



软件质量管理

彦哲研究院

李炳森

课程提纲

一、质量与质量管理

二、质量管理体系

三、项目质量管理

四、软件质量

五、软件质量保证

课程提纲（续）

六、软件配置管理

七、软件质量度量

八、软件质量标准

九、软件评审

十、软件全面质量管理

一、质量与质量管理

1.1 质量的重要性

1.2 质量的含义与理解

1.3 质量管理的定义

1.4 质量管理的发展简史

1.5 质量管理大师



一、质量与质量管理

1.1 质量的重要性

发展是硬道理，
质量是发展的硬道理。

1.1 质量的重要性

美国质量管理专家哈林顿
(H.J.Harrington)说，这不是一场
使用枪炮的战争，而是一场商
业战争，战争中的主要武器就
是产品质量。



一、质量与质量管理

1.1 质量的重要性

1.2 质量的含义与理解

1.3 质量管理的定义

1.4 质量管理的发展简史

1.5 质量管理大师

一、质量与质量管理

1.2 质量的含义与理解

◎ 定义：一组 **固有特性** 满足 **要求** 的程度。

指某事或某物中本来就有的特性，如：螺栓的直径、机器的生产率等

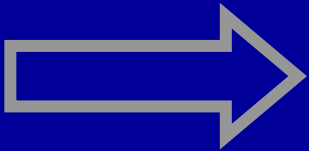

指明示的、通常隐含的或必须履行的需求或期望。

指规定的要求

指法律法规或强制性标准要求

指组织、顾客和其他相关方的惯例或一般做法

◎质量具有的“三性”

- 质量的广义性
-  质量的时效性
-  质量的相对性

质量的优劣是满足要求程度的一种体现。
它必须在同一等级基础上做比较。

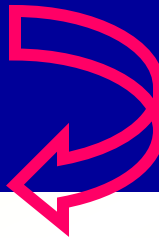
1.2 质量的含义与理解

- 质量不仅包括活动或过程的结果，还包括它们形成和实现的活动及过程本身；
- 质量不仅包括产品质量，还包括它们形成和实现过程中的工作质量；
- 质量不仅要满足顾客的需要，还要满足社会的需要，并使顾客、从业人员、业主、供方和社会都收益；（stakeholder，受益者）
- 质量问题不仅存在于工业，还存在于服务业及其他各行各业。



1.2 质量的含义与理解

与质量相关的六大概念细解



➤ 组织：

指职责、权限和相互关系得到安排的一组人员及设施。

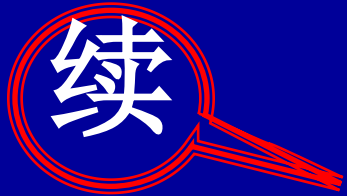
➤ 过程：

指一组将输入转化为输出的相互关联或相互作用的活动。
由输入、实施、输出三个环节。

➤ 产品：

指过程的结果。（服务、软件、硬件、流程性材料）

1.2 质量的含义与理解



➤ 顾客：

指接受产品的组织或个人。有内部顾客和外部顾客。

➤ 质量特性

指产品、过程或体系与要求有关的固有特性。（关键质量特性、重要质量特性、次要质量特性）

➤ 体系：

指相互关联或相互作用的一组要素。

一、质量与质量管理

1.1 质量的重要性

1.2 质量的含义与理解

1.3 质量管理的定义

1.4 质量管理的发展简史

1.5 质量管理大师

1.3 质量管理的定义



1.3 质量管理的定义

■质量目标：在质量方面所追求的目的，是对质量方针的展开。

定义项目目标：

S—具体

M—可量化

A—可行性

R—相关

T—基于时间

1.3 质量管理的定义

是
质
量
管
理
的
一
部
分

质量策划

致力于制定质量目标并规定必要的运行过程和相关资源，以实现质量目标

质量保证

致力于提供质量要求会得到满足的信任。

质量控制

致力于满足质量要求

质量改进

致力于增强满足质量要求的能力

1.3 质量管理的定义

◎ 质量管理的任务

- ❑ 正确制订和贯彻执行质量方针和政策；
- ❑ 保证和提高产品质量和服务质量，生产出物美价廉的产品，以满足用户需要；
- ❑ 不断降低物质消耗，降低质量成本和提高经济效益；
- ❑ 提高领导和职工的质量意识和素质，促进企业素质和管理水平的提高；
- ❑ 研究和发展质量理论和质量科学。

一、质量与质量管理

1.1 质量的重要性

1.2 质量的含义与理解

1.3 质量管理的定义

1.4 质量管理的发展简史

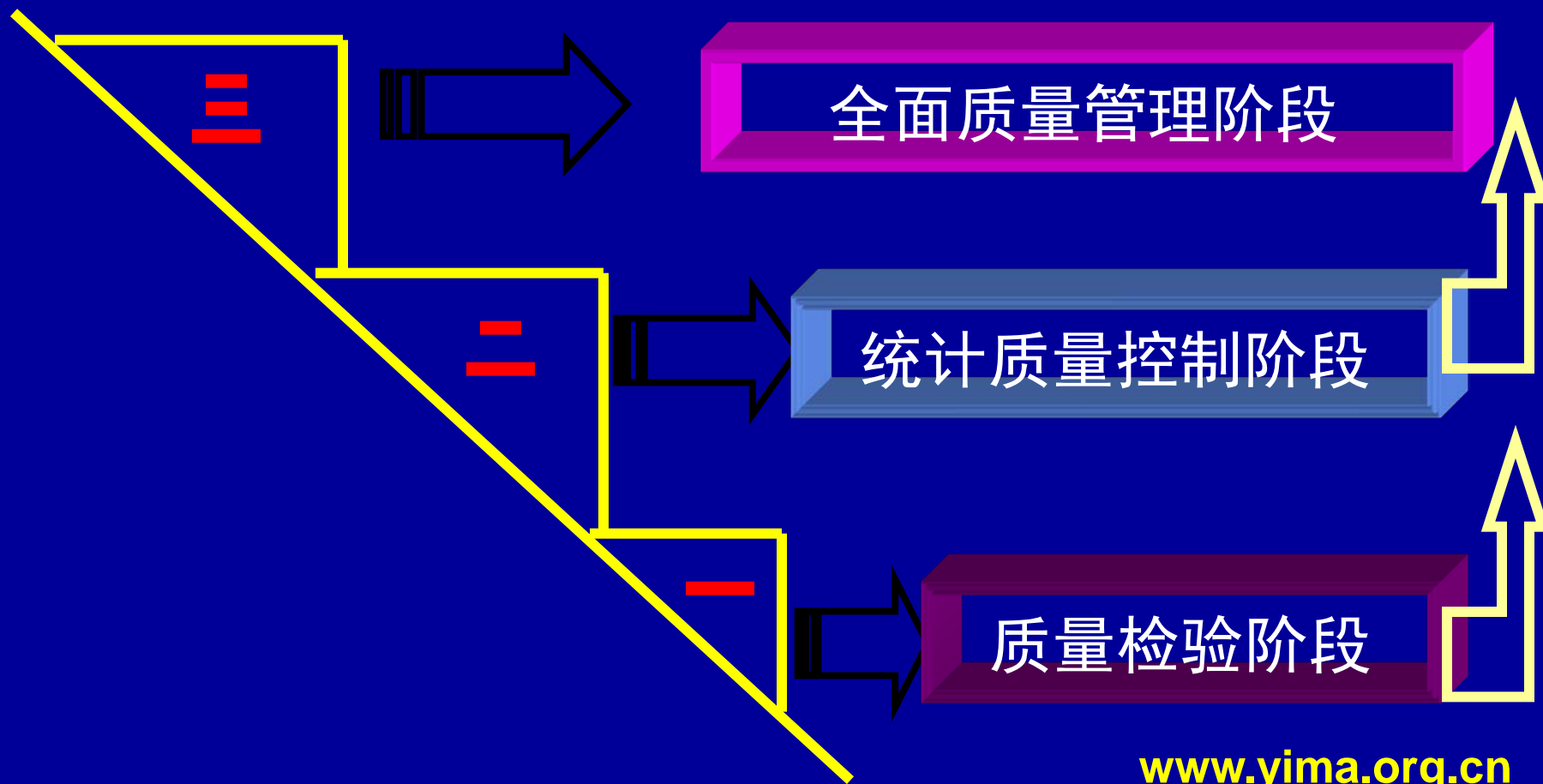
1.5 质量管理大师

1.4 质量管理的发展简史

1. 质量检验阶段
2. 统计质量控制阶段
3. 全面质量管理阶段

1.4 质量管理的发展简史

质量管理的发展

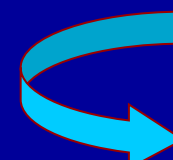



质量检验阶段（二十世纪初）



在成品中挑废品，这是事后把关，无法起到预防和控制作用

统计质量控制阶段（二十世纪30/40年代）



把数理统计与质量管理结合，由休哈特发明了控制图。但过分强调质量控制的统计方法，在当时计算机和软件的应用不广泛，难度很大。

全面质量管理阶段（二十世纪50年代后）



经过不断地发展，演变为一套以质量为中心，综合的、全面的管理方式和管理理念 www.yima.org.cn

三阶段的比较

- 1 从管理的深度:

