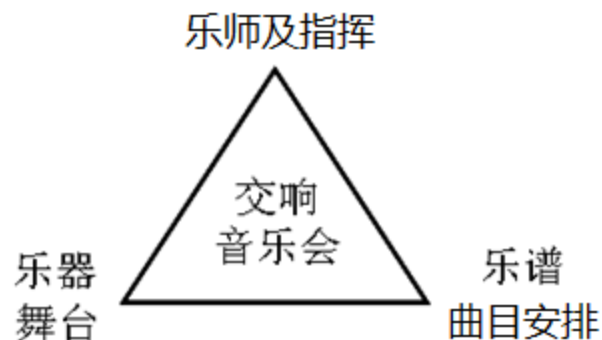
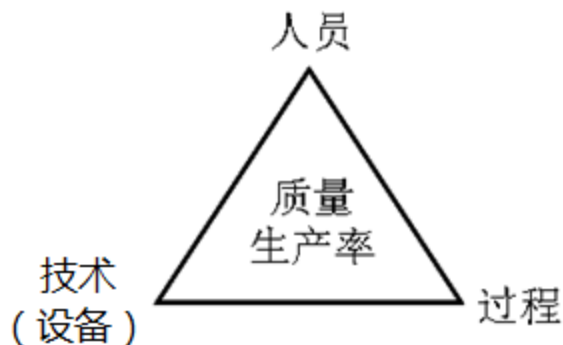


第12章 软件过程与软件过程改进

- 软件过程概述
- 软件生存期过程国际标准
- 软件过程成熟度
- 软件能力成熟度模型
- 软件过程改进

第12章 软件过程与软件过程改进

- 尽管软件工程项目的具体情况各有不同，但影响软件产品质量和软件项目生产率的主要因素主要有3个：人员、技术和过程。
- 如果把音乐会演出视为软件项目，则乐队和指挥就相当于参与项目的人员；舞台和乐器相当于支持开发的环境条件、技术和工具；而乐谱相当于软件项目的过程。



12.1 软件过程概述

- **过程概念**

- **做任何事，完成任何工作都需要有步骤、有顺序地进行。**
- **这些步骤和顺序都是由事务和工作的规律决定的，不可任意打乱。**
- **以传统的制造业为例，从原材料开始直至得到用户所需要的产品，经历了完整的生产过程。经过进一步分析，生产过程由许多生产子过程组成。**

12.1 软件过程概述

1. 直接子过程（或称基本过程）

- (1) 市场调查：了解用户和市场的需求，对需求信息进行分析，以确定生产产品的型号、功能和性能；**
- (2) 产品设计：把需求变成可实现的方案；**
- (3) 生产制作：获得产品；**
- (4) 检验包装：准备出厂；**
- (5) 仓储运输：送到用户手中。**

12.1 软件过程概述

2. 间接子过程（或称支持过程）

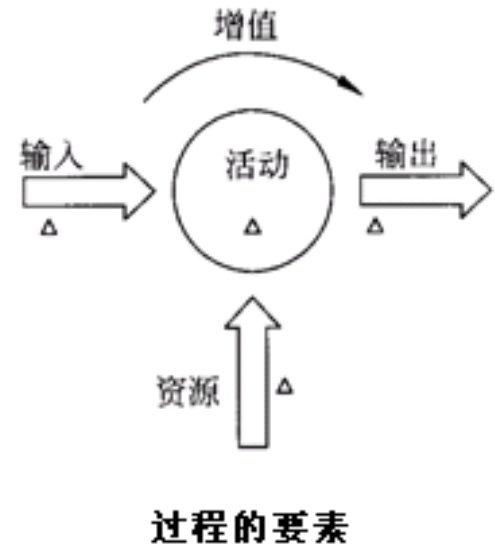
- (1) 检测手段的控制：如量测仪表的校准和调整；**
- (2) 不合格品的控制：如不合品的标识、隔离和处置；**
- (3) 人员培训：提高人员技能；**
- (4) 质量体系的建立与审核：实施质量管理规范化。**

12.1 软件过程概述

• 过程要素

过程应具有以下几个构成元素，如图所示：

- (1) 输入；
- (2) 输出；
- (3) 活动及进一步分解的任务（或作业）；
- (4) 资源是支持活动执行所必须的，包括人员、设备及相关的耗费；
- (5) 测量与验证能保证过程中的相关元素是合格的（在图中这些过程元素标有△记号）；
- (6) 过程目标，如活动取得的效果是增值。



12.1 软件过程概述

- **过程思维**

- **过程思维是近年来人们总结出的思维模式，它和已有几百年历史的任务思维有着本质的区别。**
- **面向过程的思维注重的是总体目标、各部分工作的协调性和一致性，从而消除了各部分工作之间的冲突，提高了总体的工作效率，从而有效地达到工作的总体目标。**
- **而面向任务的思维作为一种传统的思维模式，通常注重于任务、作业、人员和组织结构。它首先将任务分解，然后指派人员去完成。这样做必然会忽略了目标和整体。当各个局部工作之间出现矛盾和冲突时，再着手解决，这样做无法避免低效率运行现象的出现。**

12.1 软件过程概述

- **软件过程**

- **尽管软件危机最早于上个世纪六十年代已经提出，业内人士确已投入了大量人力和物力，试图摆脱软件业长期的发展之痛。然而，事实表明，问题至今还未得到很好地解决。**
- **在近十几年的大量研究和实践之后，许多人逐渐认识到，从软件过程的改进来解决可能是有效的方法之一。**

12.1 软件过程概述

- **软件过程**

- **著名的软件工程专家，也是CMM的主要创始人Watts Humphrey提出了以下几个重要论点：**

- (1) 软件系统的质量取决于用以开发和改进它的过程质量。**
- (2) 解决软件问题的重要一步是把整个软件工作当作一个过程来对待，使其能够控制、度量和改进。**
- (3) 软件过程是我们用以开发软件产品的一套工具、方法和实践。**
- (4) 软件过程管理的目标是按计划生产产品，同时提高软件组织的能力，以利于生产出好的产品。**

12.1 软件过程概述

- **软件过程**

- (5) **成本估算和开发期安排的承诺应该都是比较合理的，开发出的产品应该在功能和质量方面都能满足用户的期望。**
- (6) **有效的软件管理必须考虑所要完成的任务，所采用的方法和工具，以及参与工作人员的技能、培训和积极性。**
- (7) **有效的软件过程必须是可预测的。**

12.2 软件生存期过程国际标准

- 1995年8月，国际标准化组织（ISO）和国际电工委员会（IEC）联合发布了国际标准ISO / IEC 12207 Information Technology Software Life Cycle Processes。
- 2008年，这两个国际标准化机构又联合了美国电气与电子工程师学会（IEEE），共同发布了该标准的第二版：
ISO / IEC 12207 :2008 (IEEE Std. 12207-2008)
Systems and software engineering—Software life cycle processes。

12.2 软件生存期过程国际标准

- 结构

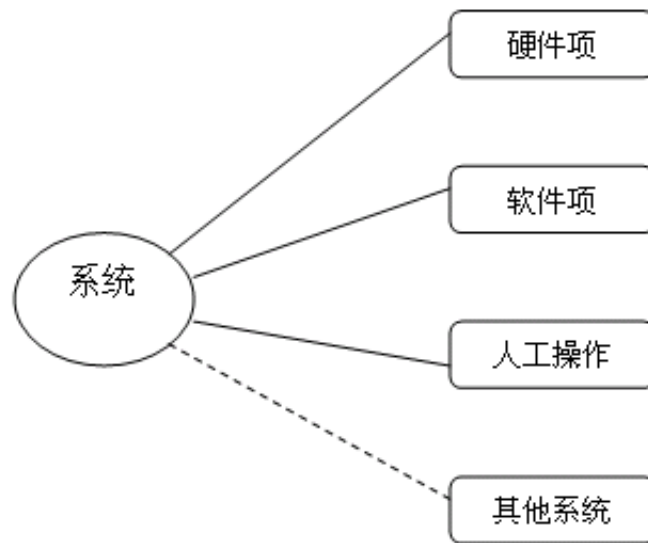
- 由于越来越多的软件作为系统的一部分开发和运行，因此我们在研究和处理软件问题时，建立系统的观念是十分必要的。软件项在系统中的位置如下图所示。

