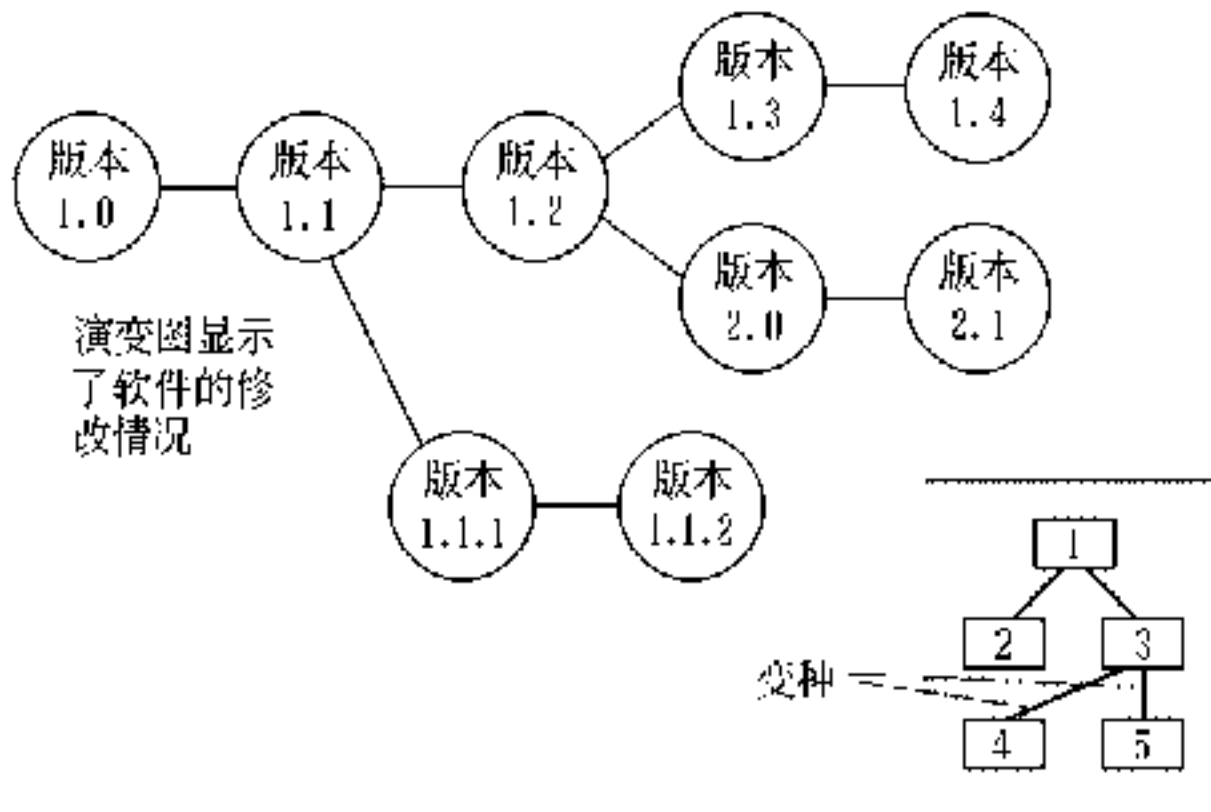
版本管理可以把一些属性结合到各个软件版本上。

通过描述所希望的属性集合来确定（或构造）所想要的配置。

n 使用演变图来表示系统的不同版本。

如

软件质量管理的主要活动--软件配置管理



软件质量管理的主要活动--软件配置管理

- 版本控制（3/7）

- n 版本管理的主要任务

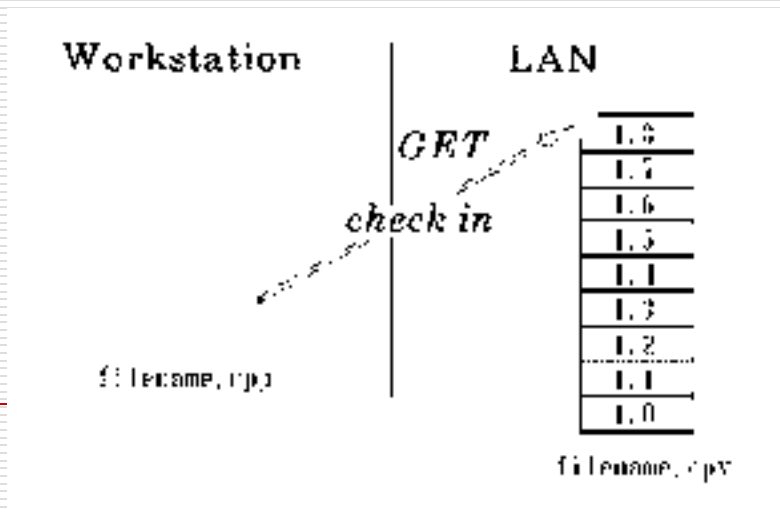
- 集中管理档案，安全授权机制：
 - 软件版本升级管理
 - 加锁功能：

软件质量管理的主要活动--软件配置管理

○ 版本控制（4/7）- 版本管理的主要任务

n 集中管理档案，安全授权机制：

- 版本管理的操作将开发组的档案集中地存放在服务器上，经系统管理员授权给各个用户。
- 用户通过登入（check in）和检出（check out）的方式访问服务器上的文件，未经授权的用户无法访问服务器上的文件。

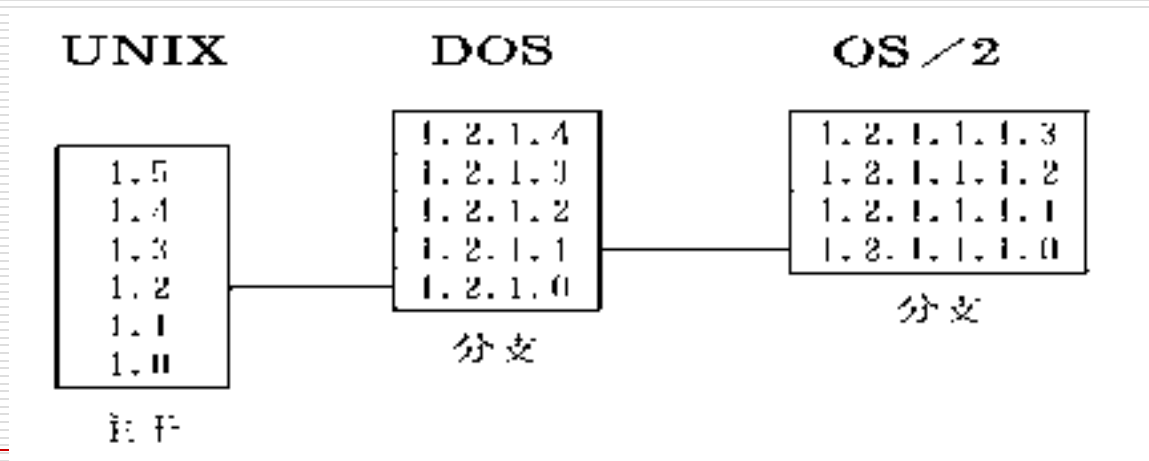


软件质量管理的主要活动--软件配置管理

○ 版本控制（5/7） - 版本管理的主要任务

n 软件版本升级管理：

- 每次登入时，在服务器上都会生成新的版本。
- 任何版本都可以随时检出编辑，同一应用的不同版本可以像树枝一样向上增长。



软件质量管理的主要活动--软件配置管理

○ 版本控制（6/7）- 版本管理的主要任务

n 加锁功能：

- 目的是在文件更新时保护文件，避免不同用户更改同一文件时发生冲突。
- 某一文件一旦被登入，锁即被解除，该文件可被其它用户使用。
- 在更新一个文件之前锁定它，避免变更没有锁定的项目源文件。

软件质量管理的主要活动--软件配置管理

○ 版本控制（7/7）- 版本管理的主要任务

- n 在文件登入和检出时，需要注意登入和检出的使用：
 - 当需要修改某个小缺陷时，应只检出完成工作必需的最少文件；
 - 需要对文件变更时，应登入它并加锁，保留对每个变更的记录；
 - 应避免长时间地锁定文件。如果需要长时间工作于某个文件，最好能创建一个分支，并在分支上工作。
 - 如果需要做较大的变更，可有两种选择：
 - a. 将需要的所有文件检出并加锁，然后正常处理；
 - b. 为需要修改的所有分支创建分支，把变更与主干“脱机”，然后把结果合并回去。

软件质量管理的主要活动--软件配置管理

○ 软件配置管理（SCM）的任务：

- n 标识SCI (配置标识、对象标识)
- n 版本控制
- n 控制变更
- n 审查软件配置
- n 报告所有加在配置上的变更。

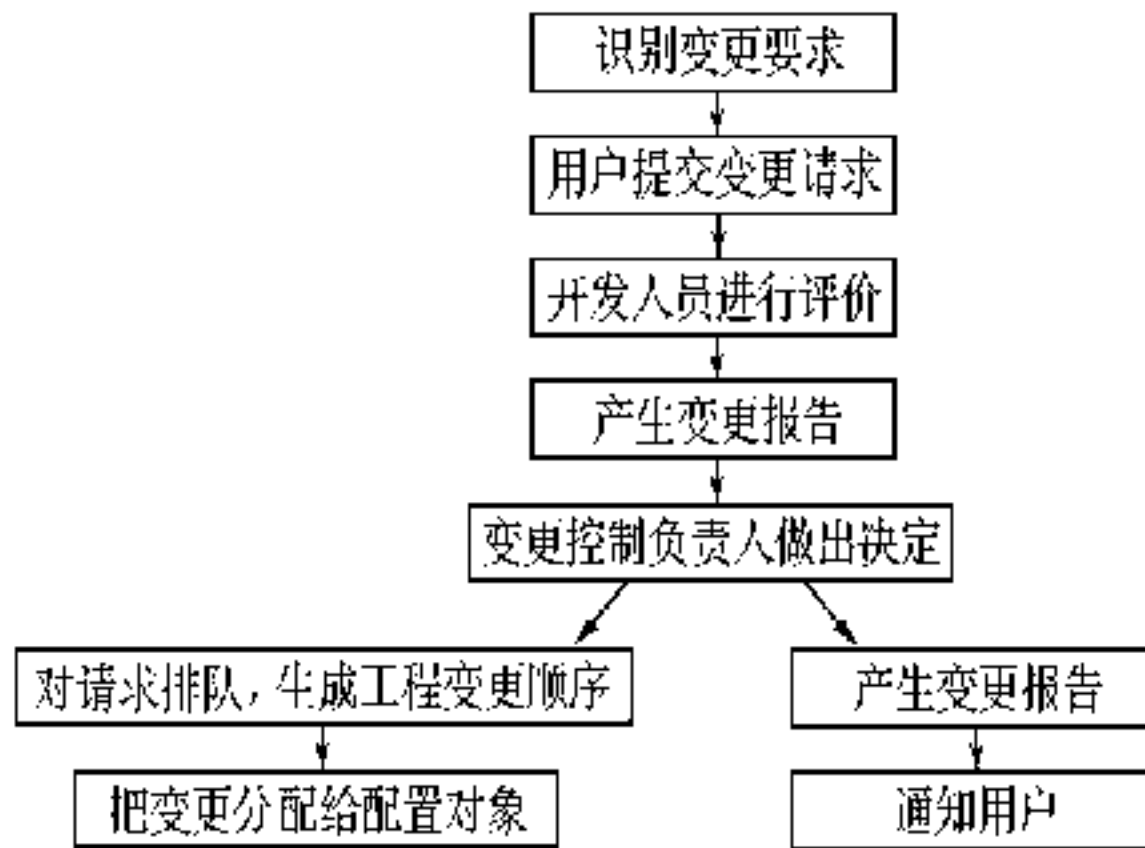
软件质量管理的主要活动--软件配置管理

○ 变更控制（1/4）

- n 软件生存期内全部的软件配置是软件产品的真正代表，必须使其保持精确。
- n 软件工程过程中某一阶段的变更，均要引起软件配置的变更，这种变更必须严格加以控制和管理，保持修改信息。
- n 变更控制包括建立控制点和建立报告与审查制度。

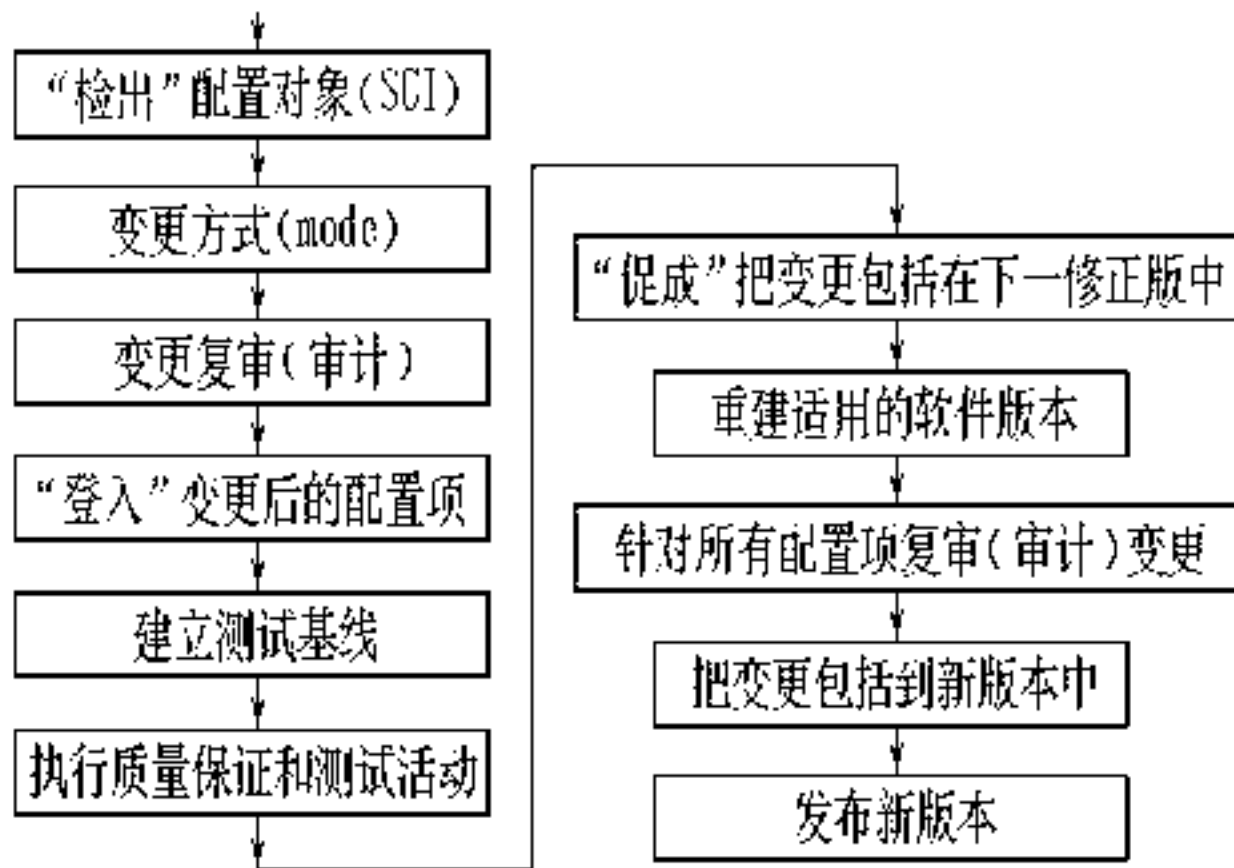
软件质量管理的主要活动--软件配置管理

○ 变更控制过程



软件质量管理的主要活动--软件配置管理

○ 变更控制过程





软件质量管理的主要活动--软件配置管理

○ 变更控制（4/4）

n 软件变更有两类不同情况：

○ 为改正小错误需要的变更。

必须，通常不需审查和批准。例外：发现错误的阶段在造成错误的阶段的后面。

○ 为增删功能、或为更改某功能的方法而需要的变更。

变更必须经过评价；如变更代价较小，通常批准变更；如代价较高，则须权衡利弊，以决定是否进行这种变更；如同意变更，需进一步确定由谁来支付变更所需要的费用。

软件质量管理的主要活动--软件配置管理

○ 软件配置管理（SCM）的任务：

- n 标识SCI (配置标识、对象标识)
- n 版本控制
- n 控制变更
- n 审查软件配置
- n 报告所有加在配置上的变更。

软件质量管理的主要活动--软件配置管理

○ 配置审计（1/2）

- n 软件的完整性，是指开发后期的软件产品能够正确地反映用户要求。
- n 软件配置审计的目的就是要
 - 证实整个软件生存期中各项产品在技术上和管理上的完整性。
 - 确保所有文档的内容变动不超出当初确定的软件要求范围。使得软件配置具有良好的可跟踪性。



软件质量管理的主要活动--软件配置管理

○ 配置审计（2/2）

- n 软件配置审计是软件变更控制人员掌握配置情况、进行审批的依据。
- n 软件的变更控制机制通常只能跟踪到工程变更顺序产生为止。为确认变更是否正确完成？一般可以用以下两种方法去审查：