单纯检验——检验与预防——控制与提高

- 2 从管理的广度:

前两个阶段局限在制造过程;

第三阶段向前延伸到设计与试制过程, 向后延伸到使用过程。

1.4 质量管理的发展简史

项目	一	二	三
生产特点	手工半机械化	大量生产	现代化大生产
质量概念	狭义质量	向广义质量过渡	广义质量
管理范围	检验	制造过程	全过程
管理对象	产品	产品和工序质量	产品和工作质量
管理依据	质量标准	质量标准、 控制标准	用户需要
管理方法	技术检验方法	数理统计方法	运用一切有效手段
参加人员	检验人员	技术部门、 检验人员	企业全体员工

一、质量与质量管理

1.1 质量的重要性

1.2 质量的含义与理解

1.3 质量管理的定义

1.4 质量管理的发展简史

1.5 质量管理大师

1.5 质量管理大师

质量管理专家的质量理念

戴明的质量理念

美国著名的质量专家之一，他认为引起效率低下和不良质量原因在管理系统而不在员工。他总结出了14条质量管理原则。

朱兰的质量理念

美国的著名质量专家，他认为质量是指那些能满足顾客需求，从而使顾客满意的“产品特性”；质量意味着无缺陷。朱兰质量管理三部曲：质量策划、质量控制、质量改进。

石川馨的质量理念

日本著名的质量管理专家，发明了“因果图”，他认为质量不仅仅是产品质量，从广义上说，还有工作质量、部门质量、人的质量、体系质量等，推行全面质量管理是公司全部门、全员参与的综合性质量管理。

费根鲍姆

前通用电器生产和质量控制经理，他因提出把质量责任推广到生产领域以外而在质量运动中闻名。

在推行“缺陷成本”方法方面起重要作用。根据这一方法，管理应服从于质量。

二、质量管理体系

2.1 质量管理体系基础

2.2 八项质量管理原则

二、质量管理体系知识

2.1 质量管理体系基础

定义

由ISO/TC176技术委员会制定的所有的国际标准称为ISO9000族质量管理体系标准。

质量方针和质量目标

- 建立质量方针和质量目标为组织提供了关注焦点。两者确定了预期的结果。并帮助组织利用其资源达到这些结果。
- 质量方针为建立和评审质量目标提供了框架。
- 质量目标需要与质量方针和持续改进的承诺相一致，其实现需要是可测量的。



2.1 质量管理体系基础

质量管理体系中使用的文件类型

- 1. 质量手册;
- 2. 质量计划;
- 3. 规范;
- 4. 指南;
- 5. 图样、作业指导书、程序;
- 6. 记录;

文件的多少和详略，取决于组织的类型、规模、过程和产品的复杂性、顾客要求、适用的法律法规要求等。

质量管理体系评价

在评价质量管理体系时，应对每一个评价的过程提出如下四个基本问题：

1. 过程是否被识别并作适当规定？
2. 职责是否被分配？
3. 程序是否得到实施和保持？
4. 在实现所有要求的结果方面，过程是否有效？

评价的形式：

质量管理体系审核、质量管理体系评审、自我评定

质量管理体系审核

1. 第一方审核：

用于内部目的，由组织自己或以组织的名义，可作为组织自我合格声明的基础。

2. 第二方审核：

由组织的顾客或由他人以顾客的名义进行。

3. 第三方审核：

由外部独立的组织进行。这类组织通常是经认可的，提供符合要求的认证或注册。

审核的目的：评价质量管理体系的符合性，评价质量管理体系的有效性，识别改进的机会。

质量管理体系认证的主要活动内容

1. 认证申请与受理

2. 审核的启动

3. 文件评审

4. 现场审核的准备

5. 现场审核的实施

6. 审核报告的编制、
批准和发布

7. 纠正措施的验证

颁发认证证书

监督审核和复评

二、质量管理体系

2.1 质量管理体系基础

2.2 八项质量管理原则



二、质量管理体系知识

2.2 八项质量管理原则

TC/176花了1年的时间埋头整理和研究，参考了QS9000 许多理念，总结和归纳出质量管理八大原则：

- ☆ 以顾客为关注焦点
- ☆ 系统管理
- ☆ 领导作用
- ☆ 持续改善
- ☆ 全员参与
- ☆ 基于事实的决策方法
- ☆ 过程方法
- ☆ 互利供方关系

被誉为二十一世纪所有提供产品和服务的组织指导质量管理的八大法宝。

原则一： 以顾客为关注焦点—— 组织存在的根本宗旨

- 了解现在和未来顾客的需求
- 满足顾客的需求
- 争取超越顾客的期望

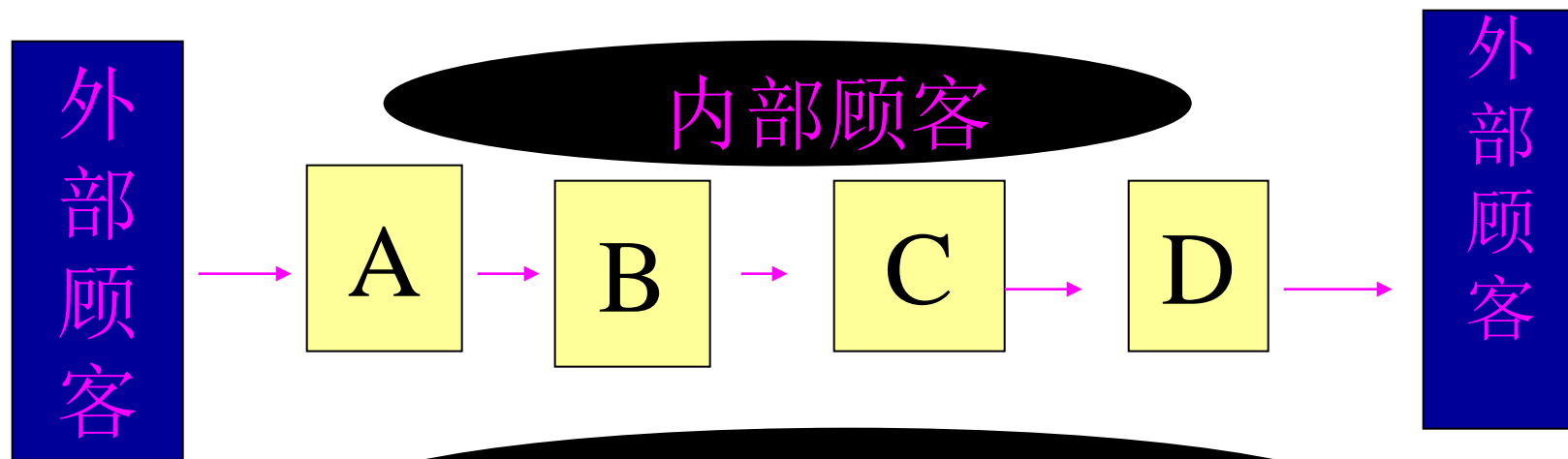


企业应

1. 平衡顾客和其他利益相关者的需要
2. 使这些需求和期望在企业内得到充分的沟通和协调
3. 测量顾客的满意度，对与顾客的关系进行管理

讨论：顾客的含义

- 我们的供货对象——外部顾客
- 外部顾客还可分为不同层次，如：代理商，中间商，最终消费者。
- 内部的部门间/工序间——内部顾客



可以理解为部门间或工序间

处理部门间、工序间工作关系的原则

☆ 下一工序就是顾客

☆ 以顾客为中心

☆ 每个部门既是供应商又
是顾客

(内部顾客满意度测评)

原则二： 领导作用 (Leadership)