- 基于这一观念，该标准分为两部分：

- 1) 与系统相关的过程；

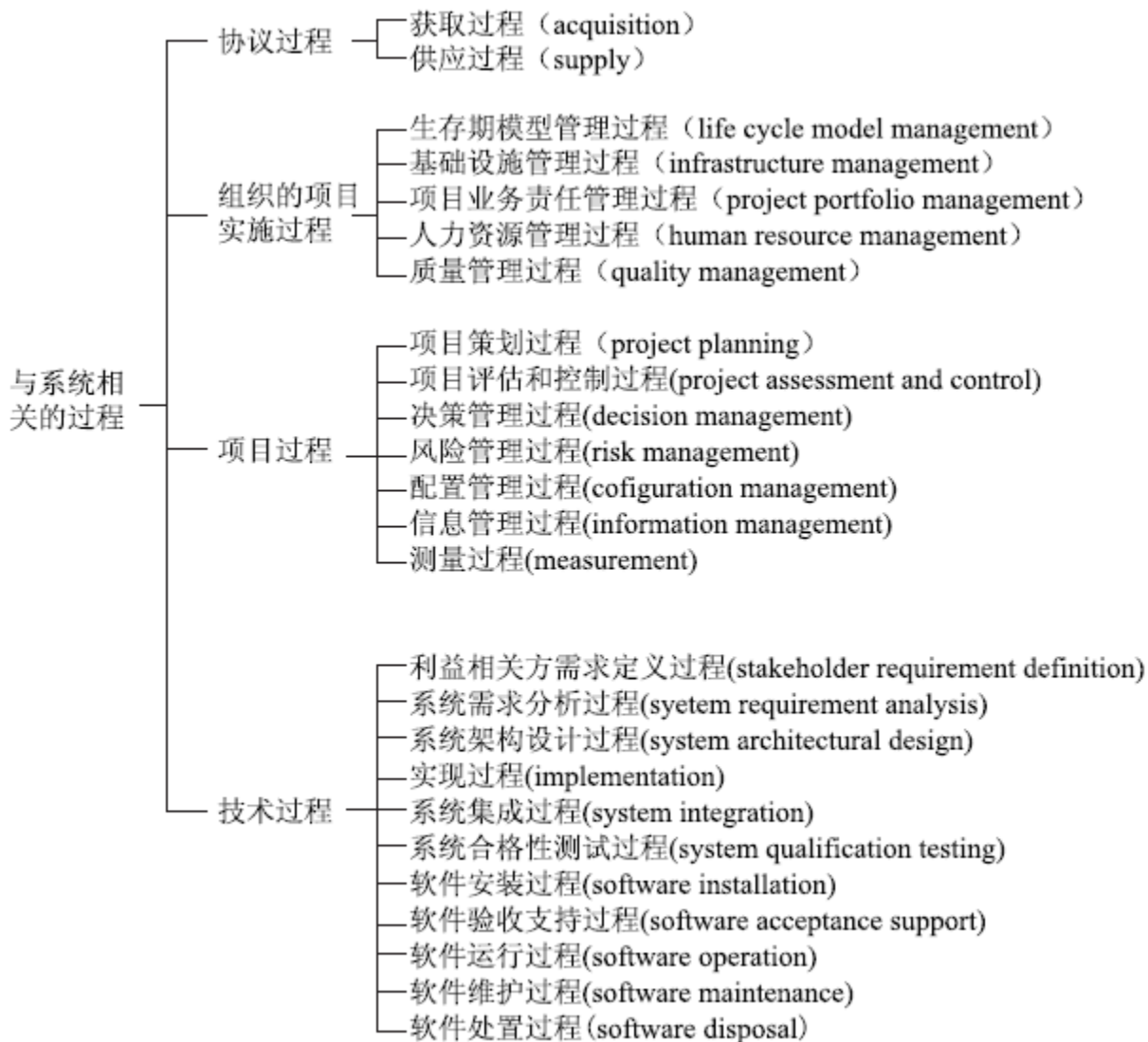
- 2) 软件特有过程。

系统元素



12.2 软件生存期过程国际标准

- 与系统相关的过程



12.2 软件生存期过程国际标准

- 软件特有过程



12.3 软件过程成熟度

- 什么是软件过程成熟度
- 过程制度化

12.3.1 什么是软件过程成熟度

- **软件过程成熟度的概念**
- **任何一个软件组织，在完成自身的开发、维护等工作中，都有自己的软件过程。这种过程有可能是初级的、低效的，也可能是高效的，在其成熟性方面存在差异，这当然是相互比较而言的。**
- **软件过程成熟度（Software Process Maturity）是软件过程改进的一个重要概念，它是指：一个特定软件过程得到清晰的定义、管理、测量、控制的有效程度。成熟度意味着能力的增长具有潜力，并表示组织软件过程是珍贵的，他在组织内所有项目中的应用是一致的。**

12.3.1 什么是软件过程成熟度

- 不成熟过程与成熟过程的对比

1) 角色与职责

不成熟过程	成熟过程
<ul style="list-style-type: none">• 没有明确规定角色和职责• 每个人做自己认为要做的事• 常会发生重叠和不清楚的所属关系和责任	<ul style="list-style-type: none">• 明确规定角色和职责• 相互关系无重叠• 有明确的目标和测量方法• 能够体现持续改进过程的机制

12.3.1 什么是软件过程成熟度

2) 处理变更的方式

不成熟过程	成熟过程
<ul style="list-style-type: none">●每个人都按自己的想法做事	<ul style="list-style-type: none">●遵循一个规划好的文件化过程●可分享取得的经验

12.3.1 什么是软件过程成熟度

3) 对发生问题的反应

不成熟过程	成熟过程
<ul style="list-style-type: none">●无秩序的混乱现象随处可见●“救火”方式解决出现问题的情况经常发生●每个人都想当英雄	<ul style="list-style-type: none">●根据已有的知识和专业规则对发生的问题进行分析和处理

12.3.1 什么是软件过程成熟度

4) 可信性

不成熟过程	成熟过程
<ul style="list-style-type: none">●有时延迟交付产品或超出预算●如有估算也不可靠	<ul style="list-style-type: none">●估算准确●项目得到有效的控制和管理●目标一般能够达到

12.3.1 什么是软件过程成熟度

5) 对工作人员的奖励

不成熟过程	成熟过程
<ul style="list-style-type: none">●奖励的对象是“救火”队员●“如果你第一次就把事情做好了，那是你的本分，没有人理睬，但你若先把事情搞乱，然后再去解决，你就成了英雄。” <p>_Deming,1986</p>	<ul style="list-style-type: none">●奖励那些生产高质量产品的团队●奖励那些防火者而不是救火者

12.3.1 什么是软件过程成熟度

6) 预见性

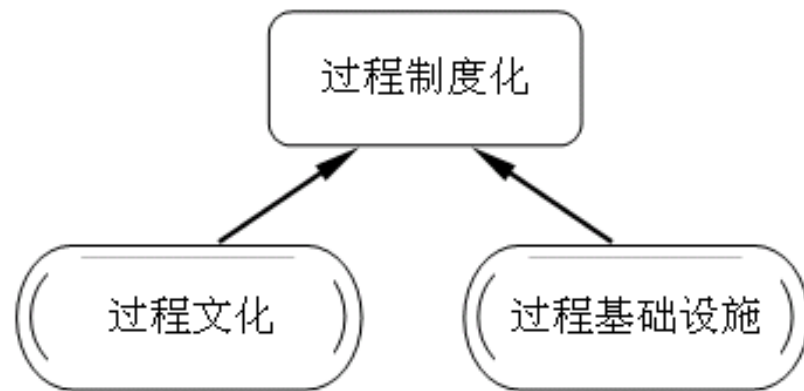
不成熟过程	成熟过程
<ul style="list-style-type: none">●质量不可把握，它依赖于个人●进度和预算不能根据以往的经验确定	<ul style="list-style-type: none">●项目的进度和产品的质量均可预见●进度和预算可根据以往项目的经验确定，并且是符合实际的