○ 正式技术评审

着重检查已完成修改的软件配置对象的技术正确性

○ 软件配置审计

作为正式技术评审的补充，评价在评审期间通常没有被考虑的SCI的特性。

软件质量管理的主要活动--软件配置管理

○ 软件配置管理（SCM）的任务：

- n 标识SCI (配置标识、对象标识)
- n 版本控制
- n 控制变更
- n 审查软件配置
- n 报告所有加在配置上的变更

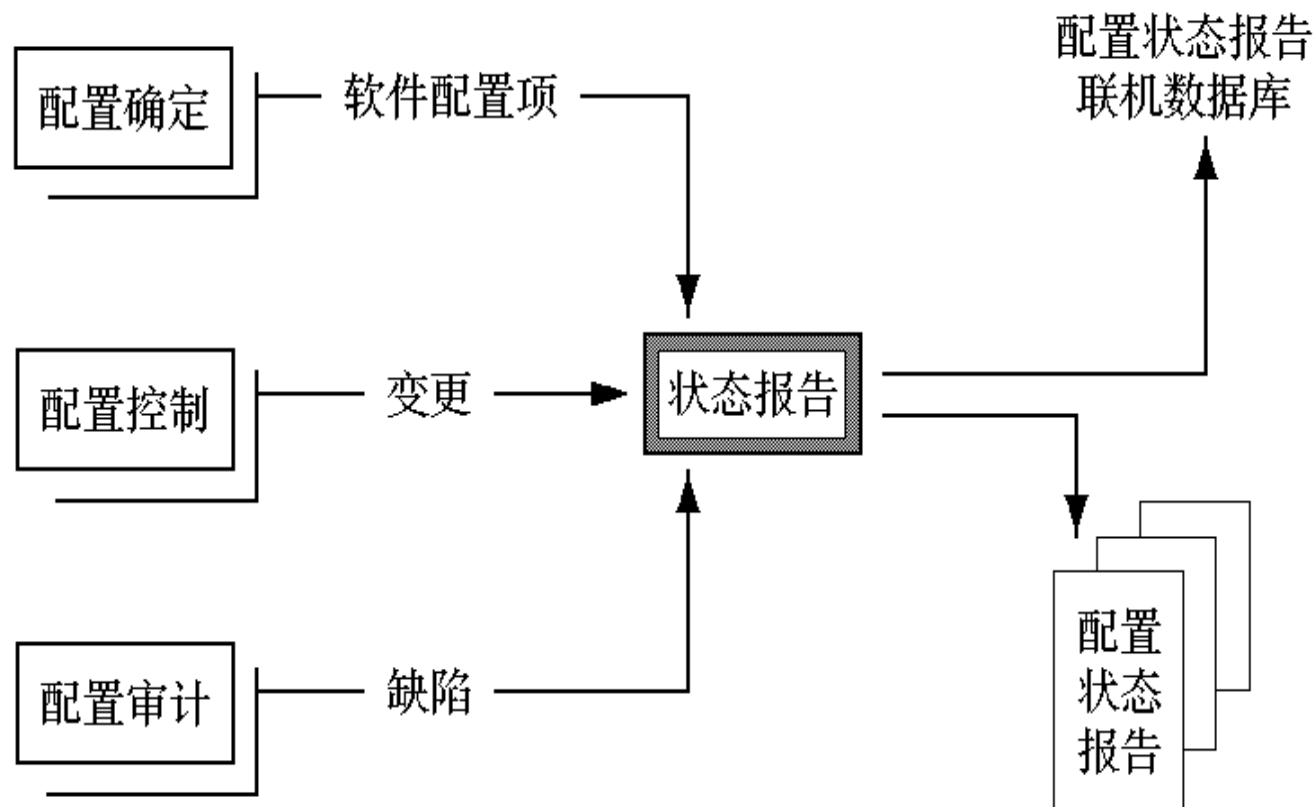
软件质量管理的主要活动--软件配置管理

○ 配置状态报告（1/4）

- n 为了清楚、及时地记载软件配置的变化，需要对开发的过程做出系统的记录，以反映开发活动的历史情况。这就是配置状态登录（记录）的任务。
- n 登录（记录）主要根据变更控制小组会议的记录，并产生配置状态报告。
- n 对于每一项变更，记录：发生了什么？为什么会发生？谁做的？什么时候发生的？会有什么影响？

软件质量管理的主要活动--软件配置管理

○ 配置状态报告信息流



软件质量管理的主要活动--软件配置管理

○ 配置状态报告（3/4）

- 每次新分配一个SCI，或更新一个已有SCI的标识，或一项变更申请被变更控制负责人批准，并给出了一个工程变更顺序时，在配置状态报告中就要增加一条变更记录条目。
- 一旦进行了配置审计，其结果也应该写入报告之中。

软件质量管理的主要活动--软件配置管理

○ 配置状态报告（4/4）

- 配置状态报告可以放在一个联机数据库中，以便软件开发人员或者软件维护人员可以对它进行查询或修改。此外在软件配置报告中新登录的变更应当及时通知给管理人员和软件工程师。
- 配置状态报告对于大型软件开发项目的成功起着至关重要的作用。避免了可能出现的不一致和冲突。

软件质量管理的主要活动－软件过程流管理

○ 软件过程流管理

现代质量理论认为：“质量形成于过程”。软件过程流管理是软件质量控制中非常重要的环节。过程流管理的基本原则是：

按计划 and 设定条件启动和结束过程流中的质量活动

按照计划对中间产品进行验证，防止不合格的产品转入下道工序。

记录和保持必要的过程活动的质量情况。

软件质量管理的主要活动 - 软件质量保证

○ 软件质量保证

质量保证是为保证产品和服务充分满足消费者要求的质量而进行的有计划、有组织的活动。

n 软件质量保证的目的

是向组织的内部或外部提供信任依据：

对内向组织的管理者表明组织的质量管理处于良好的状态，所有质量活动有效地运行；

对外向顾客表明，组织有能力满足顾客的质量要求，并提供符合质量要求的产品和服务。

软件质量管理的主要活动 - 软件质量保证

n 软件质量保证的工作内容

主要职责是检查开发和管理活动是否与已定的过程策略、标准和流程一致，检查工作产品是否遵循模板规定的内容和格式等。工作内容主要有：

- 1) 与SQA计划直接相关的工作
- 2) 参与项目的阶段性评审和审计
- 3) 对项目日常活动与规程的符合性进行检查
- 4) 对配置工作的检查和审计
- 5) 跟踪问题的解决情况
- 6) 收集新方法，提供过程改进的依据

软件质量管理的主要活动 - 软件质量保证

n 软件质量保证的主要任务

为了提高软件的质量和软件的生产率，软件质量保证的主要任务大致可归结为8点。

1. 用户要求定义

- 熟练掌握正确定义用户要求的技术
- 熟练使用和指导他人使用定义软件需求的支持工具
- 重视领导全体开发人员收集和积累有关用户业务领域的各种业务的资料和技术技能。

软件质量管理的主要活动 - 软件质量保证

n 软件质量保证的主要任务（续）

2. 力争不重复劳动

- 考虑哪些既有软件可以复用
- 在开发过程中，随时考虑所生产软件的复用性。

软件质量管理的主要活动 - 软件质量保证

n 软件质量保证的主要任务（续）

3. 掌握开发新软件的方法

在开发新软件的过程中大力使用和推行软件工程学中的开发方法和工具。

- 使用先进的开发技术：如结构化技术、面向对象技术
- 使用数据库技术或网络化技术
- 应用开发工具或环境
- 改进开发过程

软件质量管理的主要活动 - 软件质量保证

n 软件质量保证的主要任务（续）

4. 组织外部力量协作的方法

- 一个软件自始至终由同一个软件开发单位来开发，也许是最理想的。但在现实中常常难以做到。
- 改善对外部协作部门的开发管理。必须明确规定进度管理、质量管理、交接检查、维护体制等各方面的要求，建立跟踪检查的体制。

软件质量管理的主要活动 - 软件质量保证

n 软件质量保证的主要任务（续）

5. 排除无效劳动

- 最大的无效劳动就是因需求规格说明有误、设计有误而造成的返工。定量记录返工工作量，收集和分析返工劳动花费数据
- 较大的无效劳动是重复劳动，即相似的软件在几个地方同时开发
- 建立互相交流、信息往来通畅、具横向交流特征的信息流
通网

软件质量管理的主要活动 - 软件质量保证

n 软件质量保证的主要任务（续）

6. 发挥每个开发者的能力

- 软件生产是人的智能生产活动，它依赖于人的能力和开发组织团队的能力。
- 开发者必须有学习各专业业务知识、生产技术和和管理技术的能动性。
- 管理者或产品服务者要制定技术培训计划、技术水平标准，以及适用于将来需要的中长期技术培训计划。

软件质量管理的主要活动 - 软件质量保证

n 软件质量保证的主要任务（续）

7. 提高软件开发的工程能力

- 要想生产出高质量的软件产品必须有高水平的软件工程能力。
- 在软件开发环境或软件工具箱的支持下，运用先进的开发技术、工具和管理方法开发软件的能力。

软件质量管理的主要活动 - 软件质量保证

n 软件质量保证的主要任务（续）

8. 提高计划和管理质量能力

- 项目开发初期计划阶段的项目计划评价
- 计划执行过程中及计划完成报告的评价
- 将评价、评审工作在工程实施之前就列入整个开发工程的工程计划中
- 提高软件开发项目管理的精确度

软件质量管理的主要活动

--软件质量的度量和验证

3．软件质量的度量和验证

软件质量度量类型

产品质量度量

通常产品质量度量依赖于具体的产品标准，通过测量获得产品质量特性的有关数据，辅以合适的统计技术以确定产品或同批产品是否满足了规定的质量要求。

过程质量度量

通过对软件产品设计、开发、检查、评审等过程的度量技术的使用，来度量软件过程的进度、成本是否按计划保证，质量计划的变化频率，变化的诱因以及风险的管理等等。

软件质量管理的主要活动

软件质量验证

ISO 9000：2000中对验证（Verification）的定义是：“通过提供客观证据对规定要求已得到满足的认定”。CMM在关键过程域（KPA）的公共特征（Common Feature）- 验证实现（Verifying Implementation）中这样描述：“验证实现是保证活动按照已经建立的过程执行的一系列步骤，典型的验证有管理部门的评审、审核和软件质量保证”。

在软件质量管理中，对软件产品的验证通常包括：对各级设计的评审、检查，各个阶段的测试等。对软件过程的验证，则是对过程数据的评审和审核。

软件质量管理的主要活动 - 软件质量改进

4 . 软件质量改进

质量改进是现代质量管理的必然要求，ISO 9000要求组织定期进行内审和管理评审，采取积极有效的纠正预防措施，保持组织的质量方针和目标持续适合组织的发展和受益者的期望。

具体进行软件过程改进的活动包括：

度量与审核

纠正和预防措施

管理评审

质量管理认证和评估简介

- [ISO9000标准简介](#)
- [软件过程成熟度模型（CMM）](#)
- [ISO 9000和CMM的联系](#)
- [ISO/IEC15504](#)

ISO9000标准简介

- ISO9000标准概述
- ISO9000:2000总体结构
- ISO9000核心标准简介
- ISO9000:2000标准的变化
- 对ISO9000的误解



ISO9000标准简介

○ ISO管理层次

ISO技术工作高度分散，主要由下列部门承担：

- n 技术委员会(简称TC)：185个，负责相关领域标准的制定。
- n 分技术委员会(简称SC)：611个，负责相关领域标准的制定。
- n 工作组(简称WG) 2022个，完成规定的专项任务。
- n 特别工作组 38个

ISO的2856个技术机构技术活动的成果(产品)是“国际标准”。ISO现已制定出国际标准共10300多个，主要涉及各行各业各种产品(包括服务产品、知识产品等)的技术规范。



ISO9000标准简介

○ 概述

ISO9000是指由国际标准化组织（ISO）所属的质量管理和质量保证技术委员会ISO/TC176工作委员会制定并颁布的关于质量管理体系的族标准的统称。如，它包括(94版)：

ISO8402——质量术语标准

ISO9000——质量管理与质量保证标准

ISO9001——质量体系——设计、开发、生产、安装与服务的质量保证模式

ISO9002——质量体系——生产与安装的质量保证模式

ISO9003——最终检验与实验的质量保证模式

ISO9004——质量管理与质量体系要素

ISO9000标准概述

○ 适用

ISO9000族标准作为质量管理和质量保证标准适用于所有希望改进质量管理绩效和质量保证能力的组织。

○ ISO9000标准版本历史

87年被ISO组织TC/176发布以来，已经通过87版、94版、2000版和2008版多次改版。

2000版标准已于2000年3月被我国转化为国家标准，企业实施于2001年4月，是较94版标准更科学、合理和便于采用的标准。

2008版于2008年10月31日正式发布。标准修改的较少，无理由需要“过渡阶段”，ISO将用6~12个月时间来结束ISO9001:2000版的使用。

ISO9000:2000总体结构

○ 总体结构

2000版ISO 9000族标准的总体结构如下：

(1) 4个核心标准：

ISO9000：2000 基本原理和术语

ISO9001：2000 品质管制体系-要求

ISO9004：2000 品质管制体系-业绩改进指南

ISO19011 质量和环境管理审核指南；

(2) 1个其他标准：

ISO10012 《测量设备的质量保证要求》

ISO9000:2000总体结构

(3) 若干份技术报告：

ISO/TR10006 专案管理指南

ISO/TR10007 技术状态管理指南

ISO/TR10013 品质管制体系档指南

ISO/TR10014 质量经济性指南

ISO/TR10015 教育和培训指南

ISO/TR10017 统计技术在ISO9001中的应用指南；

(4) 若干份小册子：

品质管制原理、•选择和使用指南

ISO9001在小型企业中的应用指南

ISO9000核心标准简介

○ 4个核心标准简介

n ISO9000：2000

本标准规定了质量管理体系的术语和基本原理，取代1994版ISO8402和ISO9000-1两个标准。

本标准提出的8项质量管理原则，是在总结了质量管理经验的基础上，明确了一个组织在实施质量管理中必须遵循的原则，也是2000版9000族标准制定的指导思想和理论基础。

本标准第二部分提出10个部分87个术语。在语言上强调采用非技术性语言，使所有潜在用户易于理解。为便于使用，在标准附录中，推荐了以“概念图”方式来描述相关术语的关系。

第三个重点内容是，提出了质量管理体系的基本原理。作为对本标准引言中质量管理8项原则的呼应。



ISO9000核心标准简介

○ 4个核心标准简介

n ISO9001：2000

规定了质量管理体系要求，用于证实组织具有提供满足顾客要求和实用法规要求的产品的能力，目的在于增进顾客的满意。

本标准取代了1994版三个质量保证标准（ISO9001:1994、ISO9002:1994和ISO9003:1994）。新版的质量管理体系要求，采用了“过程方法模型”，以取代1994版ISO9001标准中的20个要素。

为适应不同类型的组织需要，在一定情况下，体系要求允许删减（剪裁）。

新版名称中不再出现“质量保证”一词，这反映了标准规定的质量管理体系要求包括了产品质量保证和顾客满意两层含义。

ISO9000核心标准简介

○ 4个核心标准简介

n ISO9001 : 2000

目录

- 1 . 范围
- 2 . 引用标准
- 3 . 术语和定义
- 4 . 质量管理体系
- 5 . 管理职责
- 6 . 资源管理
- 7 . 产品/服务的实现

ISO9000核心标准简介

○ 4个核心标准简介

n ISO9001 : 2008

ISO9001:2008的新标准于2008年10月31日正式发布。

标准修改的较少，2009年10月31日结束ISO9001:2000版的使用。

对于大多数组织而言，通常正常的监督评审过渡即可，不需要额外时间。

ISO9000核心标准简介

○ 4个核心标准简介

n ISO9004 : 2000

提供考虑质量管理体系的有效性和效率两方面的指南。该标准的目的是组织业绩改进和其他相关方满意。

本标准是1994版ISO9004-1的替代标准。

ISO9004 : 2000和ISO9001 : 2000是一对协调一致并可一起使用的质量管理体系标准，两个标准采用相同的原则，但其适用范围不同，而且 ISO9004标准不拟作为ISO9001标准的实施指南。通常情况下，当组织的管理者希望超越ISO9001标准的最低要求，追求增长的业绩改进时，往往以ISO9004标准作为指南。

ISO9000核心标准简介

○ 4个核心标准简介

n ISO19011

本标准是ISO/TC176与ISO/TC207（环境管理技术委员会）联合制订的，以遵循“不同管理体系，可以共同管理和审核”的原则。新版ISO19011标准将合并并取代ISO10011-1、ISO10011-2、ISO10011-3和ISO14010、ISO14011、ISO14012等几个标准。

本标准在术语和内容方面，兼容了质量管理体系和环境管理体系两方面特点。

本标准为审核基本原则、审核大纲的管理、环境和质量管理体系的实施以及对环境和质量管理体系评审员资格要求提供了指南。



ISO9000核心标准简介

○ 4个核心标准简介

n ISO19011标准的主要内容

审核基本原则

与审核员有关的原则：合乎道德的行为；公正的表达；应有的职业素养。

与审核过程有关的原则：独立性；基于证据的方法。

审核方案管理

包括一切有助于每次审核的实施而需要进行的活动

审核活动

包括具有特定目标和范围的一次审核的各个步骤。

审核员能力

审核员的能力要求和评价方法



ISO9000核心标准简介

○ ISO9000：2000质量管理体系文件结构：

1．质量手册（QM）：

主要功能是将管理层的质量方针及目标以文件形式告诉全体员工或顾客。

2．程序文件（QP）：

是指导员工如何进行及完成质量手册内容所表达的方针及目标的文件。

3．作业指导书（WI）：

详细说明特定作业是如何运作的文件。

4．记录表格（F）：

是用于证实产品或服务是如何依照所定要求运作的文件。

ISO9000:2000标准的变化

2000版标准正式发布三年期满后，94版标准立即废止。2000版标准总体变化如下：

一、由四个标准组成：

1、ISO9000作为选用标准，同时也是名词术语标准，即94版ISO9000 - 1标准与8402的结合。

2、ISO9001标准代替94版三个模式标准，按94版ISO9002标准获证的企业在复审时，允许对2000版ISO9001标准进行裁剪。

3、ISO9004标准代替94版ISO9004 - 1多项分标准。

4、ISO / CD.1 19011标准代替94版ISO10011标准和94版环境ISO14010、ISO14011、ISO14012。

ISO9000:2000标准的变化

二、思路和结构上的变化

- 1、明确质量管理8条原则作为新版质量管理体系的基础。（见下页）
- 2、把过去三个外部保证模式ISO9001、ISO9002、ISO9003合并为ISO9001标准，允许通过裁剪适用不同类型的企业，同时对裁剪也提出了明确严格的要求。
- 3、把过去按20个要素排列，改为按过程模式重新组建结构，将有关过程组织成4个大过程，即管理职责；资源管理；产品实现；测量、分析和改进四大过程。（见后页图）
- 4、引入PDCA戴明环闭环管理模式，使持续改进的思想贯穿整个标准，要求质量管理体系及各个部分都按PDCA循环，建立实施持续改进结构。（见后页）
- 5、适应组织管理一体化的需要。