---质量管理体系持续有效运作的根本保证

- ☆领导者在企业建立统一的意图
- ☆领导者指明企业统一的方向，好比企业大船的船长
- ☆领导者须建立一个让员工充分投入去完成企业目标的环境
最重要的是坚持

以顾客为关注焦点的宗旨

领导者如何发挥领导作用？

作好企业的领路人

与员工建立互信的关系

原则三：全员参与 -----实现以顾客为关注焦点的前提条件之一

- 各阶层员工是企业之本
- 只有他们的充分参与，才能使他们的才干为企业带来最大的收益

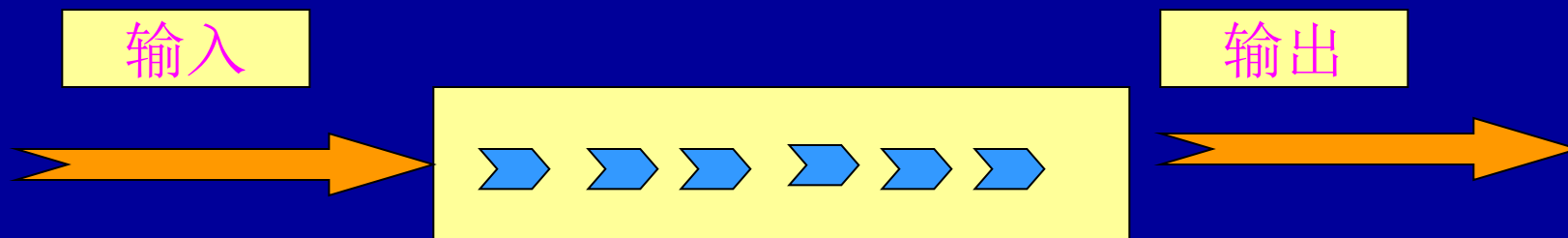
员工应该做...？

原则四：过程方法

-----提高管理效率的有效手段

- ISO9001: 2000 把过程定义为：

通过使用资源和管理，将输入转化为输出的一组活动。



2.2 八项质量管理原则

过程的特点

- 一定包含输入和输出

输入：我们要工作的对象或处理的对象（针对输出提出的要求），

输出：过程最后的结果（符合要求的成果）

- 必须运用适当的资源（人员、设备、场地、时间、信息等）

- 过程必须由若干个多个子过程或流程组成

如：对供应链来说，企业就是一个大过程

但对企业内部来说，这个大过程可以依顺序细分为若干个子过程，

如设计开发过程就是产品实现过程中的一个是一个子过程：

- 前一个过程的输出就是下一个过程的输入
- 过程必须考虑增值（满足要求）

过程方法要点

第一步：确定输入与输出

第二步：识别流程

第三步：识别职责

第四步：识别流程

第五步：设计监控点

原则五：系统的管理方法

- 针对设定的目标，识别、理解并管理一个相互关联的过程所组成的体系，有助于提高组织的有效性和效率。

原则六：持续改进

持续改进是组织的永恒目标

ISO9001: 2000标准
就是根据持续改善的原则编写的

持 续 改 进

*持续改进质量管理体系的目的：增加顾客和其他相关方满意的机会。

案例关键词：丰田精神。

持续改进

系统性的改进：*对质量方针、目标的评审与修订

*审核结果

* 数据分析

*管理评审

日常性的改进：*纠正与预防措施的实施

*质量例会

原则七：基于事实的决策方法

对数据和信息的逻辑分析或直觉判断，是有效决策的基础

要求

- 测量和搜集数据
- 分析
- 使用统计技术
- 基于事实进行决策

原则八：互利的供方关系

- 通过互利的关系，提升组织及其供方创造价值的的能力

要求

与供方的伙伴关系是指：

- ☆ 识别和选择主要的供应商
- ☆ 建立伙伴关系
- ☆ 充分的信息沟通，共享信息
- ☆ 共同改进产品和过程
- ☆ 认可供应商的改进和成就

三、项目质量管理

3.1 项目管理知识体系

3.2 质量策划

3.3 质量保证

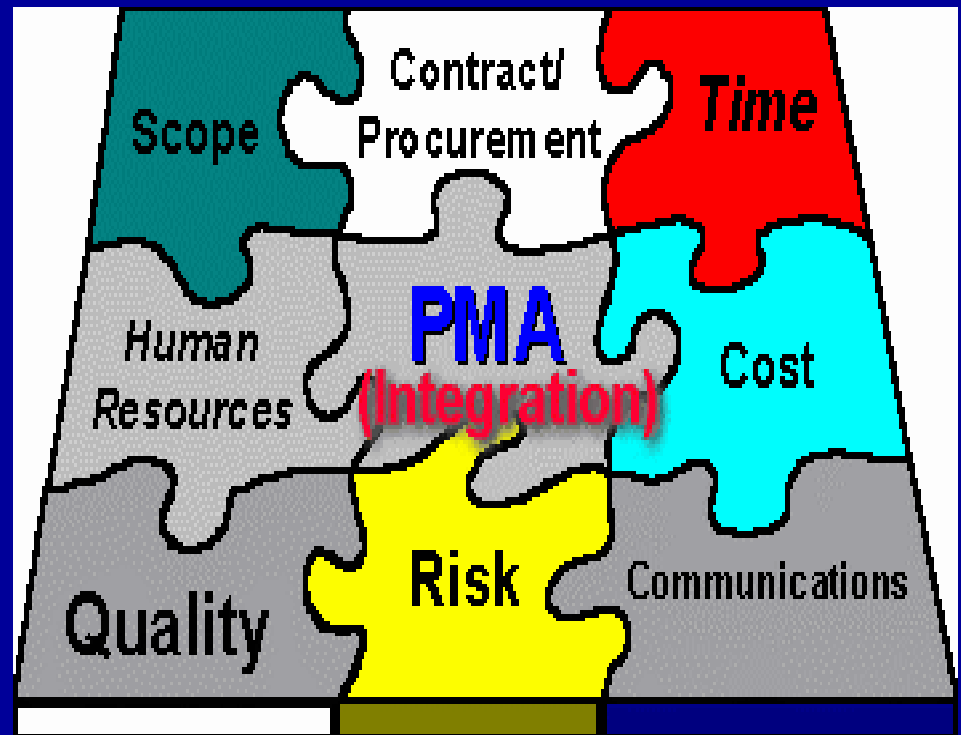
3.4 质量控制

三、项目质量管理

3.1 项目管理知识体系

PMI将项目管理知识体系分为10大知识领域：

1. 项目集成管理
2. 项目范围管理
3. 项目时间管理
4. 项目成本管理
5. 项目质量管理
6. 项目人力资源管理
7. 项目沟通管理
8. 项目风险管理
9. 项目采购管理
10. 项目干系人管理