12.3.2 过程制度化

- **过程认同与过程制度化**
 - **软件开发过程决定了在接受软件工程项目后工作人员的行动方式和反应方式。为了实现某个既定的目标，人们的行为、活动和任务都要遵循为达到此目标所经历的过程。**
 - **规范化的过程体现了一系列有序的和协调一致的行为模式，无论这个过程是由一个人来完成或是由一个团队人员共完成都是如此。**

12.3.2 过程制度化

- 过程认同与过程制度化
 - 当一个规范化过程已经渗入组织的日常生活之中，过程的要求已经变成全体员工的自觉行动，得到大家的认同和坚持遵循时，过程便成为制度化的（ process institutionalization ）。
 - 做到这一点并不容易，要靠过程文化和过程基础设施的支持，如图所示。



过程制度化的所需的支持

12.3.2 过程制度化

- **过程文化**

- **过程文化是指人们的习惯和行为受到过程思维和过程管理原则的影响。人们对于规范化过程是完全认同的，就是说，人们的活动自觉地按过程要求去作。**
- **过程文化应具有自己的特性。**

12.3.2 过程制度化

• 过程文化

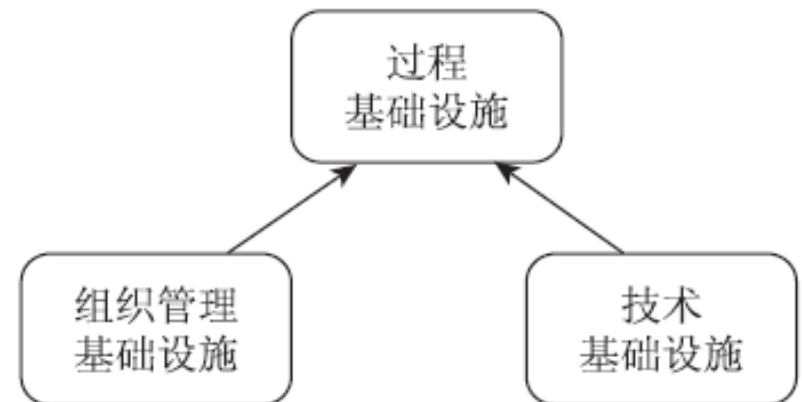
过程文化的特征

特征	表现
可见性	组织中对每个人来说过程定义和过程职责都是清晰的
规范性	遵循过程是常规的要求，过程之外的行为是个别的例外
制度化	遵循过程已列入组织方针和规程，并得到管理者的支持
管理者承诺	过程得到最高管理者以及其他管理者和员工的支持
推行	过程的推行是坚定的，也是有成效的
所有者	过程为组织基础设施所拥有，并得到维护、持续改进和支持
反馈	组织中每个人都可对过程的有效性提出反馈意见，并且已经建立了适当的反馈机制
绩效评估	对员工和团队工作的评价与过程效果紧密联系，即要看过程目标达到的情况
培训	对全体员工的过程培训是强制性的，对新员工的过程入门培训也是强制性的
改进	全体相关人员要自始至终参与过程改进的策划和实施

12.3.2 过程制度化

- **过程基础设施**

- **对于软件过程来说，基础设施指的是支持软件过程的基础框架和结构基础。它不仅包括组织和管理的岗位和职责，而且包括支持定义过程、开展过程活动、获取和分析过程有关绩效反馈以及不断进行过程改进活动所必须的技术工具和平台。**
- **事实上，过程基础设施包含了组织和管理基础设施、技术基础设施两个方面，如图所示。**



12.3.2 过程制度化

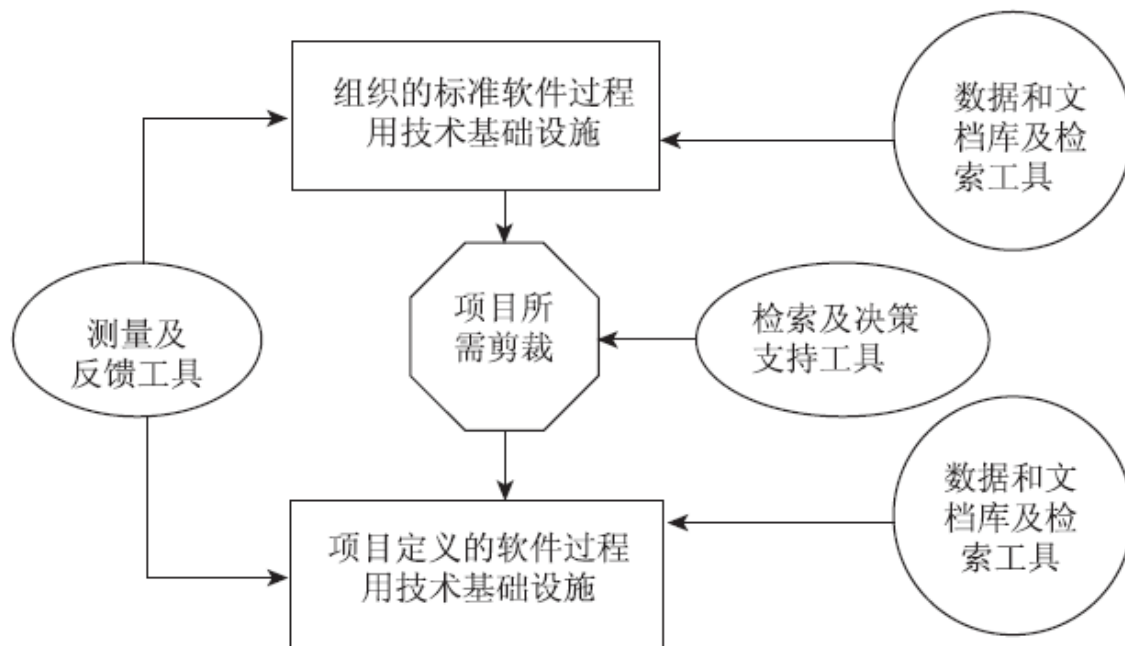
- 组织管理基础设施

- 组织和管理基础设施包括建立、监控和推进过程活动的岗位及其职责。支持过程的岗位和职责又有面向全局的和面向局部之分。
- 支持全局工作的功能组通常是在公司一级上工作的，如软件工程过程组（SPEG）。
- 支持局部工作的功能组可能是在项目级上工作，也可能是在某个特定的关键过程域上工作。
- 在这些功能组工作的人员有些是全职（或称全时）的，如SEPG组；也有的功能组人员是兼职的，如软件过程改进组（PIT Process Improvement Team）。

12.3.2 过程制度化

- 技术基础设施

- 软件过程技术基础设施是支持SEPG和PIT的技术平台、计算机设施和工具。
- 对于软件过程环境来说，基础设施的过程支持部件包括支持与过程有关活动的工具。下图表示软件过程技术基础设施的结构。



12.4 软件能力成熟度模型

- **CMM与SEI**
- **CMM的演化**
- **CMM族和CMMI**
- **CMM1.3简介**
- **CMMI评估**

12.4.1 CMM与SEI

- **什么是CMM**
- **CMM是能力成熟度模型Capability Maturity Model三个英文单词第一个字母组成的。事实上，该模型最早提出时，它指的是软件过程能力成熟度模型。**
- **该模型按软件过程的不同成熟度划分了5个等级，1级被认为成熟度最低，5级则成熟度最高。**