ISO9000:2000 质量管理八项原则

- **a.以顾客为关注焦点**

组织依存于其顾客。因此组织应理解顾客当前和未来的需求，满足顾客要求并争取超越顾客期望。

- **b.领导作用**

领导者确立本组织统一的宗旨和方向。他们应该创造并保持使员工能充分参与实现组织目标的内部环境。

- **c.全员参与**

各级人员是组织之本，只有他们的充分参与，才能使他们的才干为组织获益。

- **d.过程方法**

将相关的活动 and 资源作为过程进行管理，可以更高效地得到期望的结果。

ISO9000:2000 质量管理八项原则

- **e.管理的系统方法**

识别、理解和管理作为体系的相互关联的过程，有助于组织实现其目标的效率和有效性。

- **f.持续改进**

组织总体业绩的持续改进应是组织的一个永恒的目标。

- **g.基于事实的决策方法**

有效决策是建立在数据和信息分析基础上。

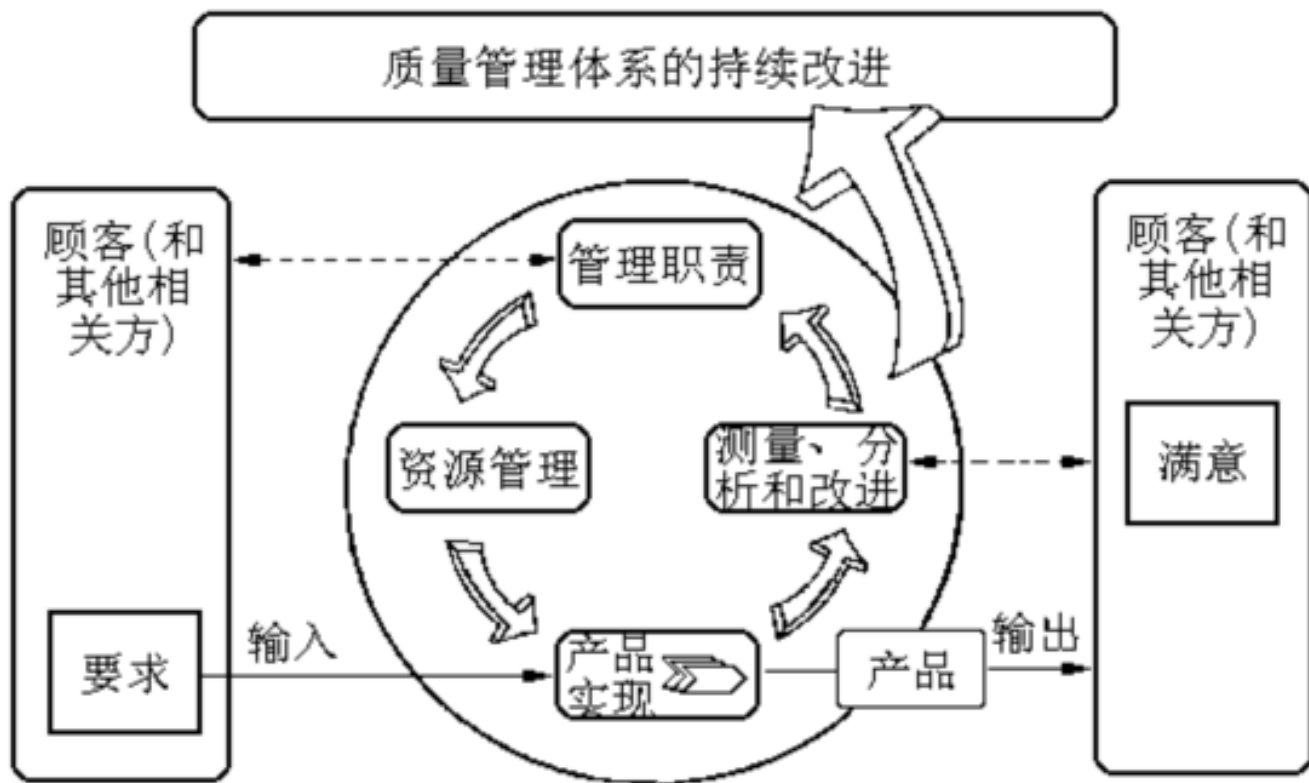
- **h.互利的供方关系**

组织与其供方是相互依存的，互利的关系可增强双方创造价值的能力。

过程方法模式

——→
增值活动
- - - - -→
信息流

括号中的陈
述不适用于
GB/T19001

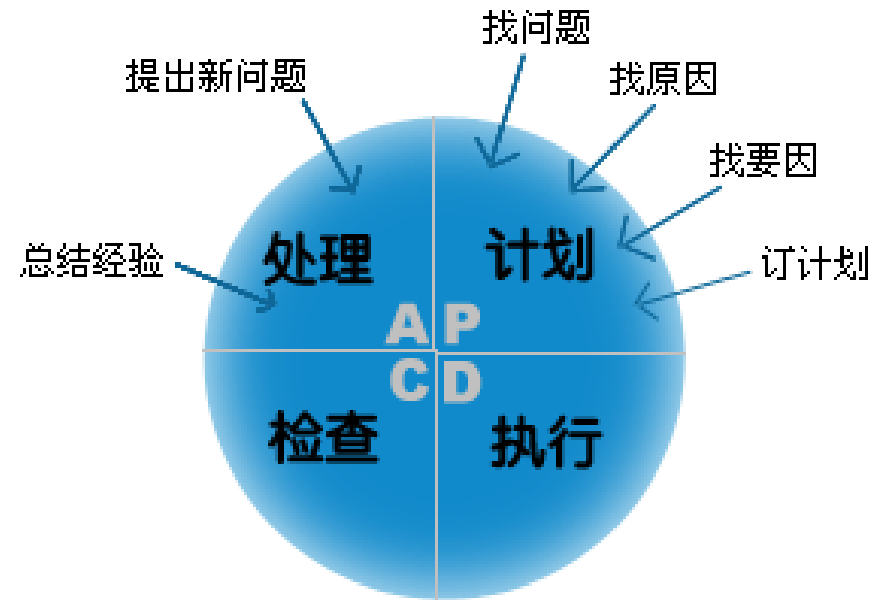


PDCA戴明环

PDCA循环又叫戴明环，是美国质量管理专家戴明博士首先提出的，它是全面质量管理所应遵循的科学程序。全面质量管理活动的全部过程，就是质量计划的制订和组织实现的过程，这个过程就是按照PDCA循环，不停顿地周而复始地运转的。

PDCA四个英文字母及其在PDCA循环中所代表的含义如下：

- 1、P(Plan)--计划，确定方针和目标，确定活动计划；
- 2、D(Do)--执行，实地去做，实现计划中的内容；
- 3、C(Check)--检查，总结执行计划的结果，注意效果，找出问题；
- 4、A(Action)--行动，对总结检查的结果进行处理，成功的经验加以肯定并适当推广、标准化；失败的教训加以总结，以免重现，未解决的问题放到下一个PDCA循环。



ISO9000:2000标准的变化

三、特点：

- 1、通用性强，94版ISO9001标准主要针对硬件制造业，新标准则适用于硬件、软件、流程性材料和服务等行业。
- 2、更先进、更科学，总结补充了企业质量管理中一些好的经验，突出了八项质量管理原则。
- 3、对94版标准进行简化，简单好用。
- 4、提高了其它管理的相容性，协调了环境管理和财务管理。
- 5、ISO9001标准和ISO9004标准作为一套标准，互相对应，协调一致。

对ISO9000的误解

○ ISO9000 不适用于软件组织

ISO9000不是产品标准,也不是技术标准,而是指导企业建立、实施质量管理体系的管理标准。它具有两种主要职能,即管理和保证职能。

管理职能是按ISO9000要求,建立和实施一个有效的质量体系,进而提高企业的管理水平;

保证职能是指通过一个第三方机构,证实企业所建立的质量体系,并证明企业具有提供符合要求的产品的能力,从而向顾客或用户提供可信度。

ISO9000将这两种职能有机结合起来,为企业提供了一个比较科学的管理和保证机制,它是任何企业都需要的。因此,ISO9000适用于所有的工业类型的企业。

对ISO9000的误解

○ 软件组织难于实施ISO9000

由于软件开发的特殊性,软件企业很难实施ISO9000。

特殊性是指:软件开发根据项目设立,不存在完全相同的项目开发;对于规模不同的项目如何实施ISO9000;软件开发进度难以估计,应用ISO9000会使工作量加大,从而导致进度拖延。

ISO9000要求建立一个文件化的质量体系,这个体系要求全员参与,并明确各个岗位的职责,只能严格地按照文件规定的要求执行。

对ISO9000的误解

○ 软件组织难于实施ISO9000(续)

对于每个软件项目的大小、规模不等这一特性,可以将软件作为一个具体的"产品"对待,至于这个产品所采用的开发过程如何,或者说采用了哪几个生存周期过程,则在体系中不作统一要求,完全可以根据该项目的大小、规模,并依据质量体系的原则要求确定。因此,"软件按项目开发及项目的大小、规模不同"完全不妨碍软件企业应用ISO9000,相反应用了ISO9000,项目管理更加科学、系统。

至于说到项目的进度问题,应用ISO9000并不会造成进度拖延,相反会使项目拖延问题得到改善。这是因为造成进度拖延的主要原因是由于管理不善,没有严格地按照规定的要求去作,而应用ISO9000则能够解决管理上的漏洞,使项目拖延问题得到最大限度的改善。

对ISO9000的误解

○ 软件企业实施ISO9000的效果不理想

软件开发是高度知识密集型的工作,对开发人员的知识和技术水平要求较高。实施ISO9000能否达到预期的效果,即保证软件企业具有开发出符合要求的软件产品的能力?

对ISO9000的误解

○ 软件企业实施ISO9000的效果不理想

ISO9000没有对“软件开发人员的技术水平”提出要求,而软件开发确实与开发人员的技术水平密切相关,但这不是质量体系所能解决的,因而并不妨碍ISO9000在软件企业的应用效果。

将ISO9000要求同CMM对比,就会发现CMM中规定的五级水平的第1级和第2级基本上都是管理上的要求,也就是说按ISO9000要求去作的软件企业基本能达到CMM所规定的第2级以上的水平。可以说ISO9000是软件企业达到高级水平所必须的。因此,我们虽然不能说应用了ISO9000的企业就具有了开发出符合要求的软件产品的能力,但却可以说软件企业要想具有开发出符合要求的软件产品,应用ISO9000则是其最佳选择之一。

软件过程成熟度模型

- 基本概念
- 软件机构的成熟性
- 软件过程成熟度模型
- 成熟度提问单

软件过程成熟度模型-基本概念

○ 软件过程

- n 过程：为达到目的而执行的所有步骤的序列。
- n 软件过程：人们在开发和维护软件及其相关产品时所涉及的各种活动、方法、实践和改革等。其中软件相关产品包括软件项目计划、设计文档、程序代码、测试用例和用户手册等。
- n 软件生成期过程

ISO/IEC-12207:1995《信息技术--软件生存期过程》
(IDT GB/T 8566-2001) : 将软件生存期过程分为3类(基本、支持和组织类) , 17个过程 ; 每个过程包含若干活动 , 总共74项活动 ; 每个活动是一组相互协调的作业 , 总共232个作业。作业表示为某种要求、自我说明、建议或可允许的活动。

软件过程成熟度模型-基本概念

○ 基本过程

包括5个过程，获取、供应、开发、运行和维护。

这些过程供各主要当事方(获取方、供方)在参与或完成软件开发、运行或维护时使用。

○ 支持过程

包括8个过程，文档编制、配置管理、质量保证、验证、确认、联合评审、审核、问题解决。

每个过程均有明确的目的支持其他过程，帮助软件项目获得成功及良好的产品质量。

○ 组织过程

包括4个过程，管理、基础设施、改进、培训。

这些过程使前二类过程得以建立、实施和改进。如被用来建立和实现与生成期过程相关的基础结构、人事制度，并使其不断改进。

过程思维

○ 过程特征

- n 规定或包含了若干主要活动
- n 要利用资源，在制约条件下给出中间产品和最终产品
- n 过程可包含若干个子过程
- n 每个过程活动都有其入出口准则，这些准则表明了活动在什么情况下开始和结束
- n 过程的活动是按顺序组织的
- n 每个过程有一套解释各项活动目标的指导原则
- n 对过程活动，所有资源和产品都会有某些约束或控制
- n 过程的活动应有自己的目标，过程的实施把组织、管理者、人员和技术基础设施聚起来，以实现过程的业务目标为目的

任务思维与过程思维

思维对比	面向任务的思维	面向过程的思维
注重	任务、设备、人员、组织结构	总体目标、各部分间协调性、一致性
流行期	近200年	近10-20年
特点	将工作任务分解，指派人员分头完成	消除了各部分工作间的冲突、提高了总体效率、有利于达到总体目标
影响	机构的组织结构、工作效率	机构的全部活动

软件过程成熟度模型

要区分不成熟和成熟软件组织，需要构造一个软件过程成熟度框架，它描述一条从无序到有序的软件过程的进化途径，框架是支持持续改进的基础。

- 软件过程成熟度框架包括四个概念：
 - n 软件过程
 - n 软件过程能力
 - n 软件过程性能
 - n 软件过程成熟度：CMM分为五级



软件过程成熟度模型

○ 几个概念

- n 软件过程能力：当遵循某个软件过程时所能达到的期望效果，它可以有效预测企业接收新的软件项目时可能得到的结果。
- n 软件过程性能：当遵循某个软件过程时所达到的实际效果。它可以用于验证软件过程能力。
- n 软件过程成熟度：指一个特定的软件过程被显式定义、管理、度量、控制和实施的程度。成熟度可以用于指示企业加强其软件过程能力的潜力。

软件过程成熟度模型

○ 软件机构的成熟度

不同的软件开发机构，管理策略不同，因而软件项目所遵循的软件过程也不同。在此，可用软件机构的成熟度（Maturity）加以区别。

n 不成熟软件机构的特征：

软件过程一般在项目进行过程中由参与开发的人员临时确定。有时即使确定，实际也不严格执行；

软件机构是反应型的，管理人员经常要集中精力去应付难以预料的突发事件；

项目的进度和经费预算由于估计的不切实际，常常突破。在项目进度拖延，交付时间紧迫的情况下，往往不得不削减软件的功能，降低软件的质量；

产品质量难以预测。质量活动，如质量评审、测试等，常被削弱或被取消。

软件过程成熟度模型

n 成熟软件机构具有的特征：

建立了机构级的软件开发和维护过程。软件人员对其有较好的理解。一切活动均遵循过程的要求进行，做到工作步骤有次序，且有章可循；

软件过程必要时可做改进，但需在经小型试验和成本-效益分析的基础上进行；

软件产品的质量和客户对软件产品的满意程度不是由开发人员，而是由负责质量保证的经理负责监控；

项目进度和预算是根据以往项目取得的实践经验确定，因而比较符合实际情况。

软件过程成熟度模型

○ 软件过程成熟度模型

1987年，美国卡内基-梅隆大学软件工程研究所(SEI)受美国国防部资助，提出了软件能力成熟度模型CMM，它用来定义和评价软件公司开发过程的成熟度，提供怎样才能提高软件质量的指导。

1991年SEI推出CMM1.1

1997年发布了CMM2.0

在CMM后，又出现了若干个基于CMM且针对特定主题的模型。为区别，通常把初始的CMM成为SW-CMM。下表列出了这些模型。

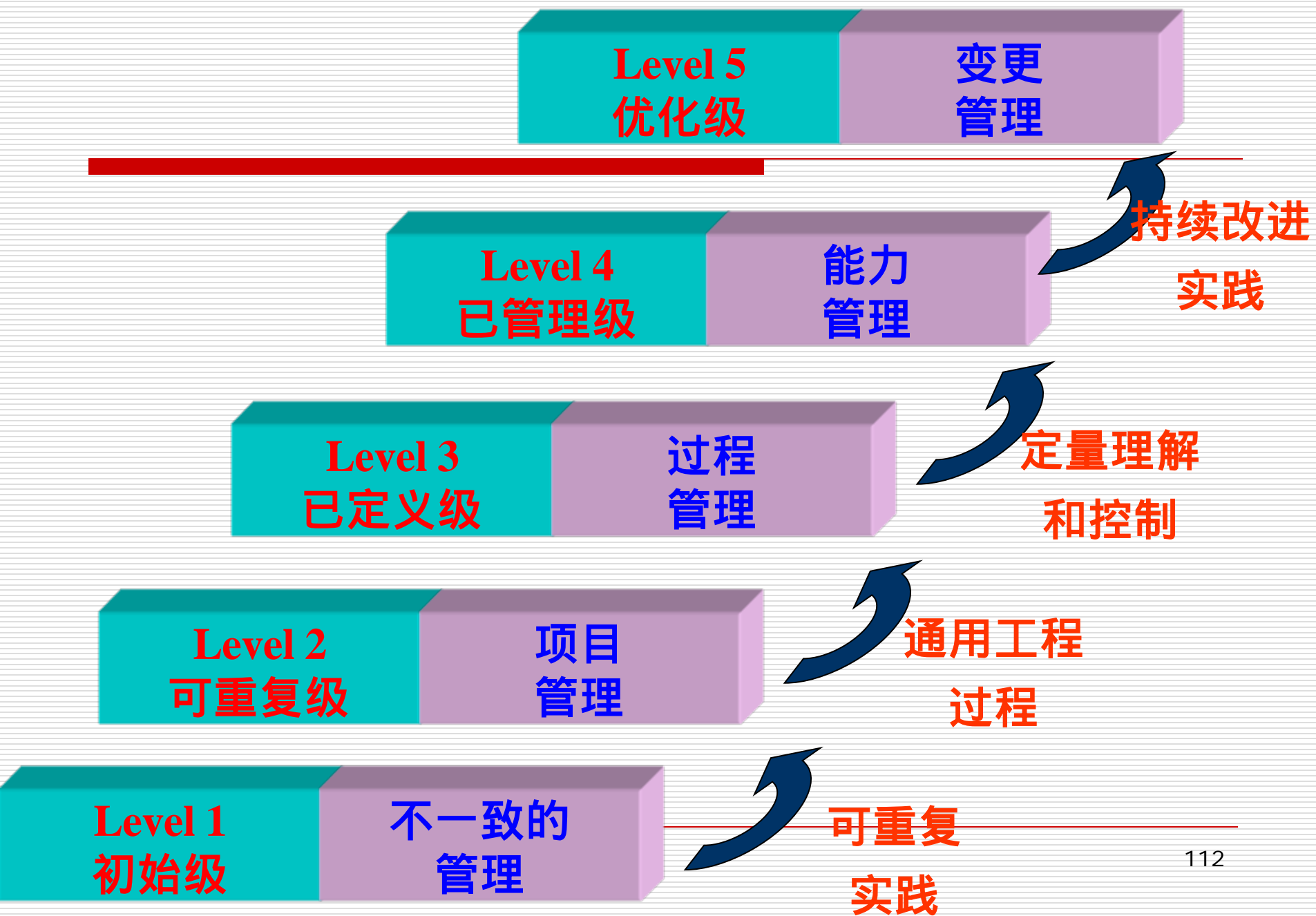
CMM的相关模型

名称	含义	内容
P-CMM	people	人力资源管理
SE-CMM	System engineering	系统工程管理
SA-CMM	Software acquisition	软件获取
IPD-CMM	Integrated systems product development	集成系统开发管理
SSE-CMM	systems security engineering	系统安全工程管理
CMMI	Capability maturity model integration	集成模型（综合上列模型）
TSP	Team software process	团队协调工作
PSP	Personal software process	软件工程师工作