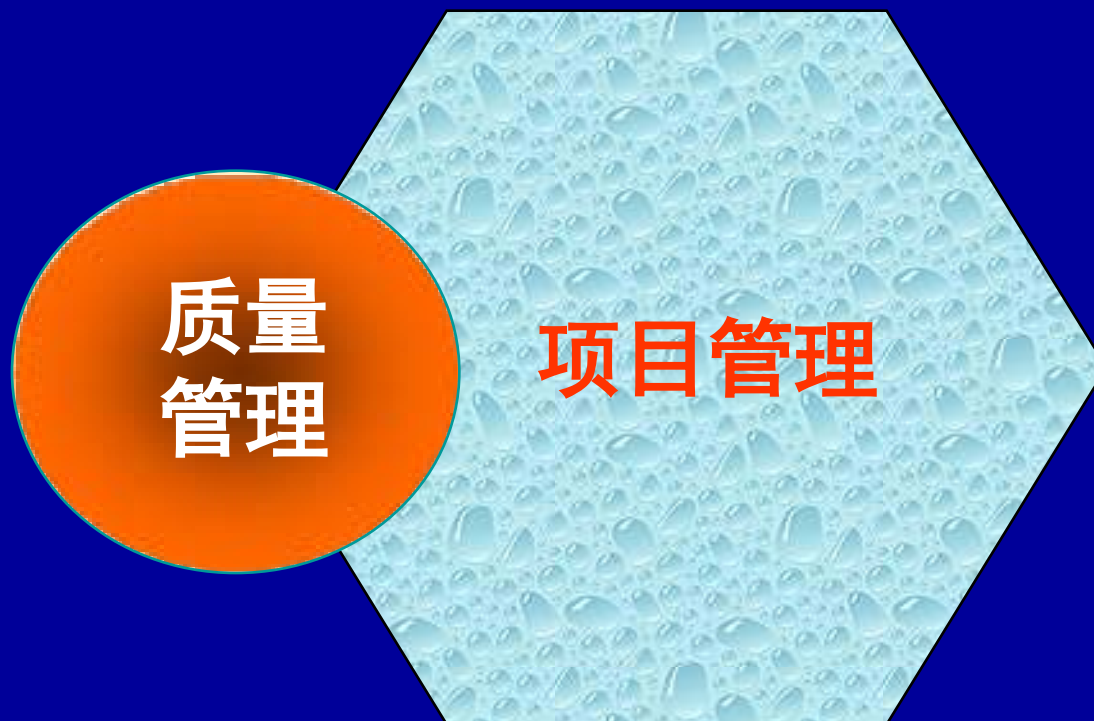
◎ 项目管理的定义：

项目管理就是将各种知识、技能、工具和技术应用于项目之中，以达到项目的要求。

项目管理就是组织、计划、管理任务和调度资源并在有限的时间和资金的前提下去完成一个特定的目标。

简单地说，就是以项目为手段进行管理。

◎ 质量管理与项目的关系



3.1 项目管理知识体系

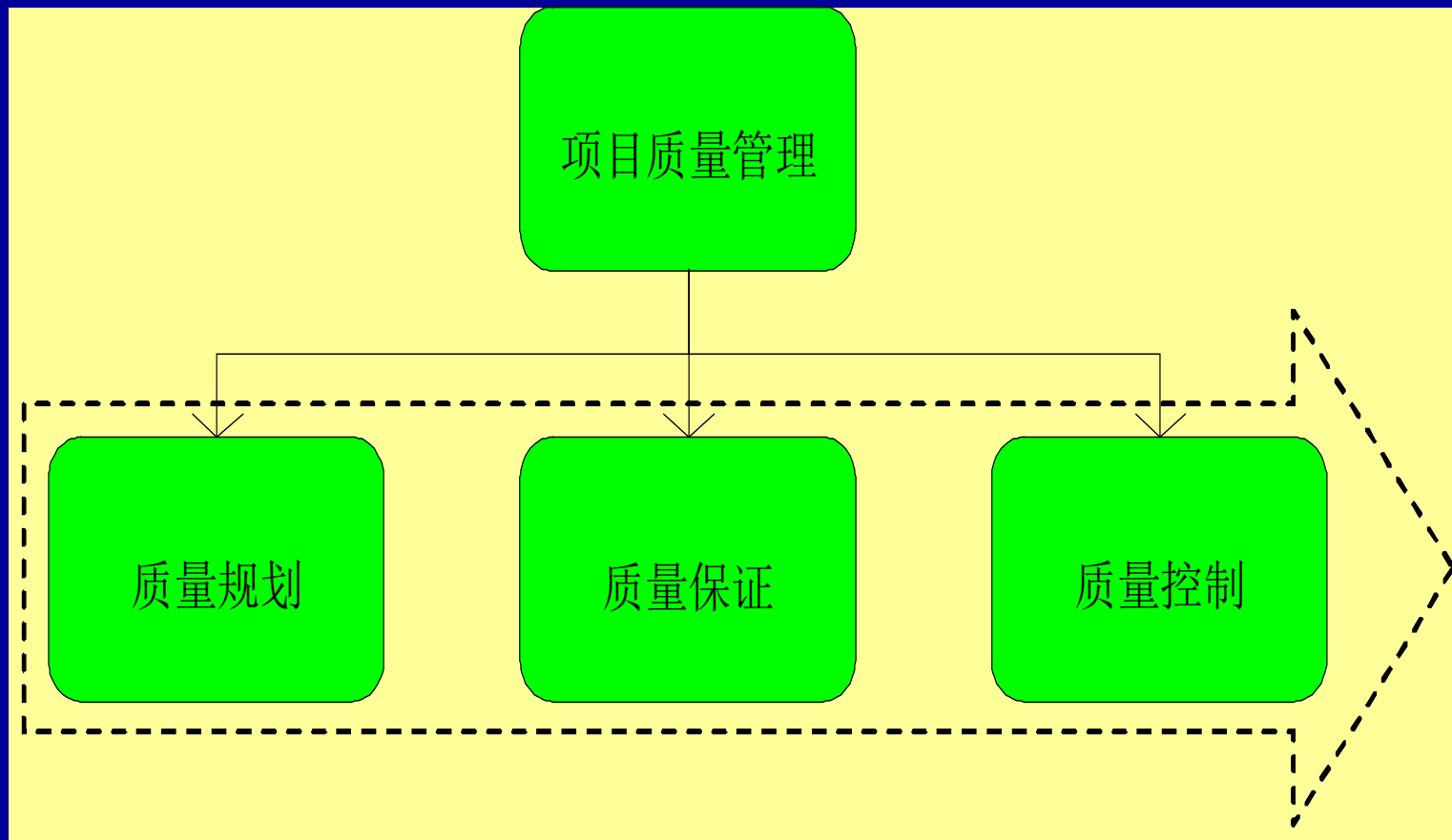
项目质量管理的概念

为保障项目产出物能够满足项目业主、客户以及项目其他相关利益者的需要，所开展的对于项目产出物质量和项目工作质量的全面管理工作。

包括：

- 项目质量方针的确定
- 项目质量目标和质量责任的制定
- 项目质量体系的建设
- 为实现项目质量目标所开展的项目质量计划、项目质量控制和项目质量保障等一系列质量管理工作

3.1 项目管理知识体系



项目质量管理

3.1 项目管理知识体系

项目质量管理思想

- 全面质量管理思想-质量管理的全员性、全过程性、全要素性。
- 项目质量管理的几个重要理念：
 - 使顾客满意是项目质量管理的目的
 - 项目质量是干出来的，不是检验出来的
 - 项目质量管理责任是全体团队成员的
 - 项目质量管理的关键是不断监控和改进
 - 项目质量确定与项目质量实施
 - 项目质量等级与项目质量好坏