12.4.1 CMM与SEI

- **SEI的软件过程研究**
- **SEI是软件工程研究所Software Engineering Institute的英文字头缩写。该研究所隶属于美国宾州的卡内基-梅隆大学于1984年成立。**
- **SEI的任务是在软件工程领域中努力提高依赖软件的系统质量，促进软件开发和维护的工程化管理，为军方服务。**
- **SEI同时开始研究协助软件组织改进软件过程的途径，以期解决软件业面临的各种问题，特别是：**
 - **软件开发和维护的成本不断提高**
 - **软件产品的质量不能令人满意**
 - **软件项目经常不能按时完成，延误交付**

12.4.1 CMM与SEI

- **SEI的软件过程研究**
 - **CMM项目的主要负责人是Watts Humphrey , Mark Paulk等。Humphrey曾在IBM工作了27年，曾领导了操作系统开发等项目，具有丰富的软件管理经验。**
 - **软件项目的实践使他深刻地理解到计划和管理的重要意义，开始领导采用软件过程思想的开发项目。**
 - **1986年他将自己的研究成果带到SEI，主持了CMM的研究。**

12.4.1 CMM与SEI

- **SEI**的软件过程研究
 - **SEI**开展的有关过程的课题包括以下一些领域：**CMM**、基于**CMM**的过程评估、软件过程定义、人员软件过程（**PSP**）、团队软件过程（**TSP**）和软件工程测量与分析。

12.4.2 CMM的演化

- 1986年11月应美国政府要求，在IBM有关软件过程研究成果的基础上，项目开始启动。任务是开发一种模型，用其促进软件承包商提高产品质量。
- 1987年6月项目组提出了初始模型框架，9月给出了包含101个问题的初步成熟度提问单，用以评价软件承包商的风险。
- 1991年SEI推出CMM 1.0版，这是在上述软件过程成熟度框架和初始成熟度提问单经过4年应用的基础上提出的。
- 于1993年公布了CMM 1.1版。
- 1997年发布了CMM 2.0版。尽管这个版本曾有草稿A、B和C，但始终没有推广，在开展了CMMI的工作时CMM 2.0的工作就停止了。

12.4.3 CMM族和CMMI

- **基于CMM的模型**
- 自从CMM面世以来在各国软件界中产生了巨大影响，它在解决软件过程存在问题方面的成功使得相关领域也纷纷采纳和仿效它的模式，于是出现了多种基于CMM的模型，构成了一个CMM族。

CMM 的相关模型

名称	含义	内容	主题
P-CMM	People	人员	人力资源管理
SE-CMM	Systems Engineering	系统工程	系统工程管理
SA-CMM	Software Acquisition	软件获取	需求方获取软件
IPD-CMM	Integrated Systems Product Development	集成系统产品开发	集成系统开发管理
SSE-CMM	Systems Security Engineering	系统安全工程	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	团队软件过程	团队工作协调
PSP	Personal Software Process	个体软件过程	软件工程师工作

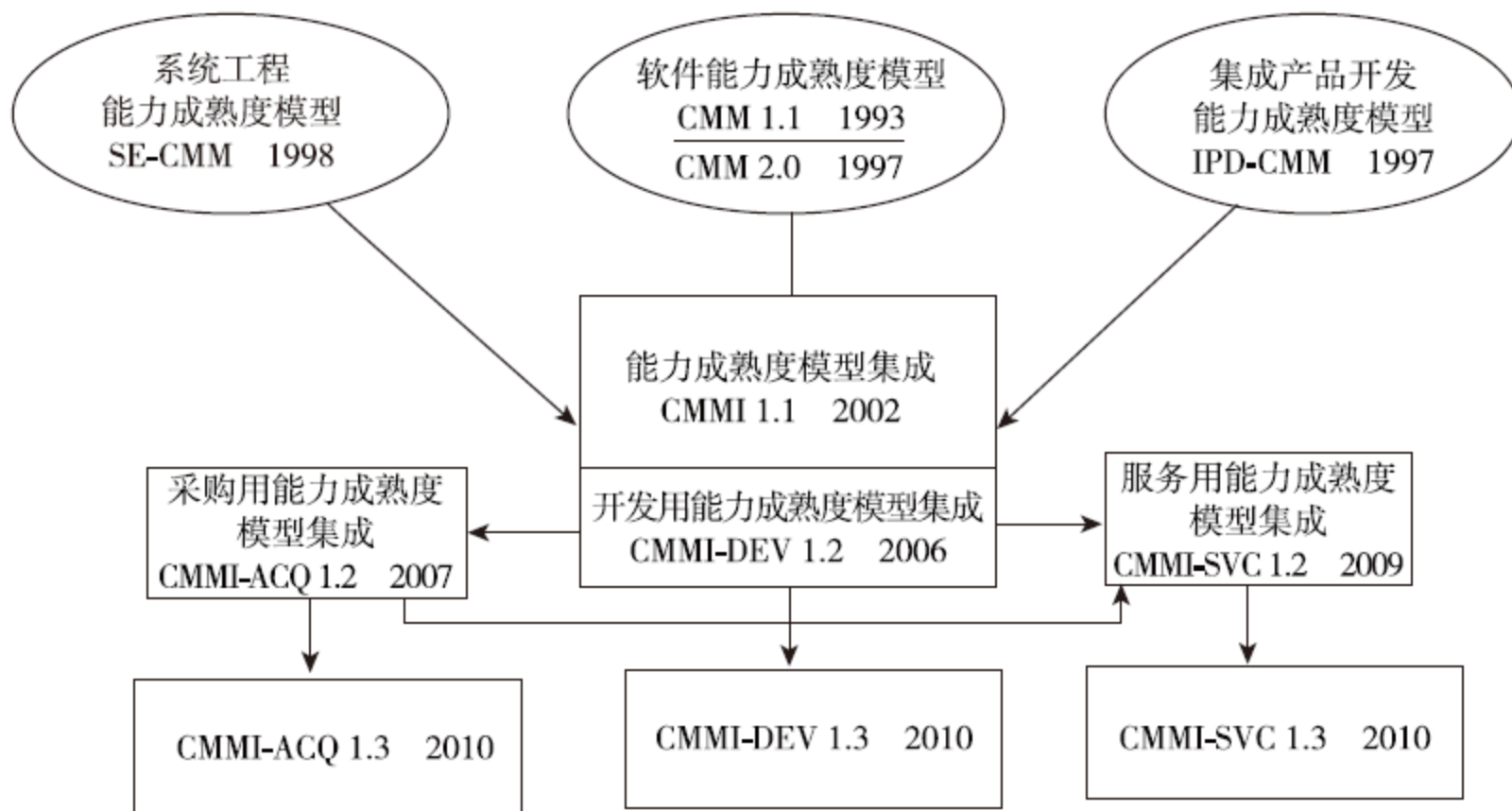
12.4.3 CMM族和CMMI

- **CMMI**

- 1997年SEI开始研究CMMI，其任务是将已有的CMM模型结合成一个模型，新的模型（CMMI）应该兼顾已采用CMM的组织，使其不致受到大的影响，又能便于它的新用户使用；同时还应该与国际标准ISO/IEC 15504相兼容。
- CMMI的1.1版于2002年发布，这一版本正是我国近年来实施的版本，它在我国软件业具有一定影响。
- 2006年发布了CMMI1.2。
- 2006年发布了CMMI1.3：包括CMMI-ACQ（用于采购）、CMMI-DEV（用于开发）、CMMI-SVC（用于服务）。