软件过程成熟度模型

○ CMM的五个成熟度等级

CMM描述了五个级别的软件过程成熟度（初始级 可重复级 已定义级 已管理级 优化级），成熟度反映了软件过程能力（Software Process Capability）的大小，任何一个软件机构的软件过程必定属于其中某个级别。五个成熟度如图所示：



软件过程成熟度模型

初始级 (initial)

- n 工作无序，项目进行过程中常放弃开始制定的计划；
- n 管理无章，缺乏健全的管理制度；
- n 开发项目成效不稳定，产品的质量和性能严重依赖于个人的能力和行为。

可重复级 (repeatable)

- n 管理制度化，建立了基本的管理制度和规程，管理工作有章可循；
- n 初步实现标准化，开发工作较好地实施标准；
- n 变更均依法进行，做到基线化；
- n 稳定可跟踪，新项目的计划和管理基于过去的经验，具有重复以前成功项目的环境和条件。

软件过程成熟度模型

已定义级 (defined)

- n 开发过程，包括技术工作和管理工作，均已实现标准化、文档化；
- n 建立了完善的培训制度和专家评审制度；
- n 全部技术活动和管理活动均稳定实施；
- n 项目的质量、进度和费用均可控制；
- n 对项目进行中的过程、岗位和职责均有共同的理解。

已管理级 (managed)

- n 产品和过程已建立了定量的质量目标；
- n 过程中活动的生产率和质量是可度量的；
- n 已建立过程数据库；
- n 已实现项目产品和过程的控制；
- n 可预测过程 and 产品质量趋势，如预测偏差，实现及时纠正

软件过程成熟度模型

优化级（optimizing）

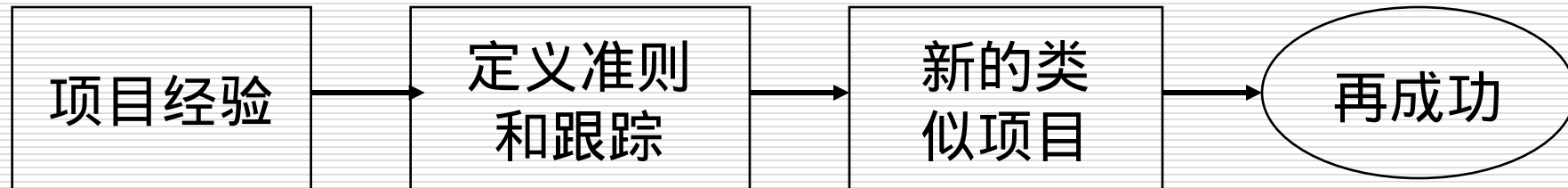
- n 可集中精力改进过程，采用新技术、新方法；
- n 拥有防止出现缺陷、识别薄弱环节以及加以改进的手段；
- n 可取得过程有效性的统计数据，并可据此进行分析，从而得出最佳方法。

五级成熟度的特性

等级1—初始级



等级2—重复级



五级成熟度的特性

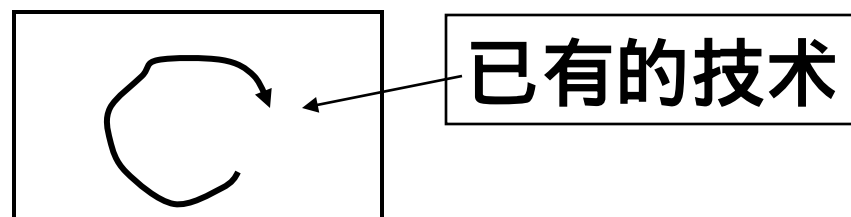
等级3—定义级



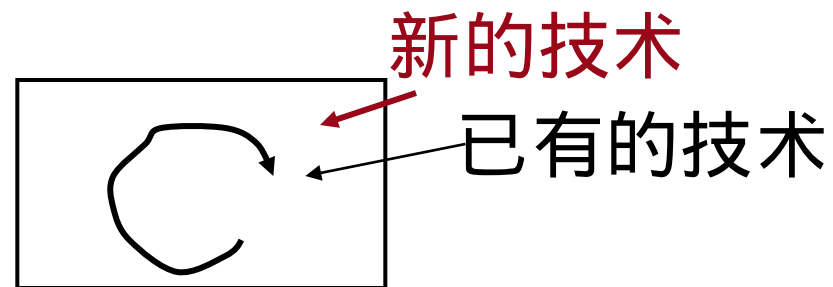
等级4—管理级



等级5—优化级



等级2—4中的改进



等级5中的优化



软件过程成熟度模型

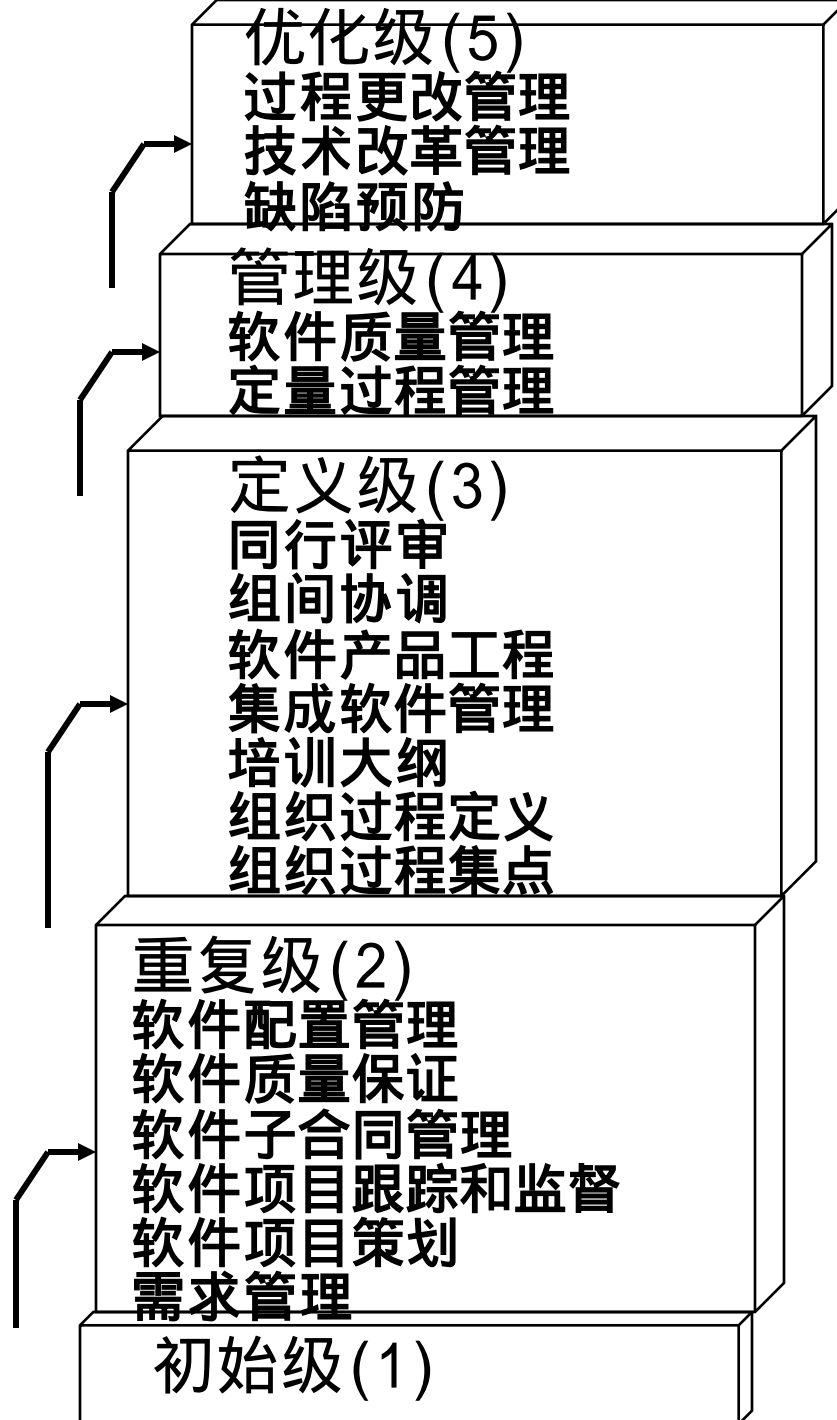
○ 关键过程领域

除去初始级以外，其他4级都有若干个引导软件机构改进软件过程的要点，称为关键过程领域（KPA：Key Process Area）。每一个关键过程领域是一组相关的活动。成功地完成这些活动，将会对提高过程能力起重要作用。

下图给出了各成熟度等级对应的关键过程领域。

关键过程域

每一个关键过程域
指出了改进过程能力的主要过程



软件过程成熟度模型

无论哪个KPA，它们的实施活动都统一按五个公共属性进行组织：

○ 目标

每一个KPA都确定了一组目标，若这组目标在每一个项目都能实现，则说明企业满足了该KPA的要求。若满足了一个级别的所有KPA要求，则表明达到了这个级别所要求的能力。

○ 实施能力

实施能力一般包括资源保证、人员培训等内容。它是企业实施KPA的前提条件。企业必须采取措施，在满足了这些条件后，才有可能执行KPA的活动。



软件过程成熟度模型

○ 执行活动

执行过程描述了执行KPA所需求的必要角色和步骤，一般包括计划、执行的任务、任务执行的跟踪等。

○ 度量分析

描述了过程的度量和度量分析要求。典型的度量和度量分析的要求是确定执行活动的状态和执行活动的有效性。

○ 实施验证

验证执行活动是否与建立的过程一致。实施验证涉及到管理的评审和审计以及质量保证活动。

软件过程成熟度模型

○ 如：KAP：需求管理 实践

软件过程成熟度模型

○ 成熟度提问单

为了把上述过程成熟度分级的方法推向实用化，需要为其提供具体的度量标尺。这个度量标尺就是成熟度提问单。

CMM在多个方面列出了大量的问题，每个问题都可针对特定的被评估软件机构给出肯定或否定的回答。提问单涉及的方面包括组织结构资源、人员及培训技术、管理文档化标准及工作步骤、过程度量数据管理和数据分析、过程控制。

下面列出了CMM2的几个KAP的成熟度提问单

CMM2成熟度提问单举例

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。
 - q 选择是，当实践被很好地建立和一致地执行。
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。
 - q 选择不是，当实践未被很好地建立或一致地执行。
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。
 - q 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。
例如，如果您不和子承包商一起工作，那么“软件子合同管理”的整个章节就可能不适用于该项目。
 - q 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。

CMM2成熟度提问单举例

一、需求管理

在客户和软件项目组之间建立对客户实际需求的共同理解，包括和客户一起建立和维护有关软件需求的协议，既包括技术需求也包括非技术需求（例如交付日期）。该协议构成软件生命周期中所有活动的基础（如预测、计划、实施、追踪、评测等）。如果客户需求有所变动，软件计划和实施也应做出相应调整，以求与需求保持一致。

CMM2成熟度提问单举例

一、需求管理

1. 是否用软件项目的需求来建立软件工程和管理的基准？
2. 当软件项目的需求改进变动时，是否对软件计划、产品和活动做出必要的调整？
3. 项目是否遵循软件组织所拟定的对项目需求的书面的管理规则？
4. 项目中负责管理需求的人员是否受到需求管理培训？
5. 是否用测量方式来确定需求管理活动的状态（例如，所提议的，未解决的，已批准的和已纳入基准的需求变动总数）？
6. 该项目需要管理活动是否受到软件质量保证的评审？

CMM2成熟度提问单举例

二、软件项目计划

为进行软件工程活动和软件项目管理所制定的合理的计划，包括预测、项目投入和工期，确定必要的承诺和执行等。

- 1.供计划和追踪软件项目的预测（例如，规模、成本和工期的预计）是否已文档化？
- 2.软件项目计划是否将准备实施的活动和对项目的承诺文档化？
- 3.所有相关的项目组及其成员对项目约定是否同意？
- 4.项目是否遵循软件组织用于项目计划的书面规则？
- 5.是否为项目计划准备了足够的资源（例如，资金、有经验的开发人员）？
- 6.是否用测量方式来确定项目计划活动的状态（例如，项目计划活动里程碑的完成情况与计划本身的比较）？
- 7.项目经理是否对软件项目计划活动进行定期的和事件驱动的审查？

CMM2成熟度提问单举例

三.软件项目追踪和监控

提供适当的对项目实际进展的信息，使管理者能在实施明显偏离计划时采取纠正措施。纠正措施包括修改软件开发计划以反映实际的完成情况，重新计划剩余工作或采取改进性能的措施。软件项目追踪和监控包括对文档化的预计，承诺和计划的评审，跟踪软件完成情况及其结果，以及在实际完成情况基础上的调整。

CMM2成熟度提问单举例

三.软件项目追踪和监控

- 1.是否比较了软件项目的实际结果（例如，规模、成本、进度）与计划中的预算？
- 2.当实际结果明显偏离计划时，是否采用纠正措施？
- 3.所有相关的项目组及其成员是否同意对项目承诺的更改？
- 4.项目是否遵循软件组织用于追踪和控制软件开发活动的书面规则？
- 5.项目组中是否有专人追踪软件产品和活动？（例如，预算、进度和工作量）
- 6.是否用测量方式来确定软件追踪和监控活动的状态（例如，在追踪和监控活动中所投入的总工作量）？
- 7.高层管理是否定期参与评审软件项目追踪和监控的活动（例如，项目性能、未解决的问题、风险和行动指导）？

CMM2成熟度提问单举例

四、子合同管理

选择合格的软件方承包商并有效地对它们进行管理，包括如何选择软件分包商，如何建立与分包商的约定，如何追踪和评审分包商的功效。这些实践包括对软件子合同的管理，也包括对子合同的构成成分的管理，如子合同中含有的软件硬件及其他系统成分的管理。

CMM2成熟度提问单举例

四、子合同管理

1. 是否按照文档化的规则来针对分包商完成项目的能力挑选软件项目子承包商？
2. 子合同的变动是否得到主承包商和子承包商双方的同意？
3. 是否与子承包商进行定期的技术交流？
4. 是否根据约定追踪子承包商的工作效能和结果？
5. 项目是否遵循软件组织管理制定的管理软件子合同的书面规则？
6. 负责管理软件子合同的人员是否经过软件子合同管理的培训？
7. 是否用测量方式来确定软件子合同管理活动的状态（例如，参照交付日期计划的进度状态以及在子合同管理上投入的工作量）？
8. 项目经理是否参与对软件子合同活动的定期的和事件驱动的评审工作？

CMM2成熟度提问单举例

五、软件质量保证

向管理者提供对软件项目所采纳的过程和所开发的产品质量信息，包括复查和审核软件产品及活动以验证它们符合试用的标准及规则，也包括向项目经理和其他相关人员提供审核数据和结果。

CMM2成熟度提问单举例

五、软件质量保证

1. 是否对软件质量保证活动作好计划？
2. 软件质量保证是否针对软件产品和活动符合试用标准、规则的情况提供了客观的验证？
3. 软件质量保证的复查和审核结果是否提供给相关的项目组及其成员（例如，负责该项目工作的管理人员和技术人员）？
4. 如有项目组不能解决的与拟定过程不符合的问题，是否交由高级管理层解决（例如，偏离适当的标准）？
5. 项目是否遵循软件组织实施软件质量保证的书面规则？
6. 是否为软件质量保证活动准备了足够的资源（例如，资金和专门负责处理过程不符合情况的经理）？
7. 是否用测量方式来确定软件质量保证活动的成本和进度状况（例如，已完成工作，投入的工作量，资金与计划的比较）？
8. 高层管理是否定期参与对软件质量保证活动的评审？

CMM2成熟度提问单举例

六、软件配置管理

建立和维护在项目的整个生命周期内软件产品的完整性，包括指明在特定时段上软件的配置（即选定的软件产品及其描述）系统的控制对配置的变动，并在整个软件生命周期内保持配置的完整性和可追踪性。软件配置管理所含的产品包括最终交付给客户的产品，以及与这些产品一起标明的事项或开发这些产品所必须的事项（如硬件、系统等）