三、项目质量管理

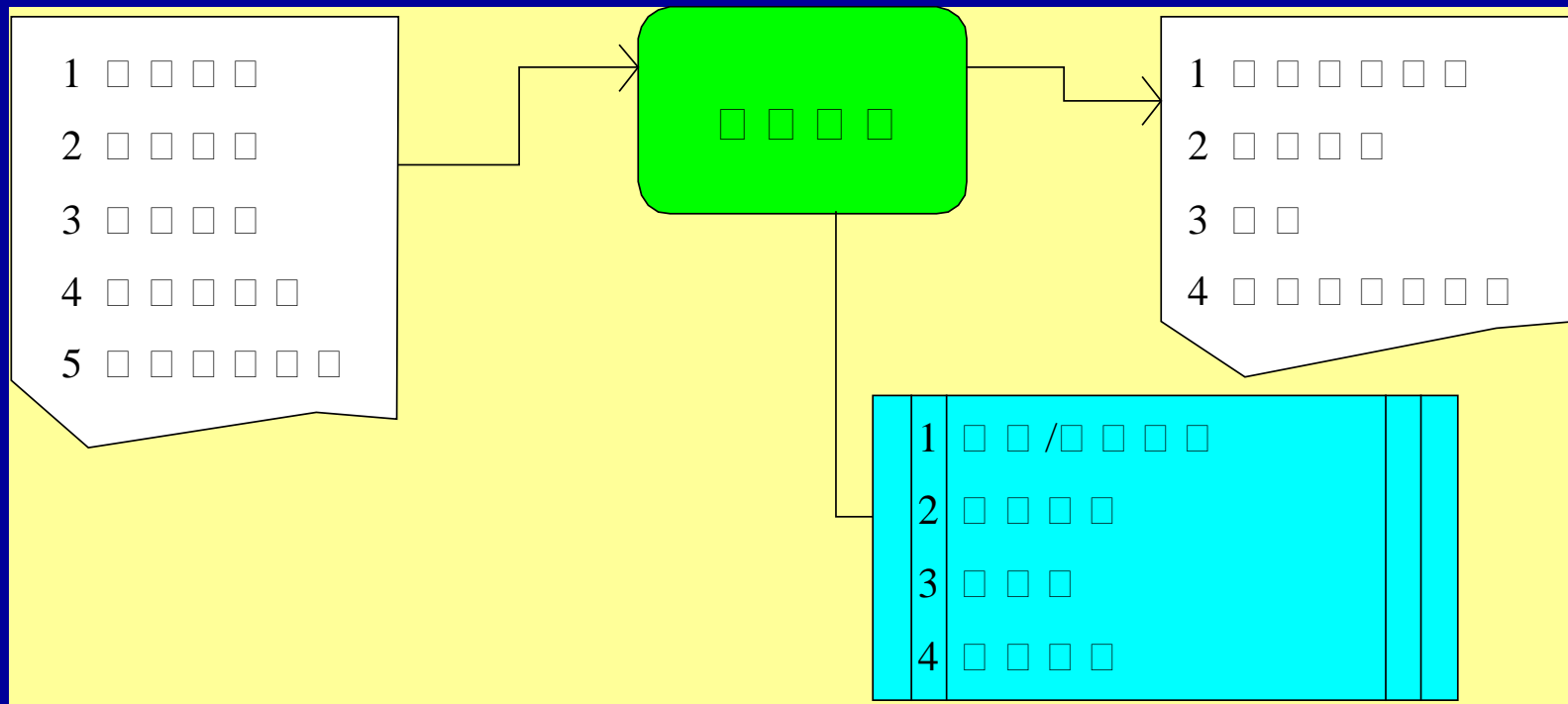
3.1 项目管理知识体系

3.2 质量策划

3.3 质量保证

3.4 质量控制

3.2 质量策划



质量规划

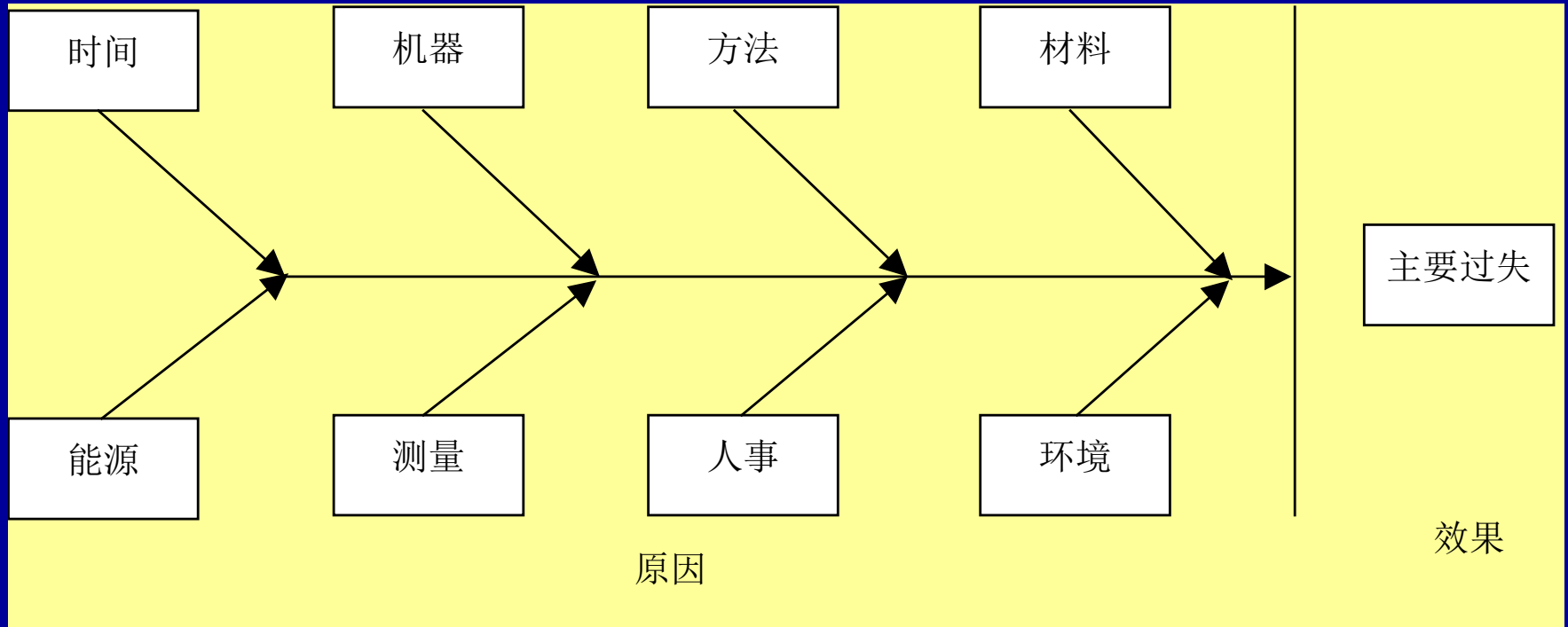
质量规划的输入元素

1. 质量政策。
2. 范围陈述。
3. 产品描述。
4. 标准和规章。
5. 其它过程输出。

质量规划的工具和技术

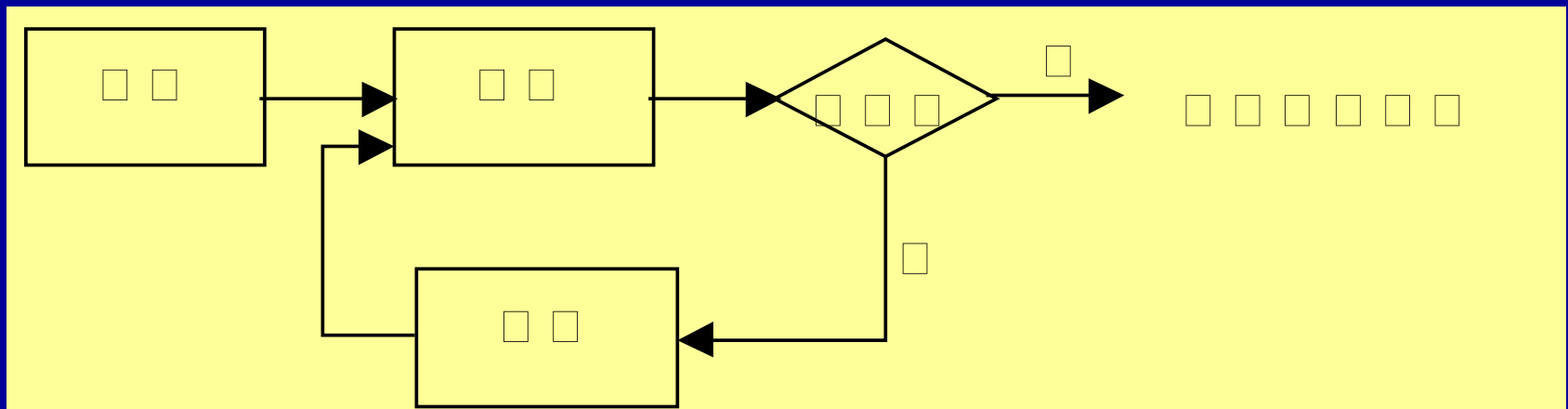
1. 效益/成本分析。
2. 基准比较。
3. 流程图。
4. 实验设计。

3.2 质量策划



因果图

3.2 质量策划



过程流图示例

质量规划的输出元素

1. 质量管理计划。
2. 操作定义。
3. 清单。
4. 给其它过程的输入。

三、项目质量管理

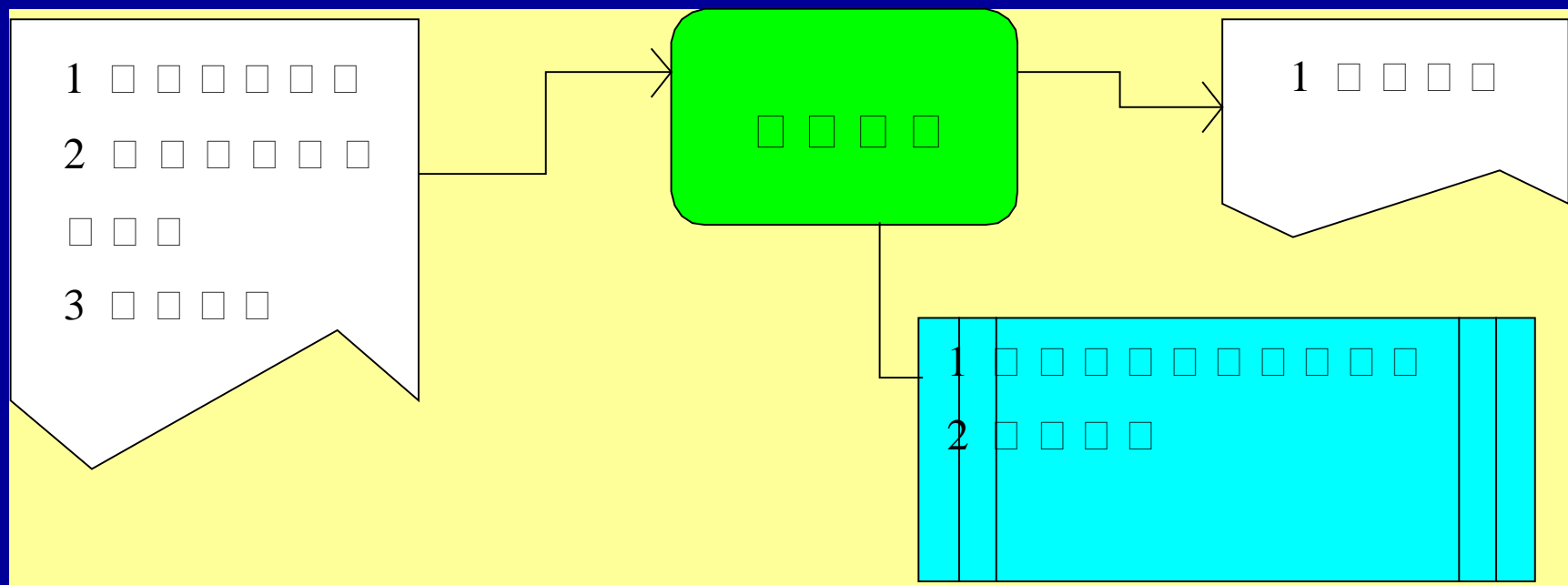
3.1 项目管理知识体系

3.2 质量策划

3.3 质量保证

3.4 质量控制

3.3 质量保证



质量保证

质量保证的输入元素

1. 质量管理计划。
2. 质量控制测量的结果。质量控制测量是按照便于比较和分析的格式记录的质量测试和测量的记录。
3. 操作定义。

质量保证的工具和技术

1. 质量规划的工具和技术。质量规划的工具和技术也能够被用于质量保证。
2. 质量审计。质量审计是对其它质量管理活动的结构化评价。

质量保证的输出元素

质量改进。

质量改进包括采取行动增加项目的效益和效率，向项目权益相关者提供额外的利益。

三、项目质量管理念

3.1 项目管理知识体系

3.2 质量策划