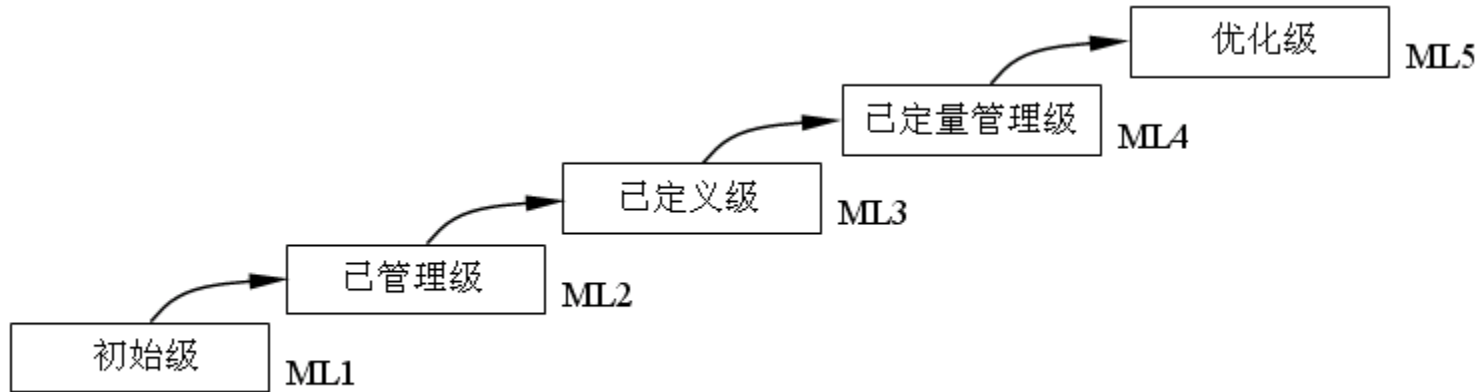
12.4.3 CMM族和CMMI

- CMMI来源和发展

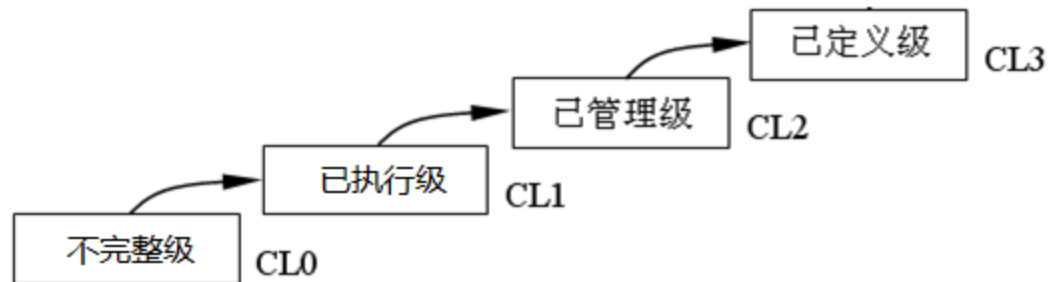


12.4.4 CMM1.3简介

- CMMI模型的两种表示



分级表示的组织成熟度等级



连续表示的过程域能力等级

12.4.4 CMM1.3简介

- 两种表示的对比

能力等级和成熟度等级的对比

等级	连续式表示 (能力等级)	分级式表示 (成熟度等级)
0级	不完整级 CL0	—
1级	已执行级 CL1	初始级 ML1
2级	已管理级 CL2	已管理级 ML2
3级	已定义级 CL3	已定义级 ML3
4级	—	已量化管理级 ML4
5级	—	持续优化级 ML5

12.4.4 CMM1.3简介

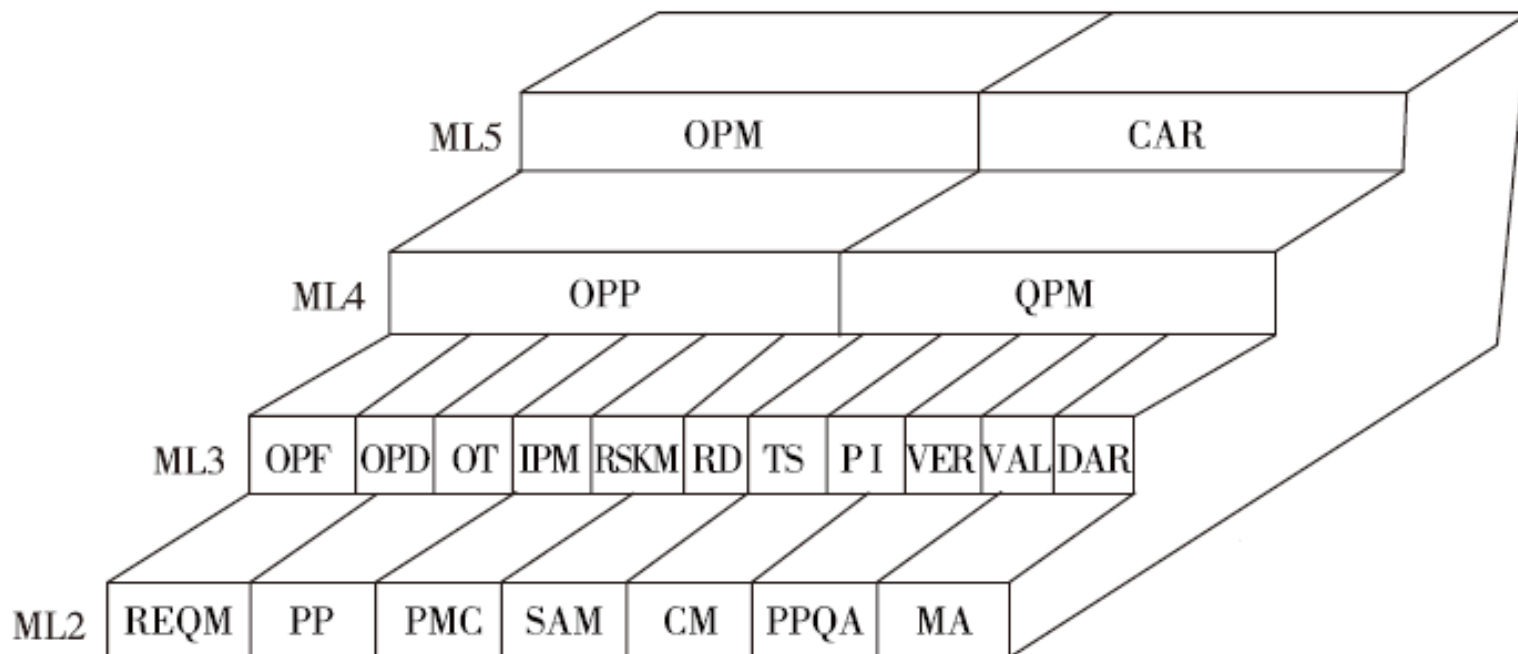
• 两种表示的对比

两种表示的特征比较

	分级式表示	连续式表示
改进路径	CMMI 模型为组织规定了过程改进的路径，即沿着 ML1 至 ML5 逐步提升	组织可根据自身业务目标的需要和风险控制要求自行决定改进路径
关注点	集中关注一组与目标等级相关的过程，即模型规定的过程域	只需关注软件组织选定的若干过程域，各过程域间并无必然的关联
改进成果的体现	以简明的形式即成熟度等级概括地表达了过程改进的成果	以软件组织选定的各过程域的能力等级表达
组织间比较	容易比较	只能针对特定的过程域逐个比较
成果差别	给出的是组织的过程成熟度等级	所选各过程域的能力等级