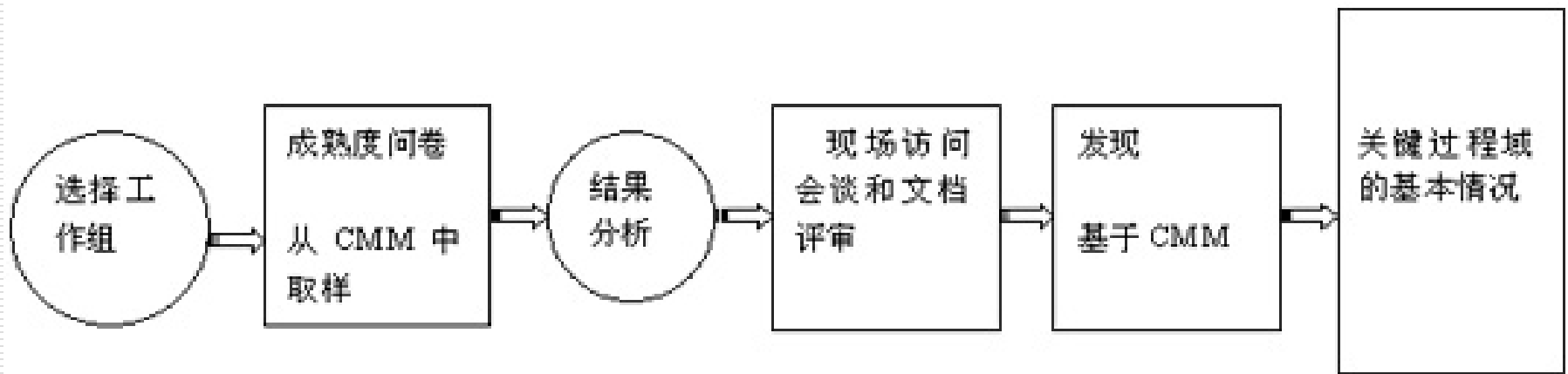
CMM2成熟度提问单举例

六、软件配置管理

1. 是否拟定对项目软件配置管理活动的计划？
2. 通过配置管理，项目是否已经对软件产品进行标明、控制并使其可用？
3. 项目是否遵循一套文档化的规则，对配置事项或配置单元的变动进行控制？
4. 是否把关于软件基准（即经过正式评审及认定的软件配置事项，它们此后可作为进一步开发的基础并只有通过正式的更改程序才能被变动）的标准报告分发给相关的项目组及成员（此类报告包括软件配置控制组会议记录，变动申请汇报，状态报告）？
5. 项目是否遵循软件组织如何实施软件配置管理活动的书面规则？
6. 项目组成员是否经过专门培训使其能完成所负责的软件配置管理活动？
7. 是否用测量方式来确定软件配置管理活动的状况（例如，为软件配置管理活动所投入的工作量和金钱）？
8. 是否进行定期审核以验证软件基准同定义基准的文档相符合（例如，由软件质量保证小组定义的文档）？

软件过程成熟度模型

- 利用CMM对软件机构进行成熟度评估
CMM为评估和评价提供了一个参考框架



ISO 9000和CMM的联系

- ISO 9001和CMM既有区别又相互联系，两者不可简单的互相替代。

尽管ISO 9001标准的一些要求在CMM中不存在，而CMM的一些要求在ISO 9001标准中也不存在，但不可否认，两者之间的关系非常密切。当然，两者之间的差别也很明显。

两者的最大相似之处在于两者都强调“该说的要说到，说到的要做到”。对每一个重要的过程应形成文件，包括指导书和说明，并检查交货质量水平。两者都强调持续改进。

很明显，取得ISO 9001认证对于取得CMM的等级证书是有益的，反之，取得CMM等级证书，对于寻求ISO 9001认证也是有帮助的。

ISO 9000和CMM的联系

○ CMM比ISO9000好 ？

CMM和ISO9000都只是一种标准,前者是根据一个研究所的研究报告制订的一套标准,有很多不完善的地方。ISO9000是一个国际标准,通用性强。就软件企业来说,两者在对企业的质量管理的指导原则上,要求差不多,而实现和表述上不同,不能笼统的说谁比谁好。

CMM注重过程要求,它的每一级对所要实现的关键过程域都有详细的要求,并且强制企业能自我更新和持续改进,以实现缺陷预防。这对于推动软件企业自身质量管理素质是非常有利的。

ISO9000的通用性太强,针对性太弱。ISO9000适用范围是所有设计\制造\开发及服务的行业,对软件质量管理体系的要求很低,而且也不具体。

ISO9000其他

○ 取得ISO 9001认证并不意味着完全满足CMM某个等级的要求

表面上看，获得ISO 9001标准的企业应有CMM第3至第4级的水平，但事实上，有些获得CMM1级的企业也获得了ISO 9001证书，原因是ISO 9001强调以顾客的要求为出发点，不同的顾客要求的质量水平也不同，而且各个审核员的水平/解释也有些差异；取得ISO 9001认证所代表的质量管理和质量保证能力的高低与审核员对标准的理解及自身水平的高低有很大的关系，而这并不是ISO 9001标准本身所决定的

ISO 9001标准只是质量管理体系的最低可接受准则，不能说已满足CMM的大部分要求。有一点可以肯定，ISO 9001认证合格的企业至少能满足CMM第2级的大部分要求以及第3级的一部分要求。

ISO9000其他

- 取得CMM第2级(或第3级)不能笼统的谈可以满足ISO 9001的要求

CMM 第2级的所有关键过程都涉及ISO 9001的要求，但都低于ISO 9001的要求。另外，一些CMM第1级的组织在满足了第2级和第3级的一些关键过程的要求后，也可以获得ISO 9001认证证书。当然，尽管CMM没有完全满足ISO 9001标准的一些特定要求，但包含了大部分的要求。

CMM是专门针对软件开发企业设计的，因此在针对性上比ISO 9001要好。

ISO和CMM的适用性

○ 管理水平的适用性

一个公司先基于 ISO9000 质量管理体系建立起质量管理体系框架，培养组织的质量意识。然后在此基础上，选择若干过程域进行重点监控，以逐步达到 CMM 某个成熟度等级的要求。

○ 复杂度的适用性

在一个研发过程本身复杂度不是很高，但组织复杂度由于组织本身的发展变得庞杂的组织中推行基于 ISO 的质量管理体系；
在一个组织机构相对完善，由于项目规模本身的扩大而迫切需要引入适当的过程监控的组织实施基于 CMM 的质量管理体系。

ISO和CMM的适用性

○ 量化管理的适用性

在具有一定质量管理基础的组织内，可以参照 CMM/SMMI-SW 相关 KPA/PA 的要求，建立自己的研发过程度量体系，逐步推进量化管理。

有关CMM

- CMM是一个学术报告基础上建立起来的一套评估体系，只适用于美国，而且它的认证结果只是由SEI授权的首席评估员寄一封带有本人签名的信给你，并在SEI备案，没有任何证书，终生受用，中间不再审查。由于SEI不是政府部门，这对中国企业来说，有些不可想象。
- 通过ISO9000认证的企业，要在中国技术监督局备案，并且发证给企业，并要求每年审查，所有参加多边认可协议的国家必须认可，适用性强。
- ISO9000认证费用是2-4万元人民币，时间4-6个月，而CMM则需要7-10万美金，每一级向上一级认证时，需要18-30个月。
- 2003年CMM标准作废，取而代之的是CMMI（Capability Maturity Model Integration 能力成熟度模型集成）。

有关CMM

○ 综述：

CMM对于推动企业自身质量管理是有好处的，它的评估标准也是很适合软件企业自身发展的，对于软件产品不出口到美国的企业，不需要参与此认证。

ISO9000：2000只是能够证明软件企业的质量管理体系能够保障产品质量，是一个最低质量保证要求，对软件企业的针对性不强，ISO也正准备发布以过程为关注焦点的软件质量体系标准ISO15504，也叫SPICE。它与CMMI很相似，但SPICE是国际通用标准，要比CMMI更有说服力，CMMI也认识到这一点，把表达形式有单一的阶段性表达形式拆成连续性和阶段性两种，便于从SPICE转到CMMI。



ISO/IEC15504

○ ISO/IEC15504

1994年SPICE(Software Process Improvement and Capability dTermination, 软件过程改进和能力测定)项目组织对“确定自我能力改造和进行软件供应商能力”的一些标准展开大规模试验, 收集试验数据。

1998年10月发布了《ISO/IEC 15504 TR 系列技术报告》。

- 1) ISO/IEC TR 15504-1:1998 信息技术- 软件过程评估- 第一部分: 概念和介绍性指南;
- 2) ISO/IEC TR 15504-2:1998 信息技术- 软件过程评估- 第二部分: 过程和过程能力的参考模型;
- 3) ISO/IEC TR 15504-3:1998 信息技术- 软件过程评估- 第三部分: 实施评估;

ISO/IEC15504

- 4) ISO/IEC TR 15504-4:1998 信息技术- 软件过程评估- 第四部分：实施和指标指南；
- 5) ISO/IEC TR 15504-5:1998 信息技术- 软件过程评估- 第五部分：过程评估模型；
- 6) ISO/IEC TR 15504-6:1998 信息技术- 软件过程评估- 第六部分：评估员资格指南
- 7) ISO/IEC TR 15504-7: 1998 信息技术- 软件过程评估- 第七部分：用于过程改进指南；
- 8) ISO/IEC TR 15504-8:1998 信息技术- 软件过程评估- 第八部分：确定供应者过程能力应用指南；
- 9)ISO/IEC TR 15504-9:1998 信息技术- 软件过程评估- 第九部分：词汇

ISO/IEC15504

2003年10月后，陆续正式发布多个标准。该标准覆盖了过程评估、过程改进和过程能力确认等，为相应工作提供了指南和模型。其框架与ISO9001:2000相兼容。

参考：

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_tc_browse.htm?commid=45086

n [ISO/IEC 15504-1:2004](#) Information technology -- Process assessment -- Part 1: Concepts and vocabulary

n [ISO/IEC 15504-2:2003](#) Information technology -- Process assessment -- Part 2: Performing an assessment

ISO/IEC15504

- n [ISO/IEC 15504-3:2004](#) Information technology -- Process assessment -- Part 3: Guidance on performing an assessment
- n [ISO/IEC 15504-4:2004](#) Information technology -- Process assessment -- Part 4: Guidance on use for process improvement and process capability determination
- n [ISO/IEC 15504-5:2006](#) Information technology -- Process Assessment -- Part 5: An exemplar Process Assessment Model

ISO/IEC15504

- [ISO/IEC TR 15504-6:2008](#) Information technology -- Process assessment -- Part 6: An exemplar system life cycle process assessment model
- [ISO/IEC TR 15504-7:2008](#) Information technology -- Process assessment -- Part 7: Assessment of organizational maturity
- [ISO/IEC PDTR 15504-8](#) Information technology -- Software process assessment -- Part 8: An exemplar process assessment model for IT service management
- [ISO/IEC DTR 15504-9](#) Information technology -- Software process assessment -- Part 9: Capability Target Profiles

ISO/IEC15504

○ [ISO/IEC PDTR 15504-10](#)

Information technology -- Software process assessment -- Part 10: Safety Extensions

阶段	国际标 准	補篇	快速	技术报 告	国际标 准化轮 廓	技术勘 误
阶段0 预阶段						
阶段1- 提案阶	NP	NP		NP	NP	
阶段2- 准备阶	WD	WD		WD	WD	缺陷报 告
阶段3-委 员会阶	CD FCD	PDAM FPDAM		PDTR	PDISP FPDISP	DCOR
阶段4-批 准阶段	FDIS	FDAM	DIS	DTR	FDISP	
阶段5-出 版阶段	IS	AMD	IS	TR	ISP	COR



ISO/IEC15504

http://www.iso.ch/iso/iso_catalogue/catalogue_tc/catalogue_tc_browse.htm?commid=45086

软件文件标准与规范

GB/T 8567-1988《计算机软件产品开发文件编制指南》是一份国家标准指导性文件。它建议在软件的开发过程中编制下述14个文件。即：

可行性研究报告

操作手册

项目开发计划

模块开发卷宗

软件需求说明书

测试计划

数据要求说明书

测试分析报告

总体设计说明书

开发进度表

详细设计说明书

项目开发总结。

数据库设计说明书

用户手册

该指南给出了这14个文件的编制提示，它同时也是这14个文件编写质量的检验准则。

软件文件标准与规范

GB/T 8567-1988版是参考英国某公司的文档标准，结合当时国内软件开发的经验，而且主要是针对瀑布模型的开发方法而制定的。

该标准的发布与实施对我国上世纪八十年代、九十年代的软件开发发挥了重要作用。但随着时间的推移，软件工程技术的发展与提高，目前来看，88版已经不适应软件产业发展的需要，因此推出了修订版GB/T 8567-2006《计算机软件文档编制规范》。

GB/T 8567新老版本的主要差异

- GB/T 8567 - 1988主要适用于瀑布模型开发方法
- GB/T 8567 - 1988给出了14种文档的编制格式要求
- GB/T 8567-2006原则上适用于各种类型的开发方法
- GB/T 8567 - 2006描述了文档编制过程
- GB/T 8567 - 2006给出25种文档的编制格式要求
 - 1) 可行性分析 (研究) 报告
 - 2) 软件开发计划
 - 3) 软件测试计划
 - 4) 软件安装计划

GB/T 8567 - 2006给出 25 种文档

- 5) 软件移交计划
- 6) 运行概念说明
- 7) 系统 / 子系统需求规格说明
- 8) 接口需求规格说明
- 9) 系统 / 子系统设计 (结构设计) 说明
- 10) 接口设计说明
- 11) 软件需求规格说明
- 12) 数据需求说明
- 13) 软件 (结构) 设计说明
- 14) 数据库 (顶层) 设计说明
- 15) 软件测试说明

GB/T 8567 - 2006给出 25 种文档

- 16) 软件测试报告
- 17) 软件配置管理计划
- 18) 软件质量保证计划
- 19) 开发进度月报
- 20) 项目开发总结报告
- 21) 软件产品规格说明
- 22) 软件版本说明
- 23) 软件用户手册
- 24) 计算机操作手册
- 25) 计算机编程手册

○ 另外给出了面向对象的 10 种文档的编制格式要求

GB/T 8567 - 2006标准结构

- 1、范围
- 2、规范性引用文件
- 3、术语和定义
- 4、缩略语
- 5、文档（编制）过程
- 6、文档编制要求
- 7、文档编制格式
- 附录 A 面向对象软件的文档编制

GB/T 8567 - 2006 文档（编制）过程

○ 概述

有两种主要类型的标准：

- a. 产品标准，它规定产品的特征和功能需求；
- b. 过程标准，它规定开发产品的过程。

GB/T 8567 - 2006 文档（编制）过程

○ 文档（编制）过程的关注点

文档编制计划

计划规定在文档创建中要执行的工作。此文档计划应经需方正式同意，以预示它完全覆盖了需方的要求。

文档开发（编制）

按文档计划规定进行文档开发。通常，在进行文档开发前，要规定文档的格式（风格）。在软件的开发和管理过程中需要那些文档，每种文档的规范说明。

文档评审

规定文档评审的要求和相关活动。

GB/T 8567 - 2006文档编制要求

○ 软件生存周期与各种文档的编制

计算机软件的生存周期中，一般地说，应该产生以下一些基本文档。

可行性分析（研究）报告；

软件（或项目）开发计划；

软件需求规格说明；

接口需求规格说明；

系统／子系统设计（结构设计）说明；

软件（结构）设计说明；

接口设计说明；

数据库（顶层）设计说明；

（软件）用户手册；

操作手册；

测试计划；

测试报告；

软件配置管理计划；

软件质量保证计划；

开发进度月报；

项目开发总结报告；

软件产品规格说明；

软件版本说明等。

GB/T 8567 - 2006文档编制要求

对于使用文档的人员而言他们所关心的文件的种类随他们所承担的工作而异。

管理人员：

- 可行性分析（研究）报告
- 项目开发计划
- 软件配置管理计划
- 软件质量保证计划
- 开发进度月报
- 项目开发总结报告

开发人员：

- 可行性分析（研究）报告
- 项目开发计划
- 软件需求规格说明
- 接口需求规格说明
- 软件（结构）设计说明
- 接口设计说明书
- 数据库（顶层）设计说明
- 测试计划
- 测试报告；

GB/T 8567 - 2006文档编制要求

维护人员：

软件需求规格说明
接口需求规格说明
软件（结构）设计说明
测试报告

用 户：

软件产品规格说明
软件版本说明
用户手册
操作手册

GB/T 8567 - 2006文档编制要求

○ 文档编制中的考虑因素

- n 文档的读者：文档的作者必须了解自己的读者，适应读者的水平、特点和要求。
- n 重复性：必要的重复是为了方便每种文档各自的读者，每种文档应自成体系，尽量避免读一种文档时不得不去参考另一篇文档。
- n 灵活性：由于软件不同，允许编制文档有一定的灵活性，文档的编制也不全同(如，文档种类、详细程度、扩展、表现形式等)。

GB/T 8567 - 2006文档编制格式

- 可行性分析（研究）报告（FAR）浏览
- 软件开发计划（SDP）浏览
- 软件需求规格说明（SRS）详述

可行性分析（研究）报告（FAR）

1. 《可行性分析（研究）报告》（FAR）它是项目初期策划的结果，它分析了项目的要求、目标和环境；提出了几种可供选择的方案；并从技术、经济和法律各方面进行了可行性分析。可作为项目决策的依据。
2. FAR也可以作为项目建议书、投标书等文件的基础。

1 引言	5.4 影响（或要求）
1.1 标识	5.4.1 设备
1.2 背景	5.4.2 软件
1.3 项目概述	5.4.3 运行
1.4 文档概述	5.4.4 开发
2 引用文档	5.4.5 环境
3 可行性分析的前提	5.4.6 经费
3.1 项目的要求	5.5 局限性
3.2 项目的目标	6 经济可行性（成本 - 效益分析）
3.3 项目的环境、条件、假定和限制	6.1 投资
3.4 进行可行性分析的方法	6.2 预期的经济效益
4 可选的方案	6.2.1 一次性收益
4.1 原有方案的优缺点、局限性及存在的问题	6.2.2 非一次性收益
4.2 可重用的系统，与要求之间的差距	6.2.3 不可定量的收益
4.3 可选择的系统方案1	6.2.4 收益 / 投资比
4.4 可选择的系统方案2	6.2.5 投资回收周期
...	6.3 市场预测
4.5 选择最终方案的准则。	7 技术可行性（技术风险评价）
5 所建议的系统	8 法律可行性
5.1 对所建议的系统的说明	9 用户使用可行性
5.2 数据流程和处理流程	10 其它与项目有关的问题
5.3 与原系统的比较（若有原系统）	11 注解