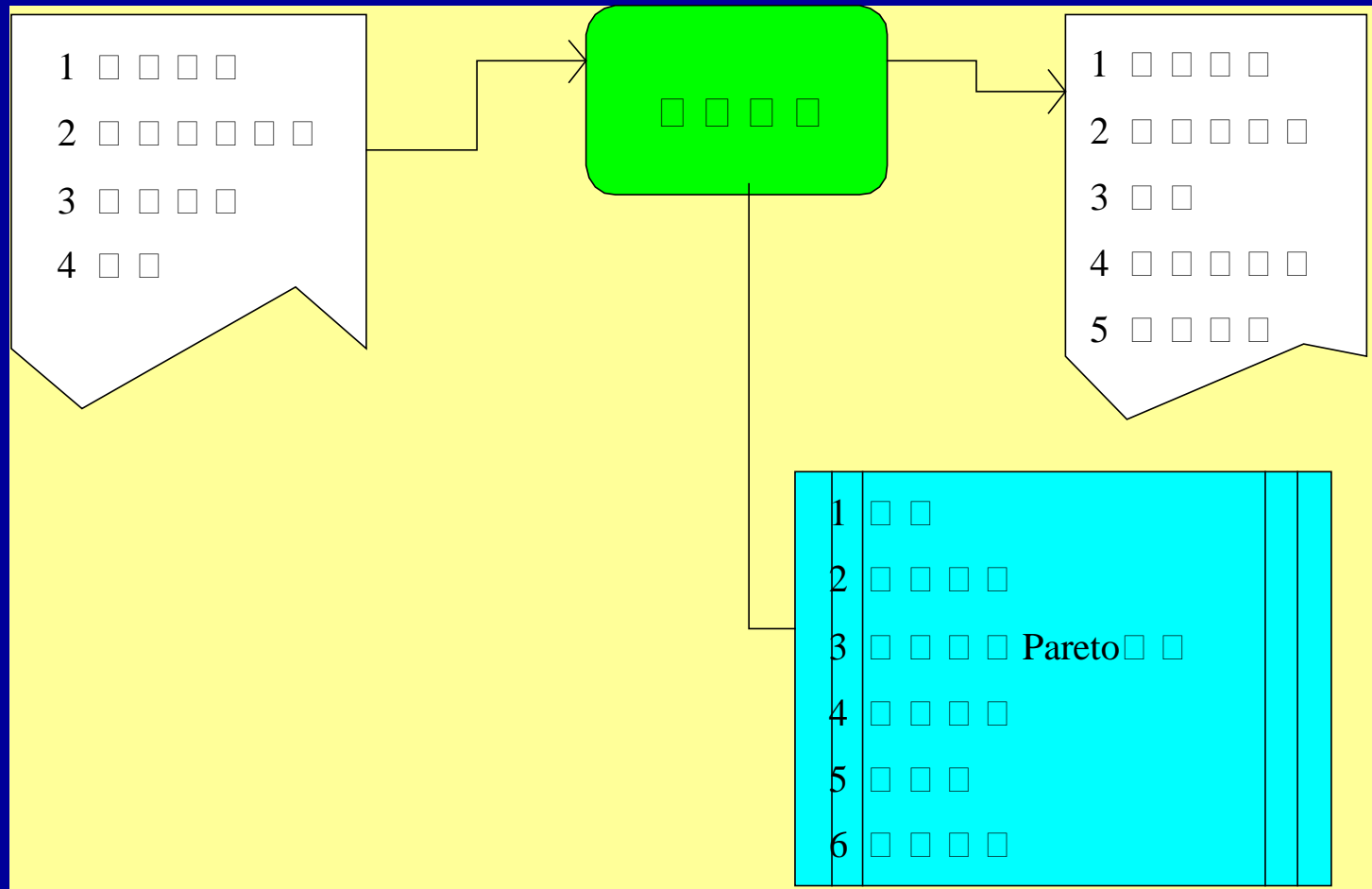
3.3 质量保证

3.4 质量控制

3.4 质量控制

质量控制包括监督特定的项目结果，以确定它们是否符合相关的质量标准和确定的方法，消除不希望有的结果的源泉。它应该在整个项目过程中持续执行。

3.4 质量控制



质量控制

3.4 质量控制

质量控制的输入元素

1. 工作结果。工作结果包括过程结果和产品结果。
2. 质量管理计划。
3. 操作定义。
4. 清单。

3.4 质量控制

质量控制的工具和技术

1. 检查。检查包括如测量、调查、和测试等用来确定结果是否符合需求的活动。
2. 控制图表。控制图表是整个时间段内过程结果的图形化显示。
3. 排列图（柏拉图，Pareto图）。排列图是一种柱形图，按照发生频率排序，显示了有多少结果是由该类型的原因产生的。
4. 统计取样。统计取样就是在调查对象中选择一部分。
5. 流程图。流程图用在质量控制中，帮助分析问题是如何发生的。
6. 趋势分析。趋势分析使用数学技术，在历史结果的基础上预测未来的结果。

3.4 质量控制

软件质量控制工具

1. 检查表
2. 排列图
3. 直方图
4. 运行图
5. 散布图
6. 控制图
7. 因果图

1. 调查表

不合格轴承分类统计表																	
产品 序号	规格	总 产量	不 合格 品数	不 合格 品率	尺寸 精度		旋转精度					噪 音	灵 活 性	残 磁	游 隙	硬 度	其 他
					外 径	内 径	内 端 面 摆	内 沟 侧 摆	内 径 向 摆	外 沟 侧 摆	内 径 向 摆						
					2	1	1	1	3	1		4		3	3		
					2	2					1	3			2		
					3	3						2		2	1		