12.4.4 CMM1.3简介

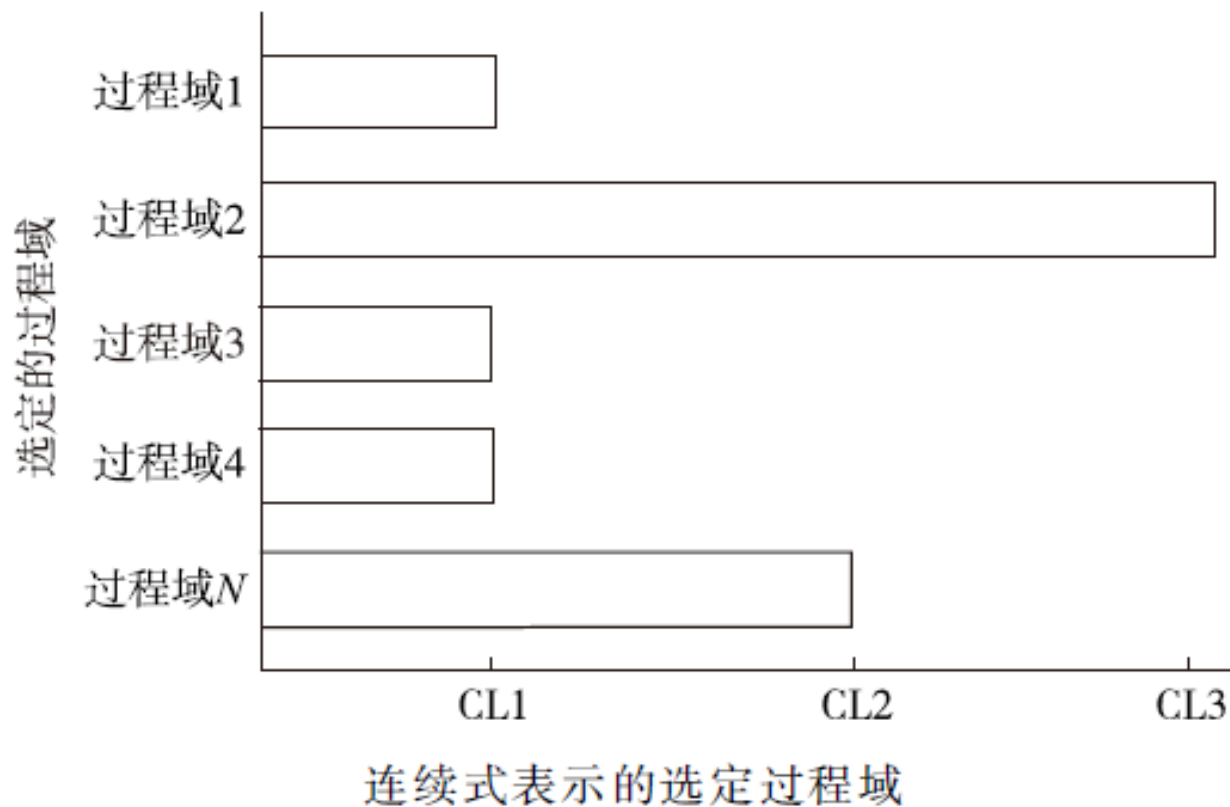
- 两种表示的对比



分级式表示的成熟度等级

12.4.4 CMM1.3简介

- 两种表示的对比



12.4.4 CMM1.3简介

- **成熟度等级与能力等级**

- (1) 成熟度等级 (ML , Maturity Level)**

- **初始级 (ML1)**
- **已管理级 (ML2)**
- **已定义级 (ML3)**
- **已量化管理级 (ML4)**
- **优化级 (ML5)**

12.4.4 CMM1.3简介

- **成熟度等级与能力等级**

- (2) 能力等级 (CL , Capability Level)**

- **不完备级 (CLO)**
- **已实施级 (CL1)**
- **已管理级 (CL2)**
- **已定义级 (CL3)**
- **定量管理级 (CL4)**
- **优化级 (CL5)**

12.4.4 CMM1.3简介

- **过程域**
 - **所谓过程域 (Process Area) 是CMMI为实施软件过程改进的组织而提出的若干个值得重视的软件过程。**
 - **尽管CMMI两种表示在过程改进的路径有所不同，但这些过程域毕竟都是改进工作需要特别给予关注的工作焦点。**
 - **CMMI V1.3共提出了22个过程域，为了便于理解和实施，将其作了分类和分级。**

12.4.4 CMM1.3简介

• 过程域

CMMI DEV 1.3 过程域分类及成熟度等级

成熟度等级 过程域分类	ML2 (二级)	ML3 (三级)	ML4 (四级)	ML5 (五级)	过程域数
过程管理 (Process Management)		组织过程焦点 (OPF)	组织过程性能 (OPP)	组织性能管理 (OPM)	5
		组织过程定义 (OPD)			
		组织培训 (OT)			
项目管理 (Project Management)	需求管理 (REQM)	集成项目管理 (IPM)	定量项目管理 (QPM)		7
	项目策划 (PP)				
	项目监督和控制 (PMC)				
	供应商协议管理 (SAM)	风险管理 (RSKM)			
工程 (Engineering)		需求开发 (RD)			5
		技术解决方案 (TS)			
		项目集成 (PI)			
		验证 (VER)			
		确认 (VAL)			

12.4.4 CMM1.3简介

- 过程域

过程域的分类、分级明细表

(续)

成熟度等级 过程域分类	ML2 (二级)	ML3 (三级)	ML4 (四级)	ML5 (五级)	过程域数
支持 (Support)	配置管理 (CM)	决策分析和解决方案 (DAR)		因果分析和解 决方案 (CAR)	5
	过程和产品质量保证 (PPQA)				
	度量和分析 (MA)				
过程域数	7	11	2	2	22 (总计)

12.4.4 CMM1.3简介

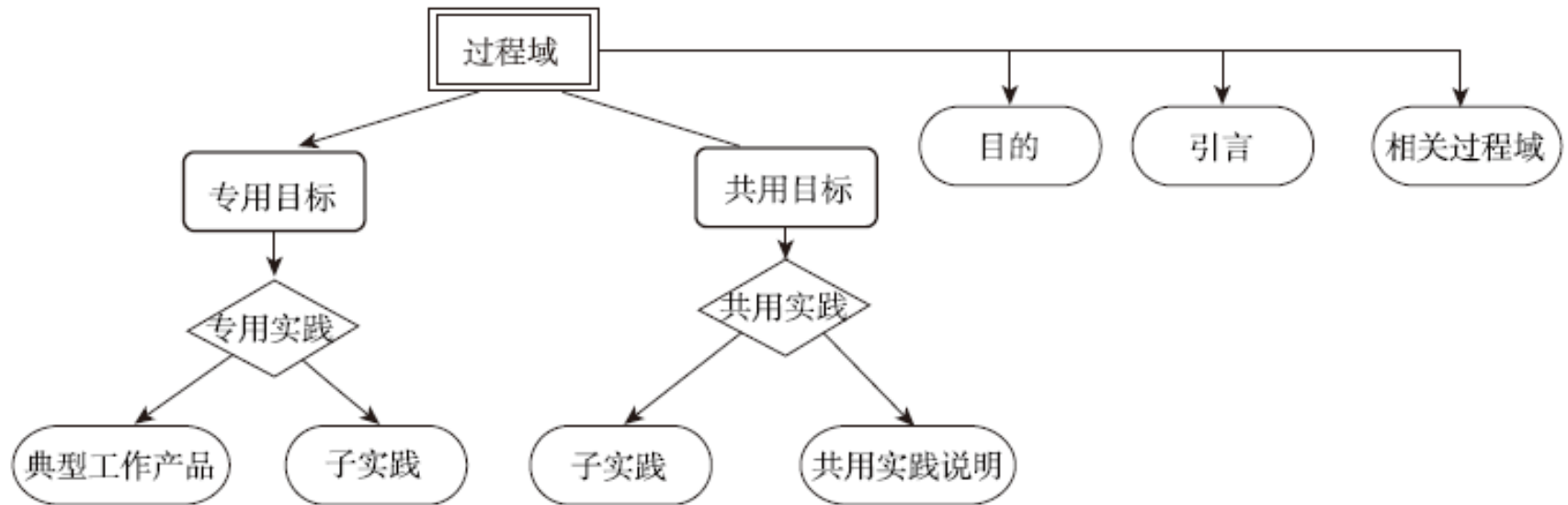
- 两种目标、两种实践

(1) 模型部件 (Model Components)