软件开发计划（SDP）

1. 《软件开发计划》（SDP）描述开发者实施软件开发工作的计划，本文档中“软件开发”一词涵盖了新开发、修改、重用、再工程、维护和由软件产品引起的其它所有的活动。
2. SDP是向需求方提供了解和监督软件开发过程、所使用的方法、每项活动的途径、项目的安排、组织及资源的一种手段。
3. 本计划的某些部分可视实际需要单独编制成册，例如，软件配置管理计划、软件质量保证计划和文档编制计划等。

1 引言

1.1 标识

1.2 系统概述

1.3 文档概述

1.4 与其它计划之间的关系

1.5 基线

2 引用文档

3 交付产品

3.1 程序

3.2 文档

3.3 服务

3.4 非移交产品

3.5 验收标准

3.6 最后交付期限

4 所需工作概述

5 实施整个软件开发活动的计划

5.1 软件开发过程

5.2 软件开发总体计划

6 实施详细软件开发活动的计划

6.1 项目计划和监督

6.2 建立软件开发环境

6.3 系统需求分析

6.4 系统设计

6.5 软件需求分析

6.6 软件设计

6.7 软件实现和配置项测试

6.8 配置项集成和测试

6.9 CSCI合格性测试

6.10 CSCI/HWCI集成和测试

6.11 系统合格性测试

6.12 软件使用准备

6.13 软件移交准备

6.14 软件配置管理

6.15 软件产品评估

6.16 软件质量保证

6.17 问题解决过程（更正活动）

6.18 联合评审（联合技术评审和联合管理评审）

6.19 文档编制

6.20 其它软件开发活动

7 进度表和活动网络图

8 项目组织和资源

8.1 项目组织

8.2 项目资源

9 培训

9.1 项目的技术要求

9.2 培训计划

10 项目估算

10.1 规模估算

10.2 工作量估算

10.3 成本估算

10.4 关键计算机资源估算

10.5 管理预留

11 风险管理

12 支持条件

12.1 计算机系统支持。

12.2 需要需方承担的工作和提供的条件。

12.3 需要分包商承担的工作和提供的条件。

13 注解

附录

软件需求规格说明（SRS）

1. 《软件需求规格说明》（SRS）描述对计算机软件配置项CSCI的需求，并确保每个要求得以满足的所使用的方法。涉及该CSCI外部接口的需求可在本SRS中给出：或在本SRS引用的一个或多个《接口需求规格说明》（IRS）中给出。
2. 这个SRS，可能还要用IRS加以补充，是CSCI设计与合格性测试的基础。

软件需求规格说明（SRS）

1 范围

本章应分为以下几条。

1.1 标识

本条应包含本文档适用的系统和软件的完整标识，（若适用）包括标识号、标题、缩略词语、版本号和发行号。

1.2 系统概述

本条应简述本文档适用的系统和软件的用途，它应描述系统和软件的一般特性；概述系统开发、运行和维护的历史；标识项目的投资方、需方、用户、开发方和支持机构；标识当前和计划的运行现场；列出其它有关的文档。

软件需求规格说明（SRS）

1.3 文档概述

本条应概述本文档的用途和内容，并描述与其使用有关的保密性或私密性要求。

1.4 基线

说明编写本系统设计说明书所依据的设计基线。

2 引用文档

本章应列出本文档引用的所有文档的编号、标题、修订版本和发行日期，也应标识不能通过正常的供货渠道获得的所有文档的来源。

软件需求规格说明（SRS）

3 需求

本章应分以下几条描述CSCI需求，也就是，构成CSCI验收条件的CSCI的特性。CSCI需求是为了满足分配给该CSCI的系统需求所形成的软件需求。给每个需求指定项目唯一标识符以支持测试和可追踪性。并以一种可以定义客观测试的方式来陈述需求。如果每个需求有关的合格性方法（见第4章）和对系统（若适用，子系统）需求的可追踪性（见5.a条）在相应的章中没有提供，则在此进行注解。

软件需求规格说明（SRS）

描述的详细程度遵循以下规则：应包含构成CSCI验收条件的那些CSCI特性，需方愿意推迟到设计时留给开发方说明的那些特性。如果在给定条中没有需求的话，本条应如实陈述。如果某个需求在多条中出现，可以只陈述一次而在其它条直接引用。

软件需求规格说明（SRS）

3.1 所需的状态和方式

如果需要CSCI在多种状态和方式下运行，且不同状态和方式具有不同的需求的话，则要标识和定义每一状态和方式，状态和方式的例子包括：空闲、准备就绪、活动、事后分析、培训、降级、紧急情况、后备等。状态和方式的区别是任意的，可以仅用状态描述CSCI，也可以仅用方式、方式中的状态、状态中的方式或其它有效方式描述。如果不需要多个状态和方式，不需人为加以区分，应如实陈述；如果需要多个状态或方式，还应使本规格说明中的每个需求或每组需求与这些状态和方式相关联，关联可在本条或本条引用的附录中用表格或其它的方法表示，也可在需求出现的地方加以注解。

软件需求规格说明（SRS）

3.2 需求概述

3.2.1 目标

- a. 本系统的开发意图、应用目标及作用范围（现有产品存在的问题和建议产品所要解决的问题）。
- b. 本系统的主要功能、处理流程、数据流程及简要说明。
- c. 表示外部接口和数据流的系统高层次图。说明本系统与其他相关产品的关系，是独立产品还是一个较大产品的组成部分（可用方框图说明）。

软件需求规格说明（SRS）

3.2.2 运行环境

简要说明本系统的运行环境（包括硬件环境和支持环境）的规定。

3.2.3 用户的特点

说明是那一种类型的用户，从使用系统来说，有些什么特点。

3.2.4 关键点

说明本软件需求规格说明书中的关键点（例如：关键功能、关键算法和所涉及的关键技术等）。

软件需求规格说明（SRS）

3.2.5 约束条件

列出进行本系统开发工作的约束条件。
例如：经费限制、开发期限和所采用的方法与技术，以及政治、社会、文化、法律等。

软件需求规格说明（SRS）

3.3 需求规格

3.3.1 软件系统总体功能 / 对象结构

对软件系统总体功能 / 对象结构进行描述，包括结构图、流程图或对象图。

3.3.2 软件子系统功能 / 对象结构

对每个主要子系统的基本功能模块 / 对象进行描述，包括结构图、流程图或对象图。

3.3.3 描述约定

通常使用的约定描述（数学符号、度量单位等）。

软件需求规格说明（SRS）

3.4 CSCI能力需求

本条应分条详细描述与CSCI每一能力相关联的需求。“能力”被定义为一组相关的需求。可以用“功能”、“性能”、“主题”、“目标”、或其它适合用来表示需求的词来替代“能力”。

软件需求规格说明（SRS）

3.4.x（CSCI能力）

本条应标识必需的每一个CSCI能力，并详细说明与该能力有关的需求。如果该能力可以更清晰地分解成若干子能力，则应分条对子能力进行说明。该需求应指出所需的CSCI行为，包括适用的参数，如响应时间、吞吐时间、其它时限约束、序列、精度、容量（大小/多少）、优先级别、连续运行需求、和基于运行条件的允许偏差：（若适用）需求还应包括在异常条件、非许可条件或越界条件下所需的行为，错误处理需求和任何为保证在紧急时刻运行的连续性而引入到CSCI中的规定。在确定与CSCI所接收的输入和CSCI所产生的输出有关的需求时，应考虑在本文3.5.x给出要考虑的主题列表。

软件需求规格说明（SRS）

对于每一类功能或者对于每一个功能，需要具体描写其输入、处理和输出的需求。

a. 说明

描述此功能要达到的目标、所采用的方法和技术，还应清楚说明功能意图的由来和背景。

b. 输入

包括：

- 1) 详细描述该功能的所有输入数据，如：输入源、数量、度量单位、时间设定和有效输入范围等。
- 2) 指明引用的接口说明或接口控制文件的参考资料。

软件需求规格说明（SRS）

c. 处理

定义对输入数据、中间参数进行处理以获得预期输出结果的全部操作。包括：

- 1) 输入数据的有效性检查。
- 2) 操作的顺序，包括事件的时间设定。
- 3) 异常情况的响应，例如，溢出、通信故障、错误处理等。
- 4) 受操作影响的参数。
- 5) 用于把输入转换成相应输出的方法。
- 6) 输出数据的有效性检查。

软件需求规格说明（SRS）

d. 输出

1) 详细说明该功能的所有输出数据，例如，输出目的地、数量、度量单位、时间关系、有效输出范围、非法值的处理、出错信息等。

2) 有关接口说明或接口控制文件的参考资料。

软件需求规格说明（SRS）

3.5 CSCI外部接口需求

本条应分条描述CSCI外部接口的需求。（如有）
本条可引用一个或多个接口需求规格说明（IRS）
或包含这些需求的其它文档。

外部接口需求，应分别说明：

- a. 用户接口；
- b. 硬件接口；
- c. 软件接口；
- d. 通信接口的需求。

软件需求规格说明（SRS）

3.5.1 接口标识和接口图

本条应标识所需的CSCI外部接口，也就是CSCI和与它共享数据、向它提供数据或与它交换数据的实体的关系。（若适用）每个接口标识应包括项目唯一标识符，并应用名称、序号、版本和引用文档指明接口的实体（系统、配置项、用户等）。该标识应说明哪些实体具有固定的接口特性（因而要对这些接口实体强加接口需求），哪些实体正被开发或修改（从而接口需求已施加给它们）。可用一个或多个接口图来描述这些接口。

软件需求规格说明（SRS）

3.5.x（接口的项目唯一标识符）

本条（从3.5.2开始）应通过项目唯一标识符标识CSCI的外部接口，简单地标识接口实体，根据需要可分条描述为实现该接口而强加于CSCI的需求。该接口所涉及的其它实体的接口特性应以假设或“当（未提到实体）这样做时，CSCI将……”的形式描述，而不描述为其它实体的需求。本条可引用其它文档（如：数据字典、通信协议标准、用户接口标准）代替在此所描述的信息。（若适用）需求应包括下列内容，它们以任何适合于需求的顺序提供，并从接口实体的角度说明这些特性的区别（如对数据元素的大小、频率或其它特性的不同期望）：

软件需求规格说明（SRS）

- a. CSCI必须分配给接口的优先级别；
- b. 要实现的接口的类型的需求（如：实时数据传送、数据的存储和检索等）；
- c. CSCI必须提供、存储、发送、访问、接收的单个数据元素的特性，如：
 - 1) 名称/标识符；
 - a) 项目唯一标识符；
 - b) 非技术（自然语言）名称；
 - c) 标准数据元素名称；
 - d) 技术名称（如代码或数据库中的变量或字段名称）；
 - e) 缩写名或同义名；

软件需求规格说明（SRS）

- 2) 数据类型（字母数字、整数等）；
- 3) 大小和格式（如：字符串的长度和标点符号）；
- 4) 计量单位（如：米、元、纳秒）；
- 5) 范围或可能值的枚举（如：0 ~ 99）；

软件需求规格说明（SRS）

6）准确度（正确程度）和精度（有效数字位数）；

7）优先级别、时序、频率、容量、序列和其它的约束条件，如：数据元素是否可被更新和业务规则是否适用；

8）保密性和私密性的约束；

9）来源（设置/发送实体）和接收者（使用/接收实体）；

软件需求规格说明（SRS）

- d. CSCI 必须提供、存储、发送、访问、接收的数据元素集合体（记录、消息、文件、显示和报表等）的特性，如：
 - 1) 名称/标识符；
 - a) 项目唯一标识符；
 - b) 非技术（自然语言）名称；
 - c) 技术名称（如代码或数据库的记录或数据结构）；
 - d) 缩写名或同义名；
 - 2) 数据元素集合体中的数据元素及其结构（编号、次序、分组）；
 - 3) 媒体（如盘）和媒体中数据元素/数据元素集合体的结构；

软件需求规格说明（SRS）

- 4) 显示和其它输出的视听特性（如：颜色、布局、字体、图标和其它显示元素、蜂鸣器以及亮度等）；
- 5) 数据元素集合体之间的关系。如排序 / 访问特性；
- 6) 优先级别、时序、频率、容量、序列和其它的约束条件，如：数据元素集合体是否可被修改和业务规则是否适用；
- 7) 保密性和私密性约束；
- 8) 来源（设置 / 发送实体）和接收者（使用 / 接收实体）；

软件需求规格说明（SRS）

e. CSCI必须为接口使用通信方法的特性。如：

- 1) 项目唯一标识符；
- 2) 通信链接 / 带宽 / 频率 / 媒体及其特性；
- 3) 消息格式化；
- 4) 流控制（如：序列编号和缓冲区分配）；
- 5) 数据传送速率，周期性/非周期性，传输间隔；
- 6) 路由、寻址、命名约定；
- 7) 传输服务，包括优先级别和等级；
- 8) 安全性/保密性/私密性方面的考虑，如：加密、用户鉴别、隔离和审核等；

软件需求规格说明（SRS）

- f. CSCI必须为接口使用协议的特性，如：
 - 1) 项目唯一标识符；
 - 2) 协议的优先级别/层次；
 - 3) 分组，包括分段和重组、路由和寻址；
 - 4) 合法性检查、错误控制和恢复过程；
 - 5) 同步，包括连接的建立、维护和终止；
 - 6) 状态、标识、任何其它的报告特征；
- g. 其它所需的特性，如：接口实体的物理兼容性（尺寸、容限、负荷、电压和接插件兼容性等）。

软件需求规格说明（SRS）

3.6 CSCI内部接口需求

本条应指明CSCI内部接口的需求（如有的话）。如果所有内部接口都留待设计时决定，则需在此说明这一事实。如果要强加这种需求，则可考虑本文档的3.5给出的一个主题列表。

3.7 CSCI内部数据需求

本条应指明对CSCI内部数据的需求，（若有）包括对CSCI中数据库和数据文件的需求。如果所有有关内部数据的决策都留待设计时决定，则需在此说明这一事实。如果要强加这种需求，则可考虑在本文档的3.5.x.c和3.5.x.d给出的一个主题列表。

软件需求规格说明（SRS）

3.8 适应性需求

（若有）本条应指明要求CSCI提供的、依赖于安装的数据有关的需求（如：依赖现场的经纬度）和要求CSCI使用的、根据运行需要进行变化的运行参数（如：表示与运行有关的目标常量或数据记录的参数）。

3.9 保密性需求

（若有）本条应描述有关防止对人员、财产、环境产生潜在的危险或把此类危险减少到最低的CSCI需求，包括：为防止意外动作（如意外地发出“自动导航关闭”命令）和无效动作（发出一个想要的“自动导航关闭”命令时失败）CSCI必须提供的安全措施。

软件需求规格说明（SRS）

3.10 保密性和私密性需求

（若有）本条应指明保密性和私密性的CSCI需求，包括：CSCI运行的保密性/私密性环境、提供的保密性或私密性的类型和程度、CSCI必须经受的保密性/私密性的风险、减少此类危险所需的安全措施、CSCI必须遵循的保密性/私密性政策、CSCI必须提供的保密性/私密性审核、保密性/私密性必须遵循的确证/认可准则。

3.11 CSCI环境需求

（若有）本条应指明有关CSCI必须运行的环境的需求。例如，包括用于CSCI运行的计算机硬件和操作系统（其它有关计算机资源方面的需求在下条中描述）。

软件需求规格说明（SRS）

3.12 计算机资源需求

本条应分以下各条进行描述。

3.12.1 计算机硬件需求

本条应描述CSCI使用的计算机硬件需求，（若适用）包括：各类设备的数量、处理器、存储器、输入/输出设备、辅助存储器、通信/网络设备和其它所需的设备的类型、大小、容量及其它所要求的特征。

软件需求规格说明（SRS）

3.12.2 计算机硬件资源利用需求

本条应描述CSCI计算机硬件资源利用方面的需求，如：最大许可使用的处理器能力、存储器容量、输入/输出设备能力、辅助存储器容量、通信/网络设备能力。描述（如每个计算机硬件资源能力的百分比）还包括测量资源利用的条件。

3.12.3 计算机软件需求

本条应描述CSCI必须使用或引入CSCI的计算机软件的需求，例如包括：操作系统、数据库管理系统、通信/网络软件、实用软件、输入和设备模拟器、测试软件、生产用软件。必须提供每个软件项的正确名称、版本、文档引用。

软件需求规格说明（SRS）

3.12.4 计算机通信需求

本条应描述CSCI必须使用的计算机通信方面的需求，例如包括：连接的地理位置、配置和网络拓扑结构、传输技术、数据传输速率、网关、要求的系统使用时间、传送/接收数据的类型和容量、传送/接收/响应的时间限制、数据的峰值、诊断功能。

软件需求规格说明（SRS）

3.13 软件质量因素

（若有）本条应描述合同中标识的或从更高层次规格说明派生出来的对CSCI的软件质量方面的需求，例如包括有关CSCI的功能性（实现全部所需功能的能力）、可靠性（产生正确、一致结果的能力）、可维护性（易于更正的能力）、可用性（需要时进行访问和操作的能力）、灵活性（易于适应需求变化的能力）、可移植性（易于修改以适应新环境的能力）、可重用性（可被多个应用使用的能力）、可测试性（易于充分测试的能力）、易用性（易于学习和使用的能力）以及其它属性的定量需求。

软件需求规格说明（SRS）

3.14 设计和实现的约束.