3.4 质量控制

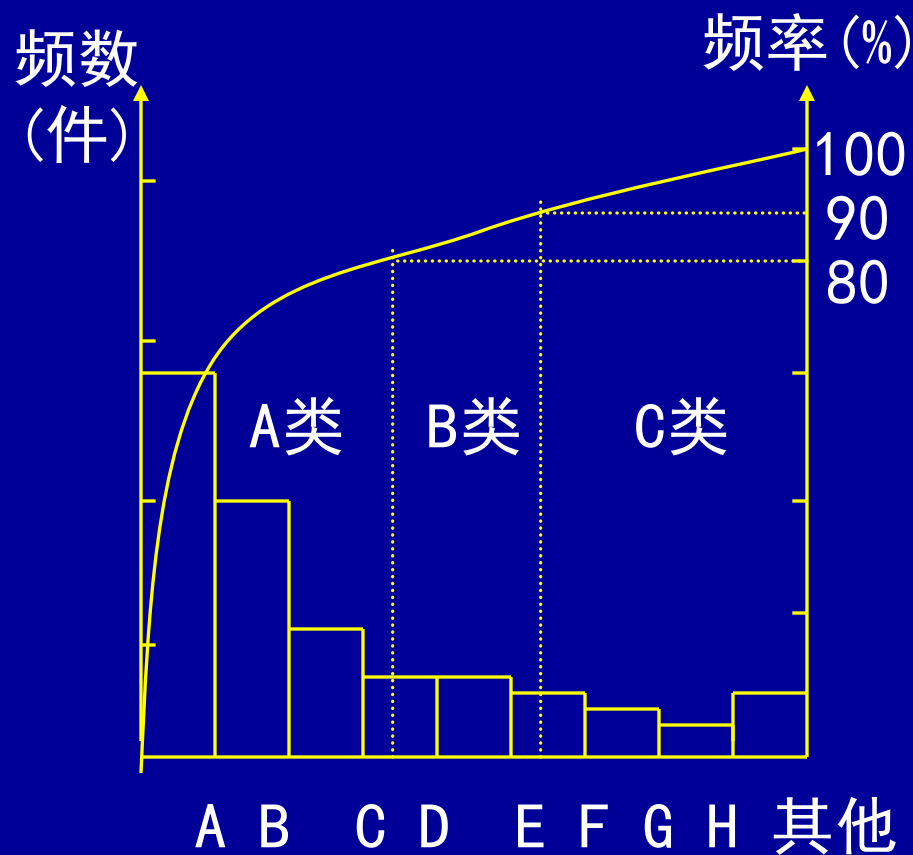
项目组	项目内容	结果↵
日志	有没有工程发布日志	有↵
	有没有任务详单	有↵
	新任务/BUG 修复	新任务↵
文档	是否用新的文档模板	是↵
	发布名称，发布日期	有
	有没有拼写错误	无↵
QA 发布报告	用的是新的报告模板	是↵
	发布名称及简介	齐全↵
	发布类型及版本号	有↵
	联系人列表	有↵
	高层已经签署	是↵