- 模型部件也称为过程域部件。它表明，CMMI提出的过程域内容将被分割成为部件并加以区分，这样可方便CMMI的用户，对于过程域的内容给予不同程度的关注。
- 在CMMI的过程域描述中包含了三个模型部件：必需的部件、期望的部件及资料性部件。

12.4.4 CMM1.3简介

• CMMI的模型部件



图例：

□ 表示必需部件 ◇ 表示期望部件 ○ 表示资料性部件

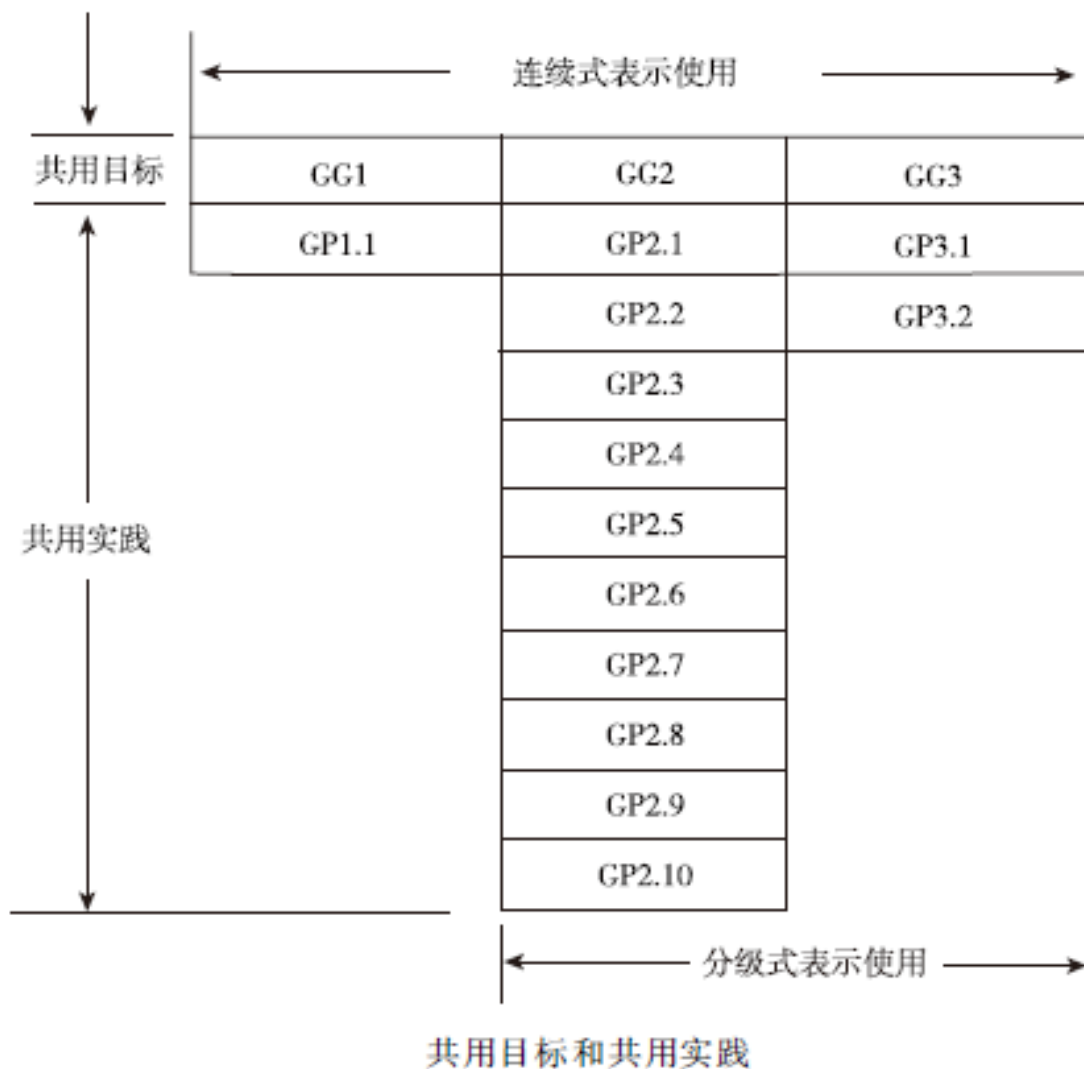
CMMI 过程域的模型部件

12.4.4 CMM1.3简介

(2) 共用目标和共用实践

- **CMMI设置了5个共用目标（GG，Generic Goals），这些目标体现了对过程域不同等级的要求，它们都是必需的部件。**
- **为达到共用目标就要有相应的措施，这些措施就是每个共用目标所对应的一组共用实践（GP，Generic Practices）。**

12.4.4 CMM1.3简介



过程域“项目策划”的专用目标和专用实践

专用目标		专用实践	
SG1	建立并维护对项目计划参数的估算	SP1.1	建立顶层的工作分解结构（WBS），以估算项目范围
		SP1.2	建立并维护对工作产品和任务属性的估算
		SP1.3	定义项目的生命周期阶段，以界定计划工作范围
		SP1.4	估算工作产品和任务所需的项目工作量与成本
SG2	制定并维护项目计划，以其作为管理项目的基础	SP2.1	制定并维护项目的预算和进度
		SP2.2	识别和分析项目风险
		SP2.3	为项目的数据管理制定计划
		SP2.4	为获得实施项目所需的资源进行策划
		SP2.5	为获得实施项目所需的知识和技能进行策划
		SP2.6	为项目相关人员的参与制定计划
		SP2.7	制定并维护项目的总体计划
SG3	取得对计划的承诺	SP3.1	评审项目相关的各项计划
		SP3.2	调整项目计划，以协调可用的资源与估计的资源
		SP3.3	获得实施与支持项目计划的相关人员的承诺

12.4.5 CMMI评估

- **标准评估方法**
- **软件组织实施CMMI的意义在于用它来指导过程改进，而实施的情况以及过程改进的成效都需要通过评估加以检验。为此CMMI的创始组织卡内基·梅隆大学的软件工程研究所（CMU SEI）制定了标准的CMMI评估方法SCAMPI（Standard CMMI Appraisal Method for Process Improvement）。**
- **制定SCAMPI的目的在于确保评估的一致性，即要求对不同的被评估组织，在多次评估中其结果是相同的，例如达到某个特定的成熟度等级或是满足某个过程域的能力等级特征。**

12.4.5 CMMI评估

- **评估原则**
 - **评估工作应由组织的高层管理者主持。实践表明，这是成功评估的关键。**
 - **关注于组织制定的业务目标。**
 - **评估过程中重视客观证据的收集，包括与被评估组织各类人员访谈的情况以及过程文档的相关信息。**
 - **对评估信息应予保密。**
 - **使用CMMI模型作为评估的依据。**
 - **评估组成员协调配合工作，妥善地处理意见分歧，最终应给出一致的结论意见。**
 - **始终着眼于过程的改进。**