（若有）本条应描述约束CSCI设计和实现的那些需求。这些需求可引用适当的商用标准和规范。例如需求包括：

- a. 特殊CSCI体系结构的使用或体系结构方面的需求，例如：需要的数据库和其它软件配置项；标准部件、现有的部件的使用；政府 / 需方提供的资源（设备、信息、软件）的使用；
- b. 特殊设计或实现标准的使用；特殊数据标准的使用；特殊编程语言的使用；
- c. 为支持在技术、威胁或任务等方面预期的增长和变更区域，必须提供的灵活性和可扩展性。

软件需求规格说明（SRS）

3.15 数据

说明本系统的输入、输出数据及数据管理能力方面的要求（处理量、数据量）。

3.16 操作

说明本系统在常规操作、特殊操作以及初始化操作、恢复操作等方面的要求。

软件需求规格说明（SRS）

3.17 故障处理

说明本系统在发生可能的软硬件故障时，对故障处理的要求。

- a. 说明属于软件系统的问题；
- b. 给出发生错误时的错误信息；
- c. 说明发生错误时可能采取的补救措施。

3.18 算法说明

用于实施系统计算功能的公式和算法的描述。

- a. 每个主要算法的概况
- b. 用于每个主要算法的详细公式

软件需求规格说明（SRS）

3.19 有关人员需求

（若有）本条应描述与使用或支持CSCI的人员有关的需求，包括人员数量、技能等级、责任期、培训需求、其它的信息。如：同时存在的用户数量的需求，内在帮助和培训能力的需求：（若有）还应包括强加于CSCI的人力行为工程需求，这些需求包括对人员在能力与局限性方面的考虑：在正常和极端条件下可预测的人为错误：人为错误造成严重影响的特定区域，例如包括错误消息的颜色和持续时间、关键指示器或关键的物理位置以及听觉信号的使用的需求。

软件需求规格说明（SRS）

3.20 有关培训需求

（若有）本条应描述有关培训方面的CSCI需求。包括：在CSCI中包含的培训软件。

3.21 有关后勤需求

（若有）本条应描述有关后勤方面的CSCI需求，包括：系统维护、软件支持、系统运输方式、供应系统的需求、对现有设施的影响、对现有设备的影响。

软件需求规格说明（SRS）

3.22 其它需求

（若有）本条应描述在以上各条中没有涉及到的其它CSCI需求。

3.23 包装需求

（若有）本条应描述需交付的CSCI在包装、加标签和处理方面的需求（如用确定方式标记和包装8磁道磁带的交付）。（若适用）可引用适当的规范和标准。

软件需求规格说明（SRS）

3.24 需求的优先次序和关键程度

（若适用）本条应给出本规格说明中需求的、表明其相对重要程度的优先顺序、关键程度或赋予的权值，如：标识出那些认为对安全性、保密性或私密性起关键作用的需求，以便进行特殊的处理。如果所有需求具有相同的权值，本条应如实陈述。

软件需求规格说明（SRS）

4 合格性规定

本章定义一组合格性方法，对于第3章中每个需求，指定所使用的方法，以确保需求得到满足。可以用表格形式表示该信息，也可以在第3章的每个需求中注明要使用的方法。合格性方法包括：

- a. 演示：运行依赖于可见的功能操作的CSCI或部分CSCI，不需要使用仪器、专用测试设备或进行事后分析；
- b. 测试：使用仪器或其它专用测试设备运行CSCI或部分CSCI，以便采集数据供事后分析使用；

软件需求规格说明（SRS）

- c. 分析：对从其它合格性方法中获得的积累数据进行处理，例如测试结果的归约、解释或推断；
- d. 审查：对CSCI代码、文档等进行可视化检查；
- e. 特殊的合格性方法。任何应用到CSCI的特殊合格性方法，如：专用工具、技术、过程、设施、验收限制。

软件需求规格说明（SRS）

5 需求可追踪性

本条应包括：

- a. 从本规格说明中每个CSCI的需求到其所涉及的系统（或子系统）需求的可追踪性。（该可追踪性也可以通过第3章中的每个需求进行注释的方法加以描述）
 - 1 注：每一层次的系统细化可能导致对更高层次的需求不能直接进行追踪。例如：建立多个CSCI的系统体系结构设计可能会产生有关CSCI之间接口需求，而这些接口需求在系统需求中并没有被覆盖，这样的需求可以被追踪到诸如“系统实现”这样的一般需求，或被追踪到导致它们产生的系统设计决策上。
- b. 从分配到被本规格说明中的CSCI的每个系统（或子系统）需求到涉及它的CSCI需求的可追踪性。分配到CSCI的所有系统（或子系统）需求应加以说明。追踪到IRS中所包含的CSCI需求可引用IRS。

软件需求规格说明（SRS）

6. 尚未解决的问题

如需要，可说明软件需求中的尚未解决的遗留问题。

7 注解

本章应包含有助于理解本文档的一般信息（例如背景信息、词汇表、原理）。本章应包含为理解本文档需要的术语和定义并给出解释，所有缩略词语和它们在文档中的含义的字母序列表。

软件需求规格说明（SRS）

附录

附录可用来提供那些为便于文档维护而单独出版的信息（例如图表、分类数据）。（若适用）在提供资料的文档主体部分应当引用附录a为便于处理，附录可单独装订成册。附录应按字母顺序（A，B等）编排。



软件质量管理实践

- 建立质量管理体系的方法
- 软件质量管理实践
- 软件生命周期质量管理

软件质量管理实践

- 建立质量管理体系的方法
 - n 软件组织进行质量管理标准选择依据
 - 中小型组织 ISO9000 或 CMM2
 - 大型组织 CMM3或以上

软件质量管理实践

n 建立和实施质量管理体系的十三个步骤

1. 确定顾客的需求和期望；
2. 建立组织的质量方针和目标；
3. 确定过程和职责；
4. 确定过程有效性的测量方法
5. 确定测定现行过程的有效性；
6. 确定防止不合格的措施；
7. 寻找改进机会；
8. 确定改进方向；
9. 改进策划；
10. 实施改进；
11. 监控改进效果；
12. 评价结果；
13. 评审改进措施和确定后续措施

软件质量管理实践

○ 软件质量管理实践（CMM2）

- n 几个概念
- n 需求管理
- n 软件项目策划
- n 软件项目跟踪与监督
- n 软件子合同管理
- n 软件质量保证
- n 软件配置管理

软件质量管理实践

几个概念

- 系统需求：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”；
- 分配需求：分配给软件的系统需求称为“分配需求”。
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。
- 基线：已经通过正式评审和批准的某规约或产品，它因此可以作为进一步开发的基础，并且只能通过正式的变更控制规程被改变。
- 约定：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：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）。

软件质量管理实践-需求管理

- 概述
- 对应的ISO9000的要求
- CMM的要求
- 实践过程
 - n 过程目标
 - n 组织方针
 - n 执行能力
 - n 执行活动
 - n 验证实施

软件质量管理实践-需求管理

○ 概述

- n 需求可划分为需求开发和需求管理。

需求开发包括需求获取、需求分析、编写需求规格定义和需求验证4个阶段。

需求管理是对需求开发过程的管理，即需求确认、需求评审、需求变更、需求跟踪。

- n 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。

软件质量管理实践-需求管理

○ ISO9000的要求

相关章节有：ISO9001:2000中，7.2 与顾客有关的过程(7.2.1 与产品有关的要求的确定,7.2.2 与产品有关的要求的评审,7.2.3 顾客沟通)

软件质量管理实践-需求管理

○ CMM的要求

需求管理是CMM二级中的第一个关键域

协商确认客户需求包括确认非技术需求、技术需求文档及评审分配需求：

- n 非技术需求：一般在协议、条件和合同条款中描述，包括提交产品、提交日期和里程碑等内容；
- n 技术需求：描述了系统的功能需求、性能需求、设计约束、编程语言和界面需求等多方面内容；
- n 评审分配需求：目的在于：1）确定分配需求用软件来实现是可行的、适当的；2）需求被清晰、正确地描述；3）需求是一致的、相互不矛盾的；4）需求是可测试的；
- n 其他

软件质量管理实践-需求管理

○ 实践过程

n 过程目标

1. 分配给软件的系统需求在受控的状态下，为软件工程和工程管理过程建立一个使用基线
2. 软件计划、产品和活动与分配给软件的系统需求始终保持一致。

软件质量管理实践-需求管理

○ 实践过程（续）

n 组织方针

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。该方针一般规定；

1)对分配需求建立文档。

2)由软件经理或其它受到影响的组评审分配需求。

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。

软件质量管理实践-需求管理

○ 实践过程（续）

n 执行能力

- 能力1：对每个项目，软件工作组工作的职责是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。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、和合同条款。如：要交付的产品，交付日期，和里程碑。
 - 2、对软件的技术需求。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。

软件质量管理实践-需求管理

○ 实践过程（续）

n 执行能力（续）

○ 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相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，应用领域等。

软件质量管理实践-需求管理

○ 实践过程（续）

n 执行的活动

○ 活动1：在分配需求被纳入软件项目之前，软件工程师组评审它们。

- 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：1）用软件来实现是可行的和恰当的，2）被清晰和正确地阐述，3）是相互一致的，和4）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：软件工程师组（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。

软件质量管理实践-需求管理

○ 实践过程（续）

n 执行的活动（续）

○ 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、被进行管理和控制。

“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）。

2、是软件开发计划的基础。

3、是制定软件需求的基础。

软件质量管理实践-需求管理

○ 实践过程（续）

n 执行的活动（续）

○ 活动3：评审对分配需求的更改，将其纳入软件项目。

1、评估它对现有约定的影响，合适时协商更改。

n 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。

n 和受到影响的组协商组织内部约定的更改。

2、对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。

软件质量管理实践-需求管理

○ 实践过程（续）

n 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。

- 验证2：项目经理既定期地又事件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，以便找到包括项目管理者监督评审的典型内容的实践。

软件质量管理实践-需求管理

○ 实践过程（续）

n 验证实施（续）

- 验证3：软件质量保证组评审和审计管理分配需求的活动和工作产品，并报告其结果。

至少，这些评审和审计要验证：

- 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。

软件质量管理实践 - 软件项目策划

- 概述
- ISO9000的要求
- CMM的要求
- 实践过程
 - n 过程目标
 - n 执行约定
 - n 执行能力
 - n 执行活动
 - n 验证实施

软件质量管理实践 - 软件项目策划

○ 概述

软件项目策划的目的是为完成和管理软件项目制定合理的计划。

由于项目的管理者是按照计划确定的内容对项目进行管理的，所以计划的合理性将直接关系到项目的成败。

策划方法

估计工作产品规模及所需资源，制定时间表，鉴别和评估软件风险和协商承诺

软件质量管理实践 - 软件项目策划

○ ISO9000的要求

在ISO9001:2000中，要求策划不但包括设计的开发，也包括过程的开发。

对项目策划有所要求的“7.3.1 设计和开发策划”如下：

组织应对产品的设计和开发进行策划和控制。在进行设计和开发策划时，组织应确定：

- a) 设计和开发阶段；
- b) 适合每个设计和开发阶段的评审、验证和确认活动；
- c) 设计和开发的职责和权限。

组织应对参与设计和开发的不同小组之间的接口实施管理，以确保有效的沟通，并明确职责分工。

策划的输出应随设计和开发的进展，在适当时予以更新。

软件质量管理实践 - 软件项目策划

○ CMM的要求

- n 项目的成本、资源、风险进行评估
- n 参照组织的项目开发计划模板，确定开发计划
- n 评审开发计划

软件质量管理实践 - 软件项目策划

○ 实践过程

n 过程目标

1. 对项目策划和跟踪软件项目用的软件估计已建立文档。
2. 软件项目的活动和约定是有计划的并已建立文档。
3. 受影响的组和个人同意他们的关于软件项目的约定。

项目策划的目的是建立和维护定义项目活动的项目计划,计划起源于需求,需求定义了产品和项目。这里提到的计划包括了生命周期过程中的所有的计划。

软件质量管理实践 - 软件项目策划

n 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：组织方针。该方针一般规定：
 - 1、将分配给软件的需求用作为策划软件项目的基础。
 - 2、在项目经理，项目软件经理，和其它的软件经理之间协商软件项目的约定。
 - 3、和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。
其它工程组有：系统工程组，硬件工程组，和系统测试组。
 - 4、受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。
受影响的组有；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。
 - 5、高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、对项目的软件开发计划进行管理和控制。

软件质量管理实践 - 软件项目策划

n 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、工作陈述包括：工作的范围，技术目标和对象，对顾客和最终用户的识别，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（包括顾客、子承包商、和合资伙伴）间的依赖关系，资源限制和目标，和对开发和维护的其它的约束和目标。
- 2、工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、对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、项目软件经理，直接工作或通过委托代表，协调项目的软件策划。

软件质量管理实践 - 软件项目策划

- 2、以可追踪，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：适当时，交付给外部顾客或最终用户的工作产品；供其它工程组使用的工作产品；和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、在可能处，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、使得支持软件项目策划的工具是可用的。工具有：电子表格程序，估计模型，和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。

软件质量管理实践 - 软件项目策划

n 执行活动

- 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，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：项目的技术目标和对象；系统和软件的技术解；软件预算、进度和资源；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。

软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。

软件质量管理实践 - 软件项目策划

- 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。
- 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、软件开发计划基于且遵守：顾客的标准(可能时)；项目的标准；经批准的工作陈述；和分配需求。
 - 2、与软件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、与其它软件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、评审软件开发计划的人员有：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、对软件开发计划进行管理和控制。

软件质量管理实践 - 软件项目策划

○ 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。软件开发计划包括：

- 1、软件项目的目的、范围、目标、和对象。
- 2、软件生存周期的选择。
- 3、精选的供开发和维护软件用的规程、方法和标准的确定。
软件标准和规程的例子有：软件开发策划，软件配置管理，软件质量保证，软件设计，问题跟踪和解决，和软件测量。
- 4、待开发软件工作产品的确定。
- 5、对软件工作产品的规模估计和对软件工作产品的更改。
- 6、对软件项目的工作量和成本的估计。
- 7、关键计算机资源的预计使用情况。
- 8、软件项目的进度，包括里程碑和评审的确定。
- 9、项目软件风险的识别和评估。
- 10、关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。

软件质量管理实践 - 软件项目策划

- 活动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。 该规程一般规定：
 - 1、对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：功能点（functionpoints），特征点（featurepoints），代码行，需求数，和页数。

要作规模估计的工作产品和活动的类型的例子有：运行软件和支持软件，可交付的和不可交付的工作产品，软件和非软件工作产品（例如文档），和开发、验证和确认工作产品的活动。
 - 2、将软件工作产品分解到能满足估计对象所需要的粒度。
 - 3、在可得到历史数据的地方使用历史数据。
 - 4、将规模估计的假定记入文档。
 - 5、对规模估计建立文档，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可以是：项目经理，项目软件经理，和其它软件经理。

软件质量管理实践 - 软件项目策划

- 活动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。
该规程一般规定：
 - 1、对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、在生产率数据（历史的和当前的）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（有：直接的劳务费，管理费，差旅费，和计算机使用成本）。
 - 3、对工作量、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、将估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。

软件质量管理实践 - 软件项目策划

- 活动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。

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、识别项目的关键计算机资源。

关键计算机资源包括：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、对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，运行处理的负载和通信量。

- 3、对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。

软件质量管理实践 - 软件项目策划

○ 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、软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，软件工作量和成本。
- 2、软件进度表基于过去的经验。
可能时，利用类似项目的经验。
- 3、软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、软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、将导出进度表时的假定记入文档。
- 6、对软件进度表建立文档、进行评审、并使之得到认可。

软件质量管理实践 - 软件项目策划

- 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、基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、鉴别风险的偶发事件。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、对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、就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、所有受到影响的组评审该计划。
- 活动15：记录软件策划数据。
 - 1、所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、对软件策划数据进行管理和控制。

软件质量管理实践 - 软件项目策划

n 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、评审技术、成本、人员配置和进度等性能。
- 2、分析在较低层次上未解决的矛盾和问题。
- 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和个人。

软件质量管理实践 - 软件项目策划

- 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、 受影响的组有代表出席。
 - 2、 对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、 分析组间的依赖关系。
 - 4、 分析较低层次上未解决的矛盾和问题。
 - 5、 评审软件项目风险。
 - 6、 分配和评审措施条款，并跟踪到结束。
 - 7、 准备每次会议的摘要报告，并将其散发给受影响的组和个人。

软件质量管理实践 - 软件项目策划

- 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。

至少，评审和审计要查证：

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

软件质量管理实践 - 软件项目策划

n 主要文件及内容要求

输入文件：组织方针，开发计划规程，工作陈述，软件开发计划模板；

输出文件：项目承诺、软件开发计划，评审意见，度量报告。

软件质量管理实践-软件项目跟踪与监督

- 概述
- ISO9000的要求
- CMM的要求
- 实践过程
 - n 过程目标
 - n 执行约定
 - n 执行能力
 - n 执行活动
 - n 验证实施

软件质量管理实践-软件项目跟踪与监督

○ 概述

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。

软件质量管理实践-软件项目跟踪与监督

○ ISO9000的要求

在ISO9001:2000中，相关章节有：8.2.3 过程的监视和测量；8.5.2 纠正措施。

○ CMM的要求

关键过程域SPTO（Software Project Tracking and Oversight），强调过程的跟踪与监控。

软件质量管理实践-软件项目跟踪与监督

○ 实践过程

n 过程目标

1. 根据软件计划跟踪实际的结果和性能
2. 当实际情况明显偏离软件计划时，采取纠正措施并对该措施加以管理直到结束
3. 软件约定的改变需要得到有关组织和个人同意

软件质量管理实践-软件项目跟踪与监督

n 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，即组织方针。该方针一般规定：
 - 1、采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、随时向项目经理报告软件项目的状态和问题。
 - 3、当软件计划未实现时，采取纠正措施，或者调整性能，或者调整计划。
 - 4、在受影响的组参与和认可的情况下对软件约定进行更改。
受影响的组有：软件工程组（包括所有小组，例如软件设计小组），软件估计组，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、高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。

软件质量管理实践-软件项目跟踪与监督

n 执行能力

- 对软件项目的软件开发计划已建立文档和批准。
- 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、待开发的软件工作产品或将提供的服务。
 - 2、这些软件活动的工作量和成本。
 - 3、这些软件活动的进度。
 - 4、这些软件活动的预算。
- 提供足够的用以跟踪软件项目的资源和投资。
 - 1、分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、使得支持软件跟踪的工具是可用的。

支持的工具包括：电子表格程序，项目策划 / 制定进度表程序。

软件质量管理实践-软件项目跟踪与监督

n 执行能力（续）

- 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，管理职员。
- 一线软件经理在软件项目的技术方面受到定向培训。定向培训包括：
项目的软件工程标准和规程，项目的应用领域。

软件质量管理实践-软件项目跟踪与监督

n 执行活动

- 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。这个软件开发计划：
 - 1、随着工作进展而更新，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。

软件质量管理实践-软件项目跟踪与监督

n 执行活动（续）

o 活动2：按照已文档化的规程修订项目的软件开发计划。

该规程一般规定：

- 1、当合适时，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、更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、在每次修订时，评审软件开发计划。
- 4、对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）。

软件质量管理实践-软件项目跟踪与监督

n 执行活动（续）

- 活动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。
- 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有关组的成员（软件质量保证组，软件配置管理组，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。
 - 1．跟踪所有主要软件工作产品的规模（或更改的规模）。
 - 2．将实际代码规模（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．将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．定期精炼、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．和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。

软件质量管理实践-软件项目跟踪与监督

n 执行活动（续）

- 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。
 - 1．将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．将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．和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：跟踪项目的关键计算机资源，必要时采取纠正措施。
 - 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。

软件质量管理实践-软件项目跟踪与监督

n 执行活动（续）

- 活动8：跟踪项目的软件进度，必要时采取纠正措施。
 - 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．跟踪问题报告直至结束。
- 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。
 - 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。

软件质量管理实践-软件项目跟踪与监督

n 执行活动（续）

- 活动11：记录软件项目的实际测量数据和重新策划的数据。
 - 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：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展、计划、性能和问题。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。

软件质量管理实践-软件项目跟踪与监督

n 执行活动（续）

- 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、当合适时，由雇客、最终用户和组织内部受影响的组参与进行。
 - 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。

软件质量管理实践-软件项目跟踪与监督

n 验证实施

- 实施1：高级管理者定期参与评审软件项目跟踪和监督活动。

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。

软件质量管理实践-软件项目跟踪与监督

n 验证实施（续）

- 实施2：项目经理既定期地又事件驱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。

软件质量管理实践-软件项目跟踪与监督

n 验证实施（续）

- 实施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。

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。

软件质量管理实践-软件项目跟踪与监督

n 过程实施框架的文件及内容要求

输入文件：组织方针，开发计划**更改**规程；

输出文件：实际测量数据和再计划数据。

软件质量管理实践-软件项目跟踪与监督

实际测量数据和再计划数据 包括软件计划数据、再计划数据、软件估计数据、相应的估计理由和实际的测量结果

- n 软件工作产品大小的估计变更数据
- n 项目工作量和花费的估计变更数据
- n 项目关键计算机资源的估计变更数据
- n 项目进度的估计变更数据
- n 软件花费、资源、进度和技术风险的记录
- n 项目经理对高风险区域的评审记录
- n 软件工作产品中出现的问题的记录
- n 软件工程组定期内部评审的记录
- n 正式评审的记录
- n 正式评审产生的重要问题的决定
- n 对于承诺的更改的评审记录

软件质量管理实践-软件子合同管理

- 概述
- 对应的ISO9000的要求
- 实践过程
 - n 过程目标
 - n 执行约定
 - n 执行能力
 - n 执行活动
 - n 验证实施

软件质量管理实践-软件子合同管理

○ 概述

软件子合同管理，即Software Subcontract Management。由于SW-CMM是美国国防部投资研究的项目，而美国军方有大量的子合同转包，因此子合同管理成为一个基本的关键过程域。

子合同管理的目的就是选择合格的软件承包商，并可进行有效的管理。

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。

软件质量管理实践-软件子合同管理

○ 对应的ISO9000的要求

在ISO9001:2000中，相关章节有：7.4 采购

它实际上既包括一般采购产品的管理，也包括了子合同的管理。主要要求和内容包括：

- n 建立采购信息：包括采购要求和人员资格、质量管理体系等要求
- n 采购产品的验证：组织应建立并实施检验或其他必要的活动，以确保采购的产品满足规定的采购要求。

软件质量管理实践-软件子合同管理

○ 实践过程

n 过程目标

- 主承包商选择合格的软件予承包商。
- 主承包商和软件子承包商认同他们相互的约定。
- 主承包商和软件子承包商保持不断的通信。
- 主承包商对照约定跟踪软件予承包商的实际结果和性能。

软件质量管理实践-软件子合同管理

○ 实践过程（续）

n 执行约定

○ 组织方针。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、将合同协议形成管理子合同的基础。
- 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。

○ 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，如采购、财务和法律组，建立和监控子合同的条款和条件。

- 3、子合同经理负责：选择软件子承包商，管理软件子合同，安排子合同产品的后续合同支持。

软件质量管理实践-软件子合同管理

○ 实践过程（续）

n 执行能力

- 为选择软件子承包商和管理子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。

支持工具包括：估计模型，电子表格程序，和项目管理和编制进度程序。

- 参与建立及管理软件子合同的软件经理和其它个人受到完成这些活动的培训。

培训有：准备和策划软件子合同的签订，评价子合同投标者的软件过程能力，评价子合同投标者的软件估计和计划，选择子承包商，和管理子合同。

- 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。

定向培训的例子有：应用领域，正运用的软件技术，正使用的软件工具，正使用的方法论，正使用的标准，和正使用的规程。

软件质量管理实践-软件子合同管理

○ 实践过程（续）

n 执行活动

○ 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，使其与潜在子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，和软件标准和现程。

软件质量管理实践-软件子合同管理

- 3、对子合同的工作陈述进行：准备，评审，认同，当需要时修订，和管理和控制。

评审和认同子合同工作陈述的个人包括：项目经理，项目软件经理，应负责的软件经理，软件配置管理经理，软件质量保证经理，和子合同经理。

“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）。

- 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。

软件质量管理实践-软件子合同管理

○ 实践过程（续）

n 执行活动（续）

- 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（如有）。
- 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。
- 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。资源包括：设施，硬件，软件，和培训。

软件质量管理实践-软件子合同管理

○ 实践过程（续）

n 执行活动（续）

- 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。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（源代码，软件开发计划，仿真环境，设计文档，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。

软件质量管理实践-软件子合同管理

○ 实践过程（续）

n 执行活动（续）

- 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。
- 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。
 - 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。

软件质量管理实践-软件子合同管理

实践过程（续）

n 执行活动（续）

- 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。
 - 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。

软件质量管理实践-软件子合同管理

○ 实践过程（续）

n 执行活动（续）

○ 活动8：软件子承包商参与定期技术评审和交流。

这些评审包括：

- 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。

软件质量管理实践-软件子合同管理

○ 实践过程（续）

n 执行活动（续）

- 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。

软件质量管理实践-软件子合同管理

○ 实践过程（续）

n 执行活动（续）

- 活动10：主承包商的软件质量保证组按照已文档化的规程监控子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，以保证它们适用监控子承包商的性能。
- 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、定期审计子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。

软件质量管理实践-软件子合同管理

○ 实践过程（续）

n 执行活动（续）

- 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子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、定期地审计子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。

软件质量管理实践-软件子合同管理

实践过程（续）

n 执行活动（续）

- 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。

软件质量管理实践-软件子合同管理

○ 实践过程（续）

n 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。

- n 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。

软件质量管理实践-软件子合同管理

○ 实践过程（续）

n 验证实施（续）

- 验证3：软件质量保证组评审（或）审计管理软件子合同的活动和工作产品，并报告其结果。

至少，该评审和（或）审计要查证：

- 1、选择子承包商的活动。
- 2、管理软件子合同的活动。
- 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。

软件质量管理实践-软件质量保证

- 概述
- 对应的ISO9000的要求
- CMM的要求
- 实践过程
 - n 过程目标
 - n 执行约定
 - n 执行能力
 - n 执行活动
 - n 验证实施

软件质量管理实践-软件质量保证

○ 概述

- n 如何保证软件产品的高质量是软件生产的目标。软件质量保证过程作为一种第三方的、独立的审查活动贯穿于整个软件生产过程。
- n 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。
- n 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。

软件质量管理实践-软件质量保证

○ 对应的ISO9000的要求

相关章节有：ISO9001:2000中，5.4.2 质量管理体系策划，5.6 管理评审，7.3.4 设计和开发评审，7.5.2 生产和服务提供过程的确认，8.2.2 内部审核，8.5.2 纠正措施，8.5.3 预防措施

以上章节，实际上包括了对质量保证的整体性策划(5.4.2)，过程的评测(7.5.2)和结果产品的审计(7.3.4)，以及发现问题(8.2.2)和发现问题后的处理(8.5.2)

软件质量管理实践-软件质量保证

○ CMM的要求

软件质量保证过程主要内容可概括为三方面：

- n 通过监控软件的开发过程来保证产品的质量；
- n 保证生产出的软件和软件开发过程符合相应的标准与规程；
- n 保证软件产品、软件过程中存在的不符合问题得到处理，必要时将问题反映给高级管理者。

结合这三项内容，软件质量保证过程主要有审计、评审和处理不符合问题等三项主要活动。

- n 审计包括对软件工作产品、软件工具和设备的审计。审计是为了评估软件工作产品及工具设备是否符合组织和项目的标准，鉴别偏差及疏漏以便跟踪评价。
- n 评审是指对软件过程的评审，其主要任务是保证组织定义的软件过程在项目中得到了遵循。审计和评审的结果记录在相应的报告中。
- n 对于审计和评审过程中发现的不符合问题，软件质量保证负责人要进行跟踪和处理。一般处理问题的原则是发现问题首先进行项目内部处理，内部不能解决的，依据管理层次层层提升，直至问题得到解决。

软件质量保证过程的核心在于验证产品和活动的符合性，而软件质量保证过程并不对软件产品的质量负责。

软件质量管理实践-软件质量保证

○ 实践过程

n 过程目标

- 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。

软件质量管理实践-软件质量保证

○ 实践过程（续）

n 执行约定

- 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，和其它的软件有关组（软件配置管理组，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：

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”。

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。

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。

软件质量管理实践-软件质量保证

○ 实践过程（续）

n 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。

软件质量管理实践-软件质量保证

○ 实践过程（续）

n 执行能力（续）

○ 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。

工具有：工作站，数据库程序，电子表格程序，和审计工具。

软件质量管理实践-软件质量保证

○ 实践过程（续）

n 执行能力（续）

- **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：软件工程技巧和实践；软件工程组和其它软件一有关组的岗位任务及职责；用于软件项目的标准、规程和方法；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。

软件质量管理实践-软件质量保证

○ 实践过程（续）

n 执行活动

○ 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；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）。

软件质量管理实践-软件质量保证

○ 实践过程（续）

n 执行活动（续）

○ 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，可交付的和不可交付的产品，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），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。

软件质量管理实践-软件质量保证

○ 实践过程（续）

n 执行活动（续）

○ 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。

1、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，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。

软件质量管理实践-软件质量保证

○ 实践过程（续）

n 执行活动（续）

○ 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。

软件质量管理实践-软件质量保证

○ 实践过程（续）

n 执行活动（续）

○ 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。

软件质量管理实践-软件质量保证

○ 实践过程（续）

n 执行活动（续）

- **活动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。

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。

软件质量管理实践-软件质量保证

- 实践过程（续）

- n 执行活动（续）

- **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。

软件质量管理实践-软件质量保证

○ 实践过程（续）

n 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。

○ 验证2：项目经理既定期地也事件驱动地参与评审SQA 活动。

○ 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。

软件质量管理实践-软件配置管理

- 概述
- 对应的ISO9000的要求
- CMM的要求
- 实践过程
 - n 过程目标
 - n 执行约定
 - n 执行能力
 - n 执行活动
 - n 验证实施

软件质量管理实践-软件配置管理

○ 概述

- n 软件配置管理 (Software Configuration Management, SCM) 是一组管理整个软件生成期中变更的活动。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。
- n 配置管理活动主要包括标识软件工作产品、进行配置控制、记录配置状态和配置审计。
 - 配置标识是配置管理的基础。
 - 配置控制是配置管理的核心工作。配置控制主要包括存取控制、版本控制、变更控制和产品发布4个方面
 - 配置状态的记录用来协调对软件产品的控制。
 - 配置审计用来验证软件基准库中软件产品的一致性和完整性。

软件质量管理实践-软件配置管理

○ ISO9000的要求

相关章节有：ISO9001:2000中，7.5.3 标识和可追溯性，内容如下：

适当时，组织应在产品实现的全过程中使用适宜的方法识别产品。

组织应针对监视和测量要求识别产品的状态。

在有可追溯性要求的场合，组织应控制并记录产品的唯一性标识。

软件质量管理实践-软件配置管理

○ CMM的要求

- n 项目基线项（含项目描述、项目的产品和过程的规格说明）的选择；
- n 系统地使用已确定的更改控制过程来控制基线项的更改；
- n 在任何时候都能明显地标识出一个系统的技术状态（配置、软件和文件）或任何一个受控的中间产品或支持产品。

软件质量管理实践-软件配置管理

○ 实践过程

n 过程目标

- 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。

软件质量管理实践-软件配置管理

实践过程（续）

n 执行约定

- 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。

软件质量管理实践-软件配置管理

○ 实践过程（续）

n 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。

软件质量管理实践-软件配置管理

○ 实践过程（续）

n 执行能力（续）

- **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。

软件质量管理实践-软件配置管理

- 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。

软件质量管理实践-软件配置管理

○ 实践过程（续）

n 执行能力（续）

○ 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，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；SCM工具。

软件质量管理实践-软件配置管理

○ 实践过程（续）

n 执行能力（续）

- 能力5：软件工程组和其它软件有关组的成员受到培训以便完成其SCM 活动。

其它软件有关组有：软件质量保证组，和文档支持组。

培训的例子有：1、在软件工程组和其它软件有关组的内部进行SCM 活动要遵循的标准、规程和方法，2、SCM 组的角色、职责和权力。

软件质量管理实践-软件配置管理

○ 实践过程（续）

n 执行活动

- **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）。

- **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有关组进行的SCM 活动。

软件质量管理实践-软件配置管理

○ 实践过程（续）

n 执行活动（续）

○ 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：1）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），2）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。

软件质量管理实践-软件配置管理

○ 实践过程（续）

n 执行活动（续）

○ 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：过程一有关文档（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说的所有者）。

软件质量管理实践-软件配置管理

○ 实践过程（续）

n 执行活动（续）

- **活动5**：按照已文档化的规程，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。

软件质量管理实践-软件配置管理

○ 实践过程（续）

n 执行活动（续）

- **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）。

软件质量管理实践-软件配置管理

○ 实践过程（续）

n 执行活动（续）

- **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。

软件质量管理实践-软件配置管理

○ 实践过程（续）

n 执行活动（续）

○ 活动10：活动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。

软件质量管理实践-软件配置管理

○ 实践过程（续）

n 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。

○ 验证2：项目经理既定期地也事件驱动地参加评审SCM活动。

○ 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。

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。



建立软件测试管理体系

- 软件测试是一个必不可少的活动，是对软件需求分析、设计规约和编码的最终复审；是软件质量保证的关键步骤。
- 软件测试是根据软件开发各阶段的规约和软件的内部结构，精心设计一批测试用例（包括输入数据及其预期的输出结果），并利用这些测试用例去运行程序，以发现软件中不符合质量特性要求（即缺陷或错误）的过程。
- 建立软件测试管理体系的主要目的是确保软件测试在软件质量保证中发挥应有的关键作用

建立软件测试管理体系

一般应用过程方法和系统方法来建立软件测试管理体系，在设定的条件限制下，尽可能发现和排除软件缺陷。测试系统主要由下面6个相互关联、相互作用的过程组成：

n 测试规划

确定各测试阶段的目标和策略。这个过程将输出测试计划，明确要完成的测试活动，评估完成活动所需要的时间和资源，设计测试组织和岗位职权，进行活动安排和资源分配，安排跟踪和控制测试过程的活动。

测试规划与软件开发活动同步进行。在需求分析阶段，要完成验收测试计划，并与需求规格说明一起提交评审。类似地，在概要设计阶段，要完成和评审系统测试计划；在详细设计阶段，要完成和评审集成测试计划；在编码实现阶段，要完成和评审单元测试计划。对于测试计划的修订部分，需要进行重新评审。

建立软件测试管理体系

n 测试设计

根据测试计划设计测试方案。测试设计过程输出的是各测试阶段使用的测试用例。测试设计也与软件开发活动同步进行，其结果可以作为各阶段测试计划的附件提交评审。测试设计的另一项内容是回归测试设计，即确定回归测试的用例集。对于测试用例的修订部分，也要求进行重新评审。

n 测试实施

使用测试用例运行程序，将获得的运行结果与预期结果进行比较和分析，记录、跟踪和管理软件缺陷，最终得到测试报告。

n 配置管理

测试配置管理是软件配置管理的子集，作用于测试的各个阶段。其管理对象包括测试计划、测试方案（用例）、测试版本、测试工具及环境、测试结果等。

建立软件测试管理体系

n 资源管理

包括对人力资源和工作场所，以及相关设施和技术支持的管理。如果建立了测试实验室，还存在其他的管理问题。

n 测试管理

采用适宜的方法对上述过程及结果进行监视，并在适用时进行测量，以保证上述过程的有效性。如果没有实现预定的结果，则应进行适当的调整或纠正。

此外，测试系统与软件修改过程是相互关联、相互作用。测试系统的输出（软件缺陷报告）是软件修改的输入。反过来，软件修改的输出（新的测试版本）又成为测试系统的输入。

建立软件测试管理体系

- 根据上述6个过程，可以确定建立软件测试管理体系的6个步骤：
 - n 识别软件测试所需的过程及其应用，即测试规划、测试设计、测试实施、配置管理、资源管理和测试管理；
 - n 确定这些过程的顺序和相互作用，前一过程的输出是后一过程的输入。其中，配置管理和资源管理是这些过程的支持性过程，测试管理则对其他测试过程进行监视、测试和管理；
 - n 确定这些过程所需的准则和方法，一般应制订这些过程形成文件的程序，以及监视、测量和控制的准则和方法；
 - n 确保可以获得必要的资源和信息，以支持这些过程的运行和对它们的监测；
 - n 监视、测量和分析这些过程；
 - n 实施必要的改进措施。