12.4.5 CMMI评估

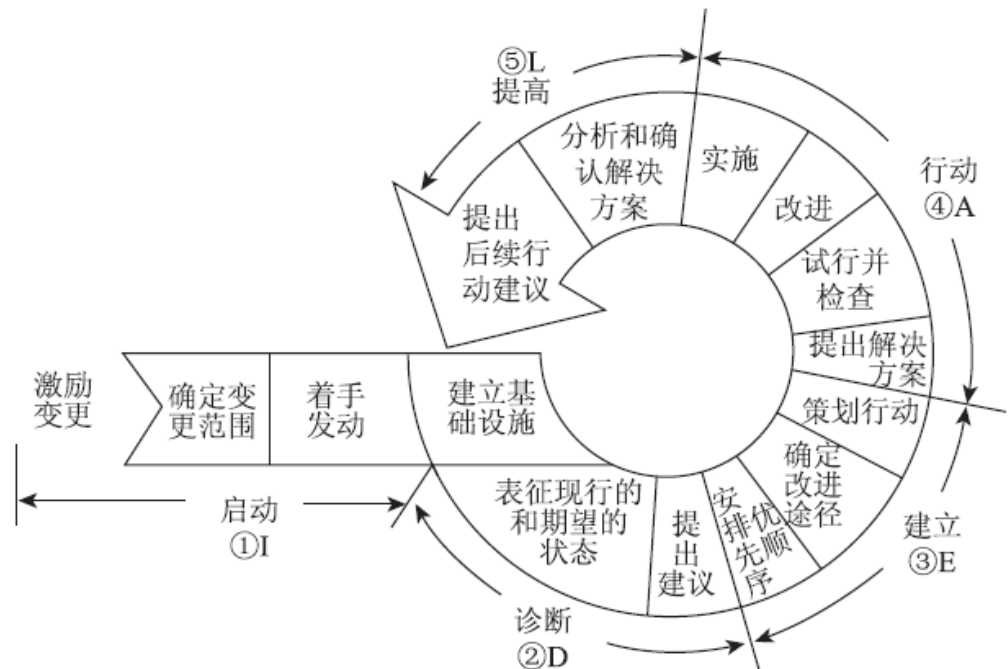
- 评估实施

- 对软件组织实施评估需要考虑以下的问题：

- 确定评估的范围。这包括确定组织中哪些部门、哪些项目参加评估，以及选定CMMI的哪种表示（例如，选定针对哪些过程域的什么能力等级，或是哪个成熟度等级）。
- 选定评估实施的等级A、B或C。A级评估最为严格，最为广泛（不仅要考察过程的定义、试点工作、推广工作、还要考察制度化的情况），B级次之，C级要求最低。
- 确定评估组成员，视实际需要评估人员进行培训。
- 确定被评组织参加访谈的人员。
- 确定评估要得到的结论形式。
- 制定评估的约束条件，如评估时间和评估地点的要求。
- 制定评估实施计划。

12.5 软件过程改进

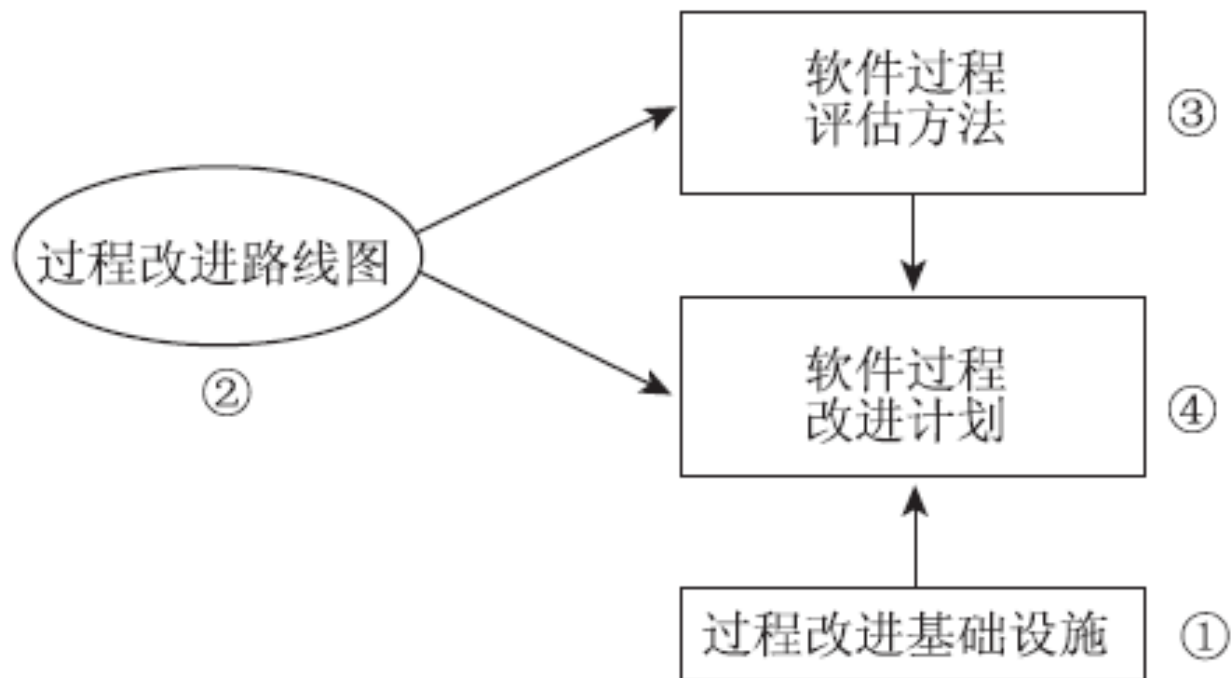
- 软件过程改进的**IDEAL**模型
- 美国卡内基·梅隆大学的软件工程研究所（SEI）在总结了软件组织实施过程改进的大量经验和教训后，提出了一个十分具有指导意义的IDEAL模型，为开展过程改进的组织提供了有益的帮助。



软件过程改进的 IDEAL 模型

12.5 软件过程改进

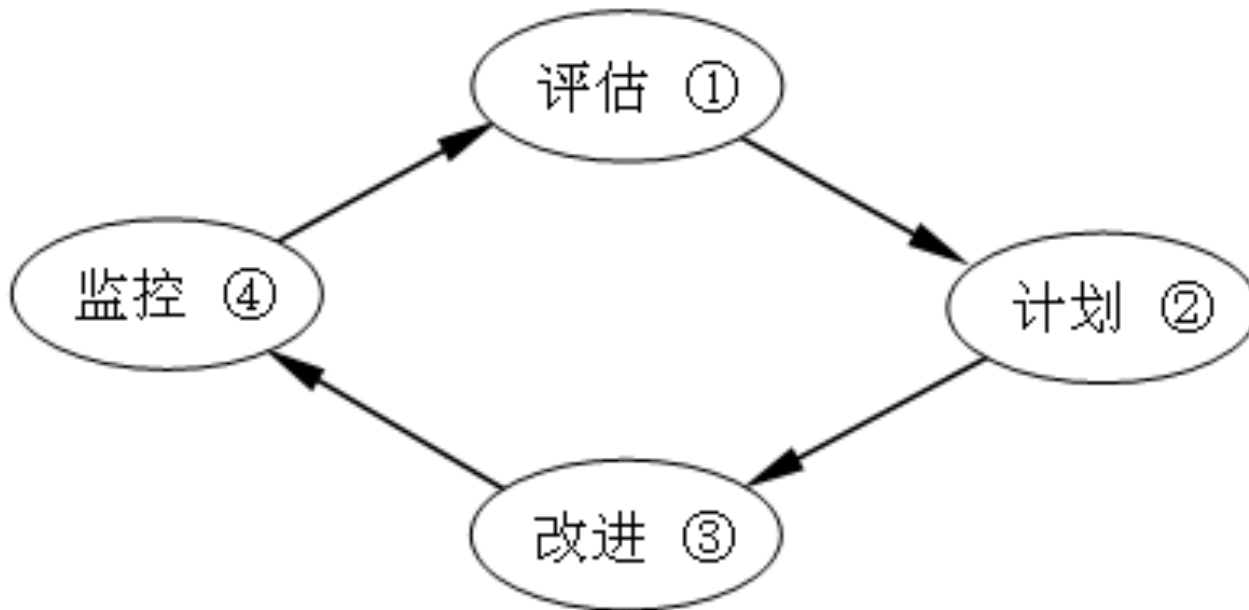
- 软件过程改进框架



软件过程改进框架

12.5 软件过程改进

- 软件过程改进循环





That's All!