检查表

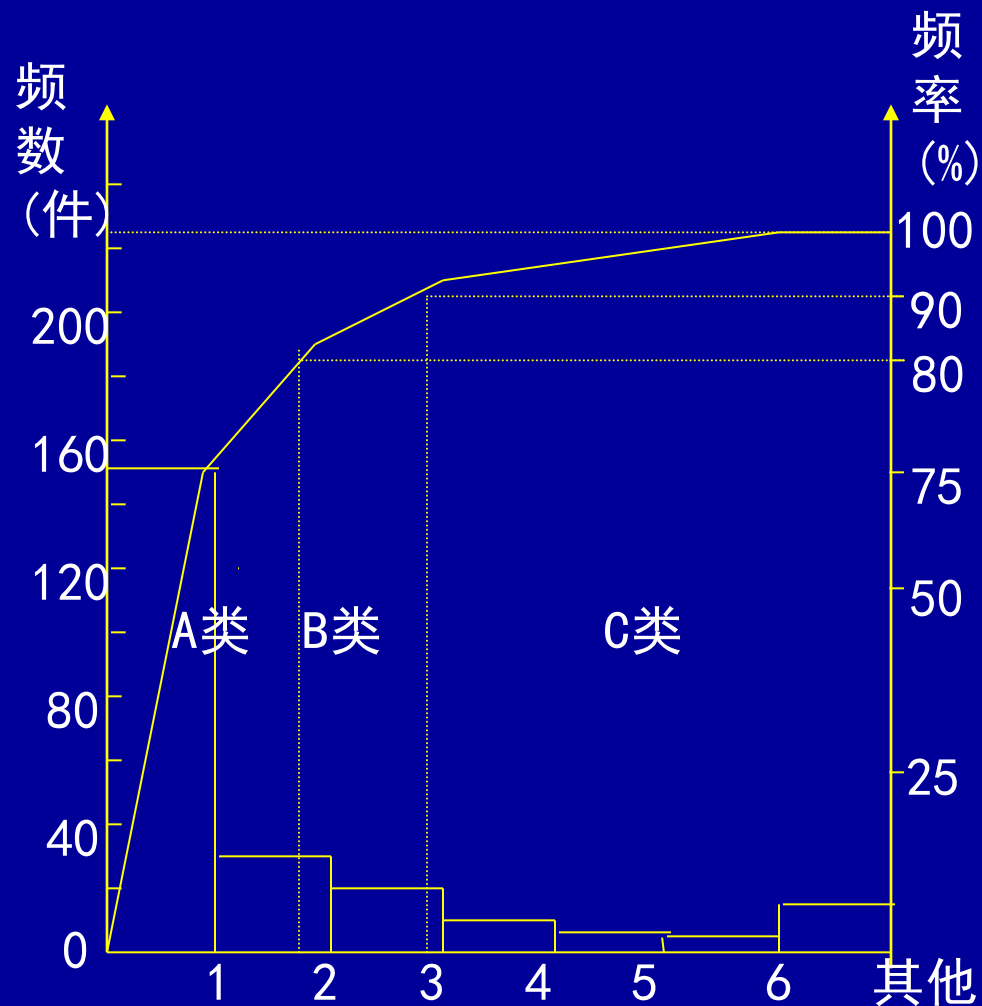
3.4 质量控制

2. 排列图

- ①排列图法是从许多影响质量的因素中分析、寻找主要因素的方法。
- ②所遵循的原理是关键的占少数,次要的占多数。

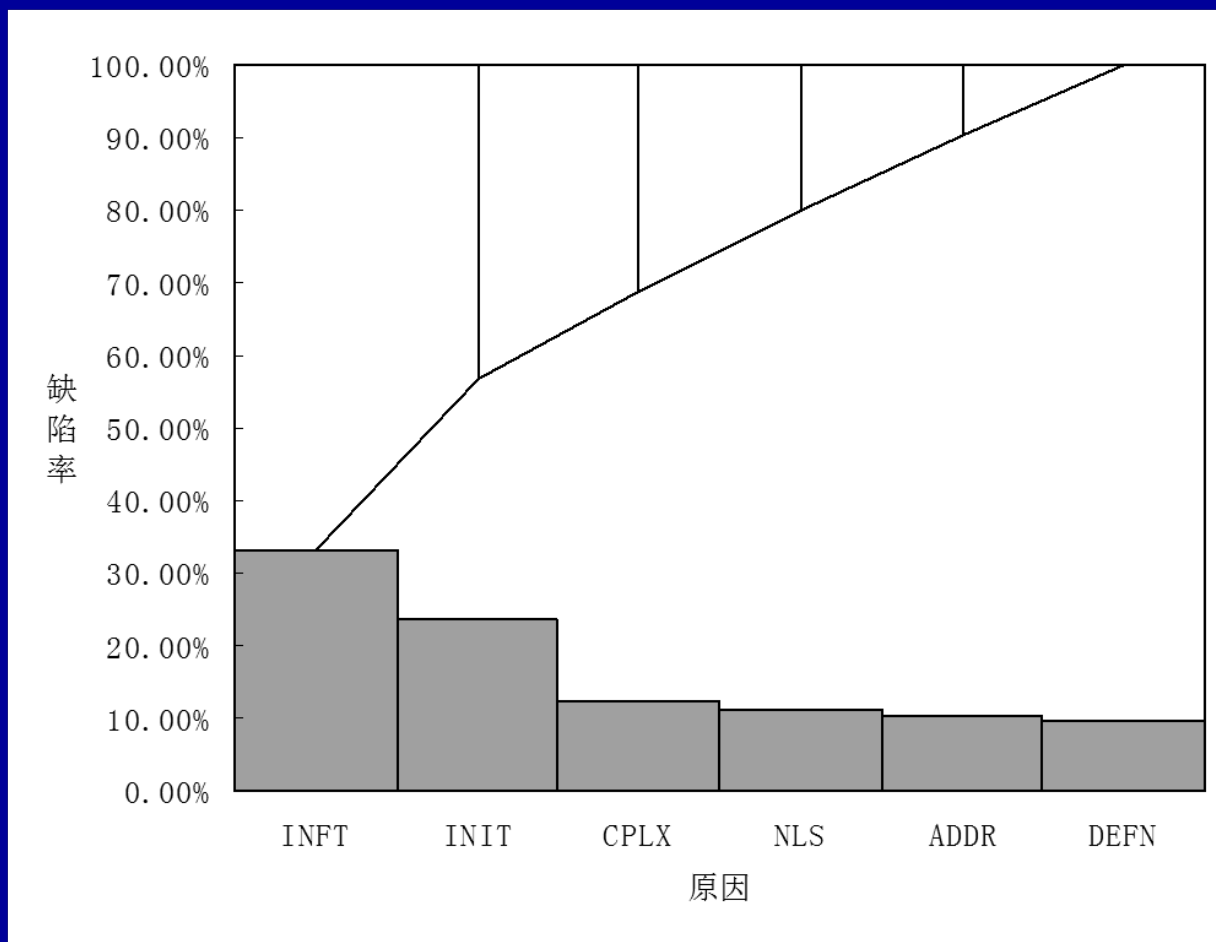


3. 绘制排列图。



4. 分析主次因素。

3.4 质量控制



排列图



3.4 质量控制

3. 直方图

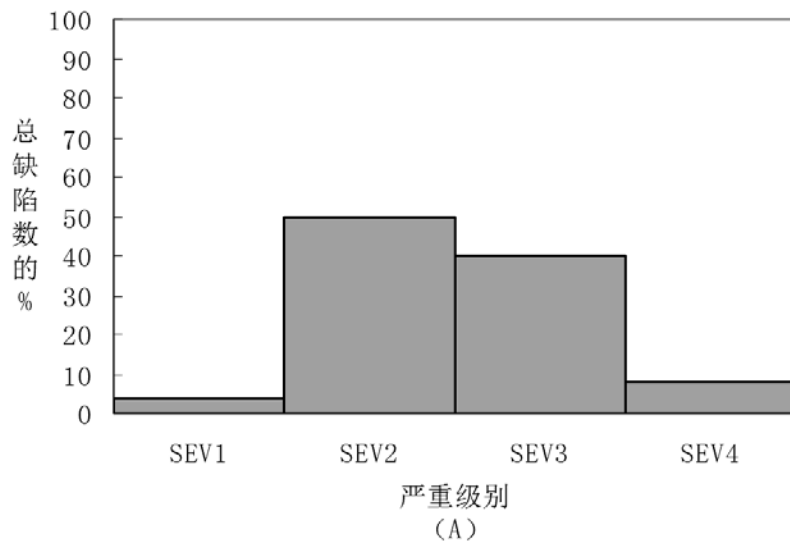
直方图的绘制

- (1) 收集数据
- (2) 找出数据中的最大值 X_{\max} 和最小值 X_{\min}
- (3) 计算极差 $R = X_{\max} - X_{\min} = 5 - (-3) = 8$
- (4) 确定组数 K ; $K=10$
- (5) 计算组距 $H = R/K = 8/10 = 0.8 \approx 1$
- (6) 确定组界; 第一组下界: $X_{\min} - h/2 = -3.5$
- (7) 计算各组中心值 X_i ; $X_1 = -3$
- (8) 计算频数 f_i , 整理频数分布表
- (9) 计算各组简化中心值 U_i
- (10) 计算均值 \bar{X}
- (11) 计算标准偏差 S

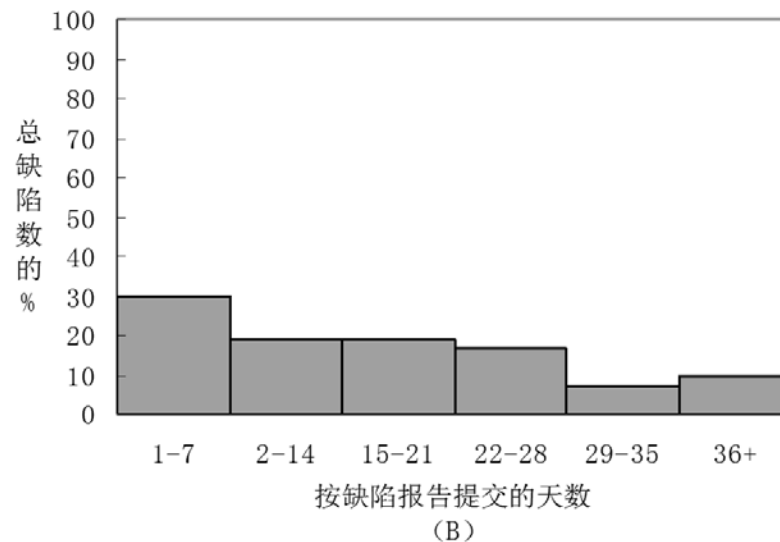
3.4 质量控制



按严重级别的缺陷分布

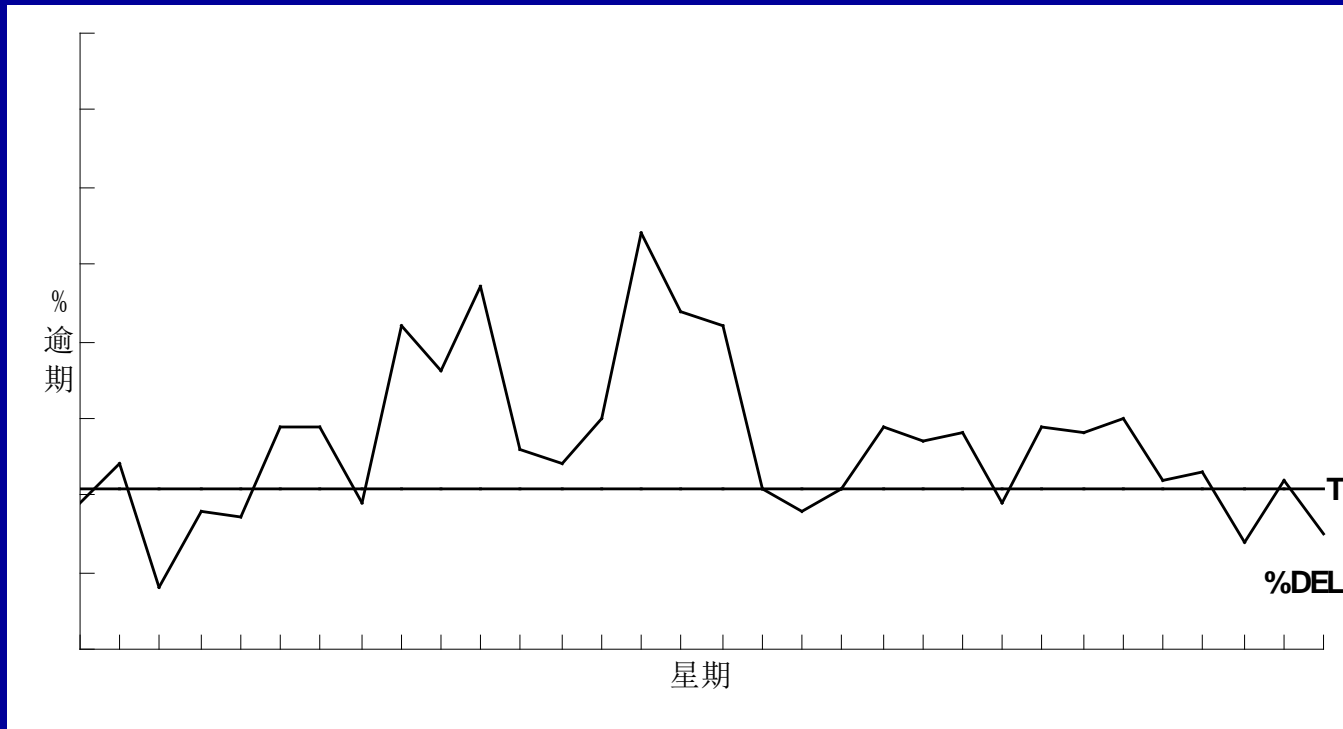


按已提交天数的缺陷分布



直方图

4. 运行图



运行图

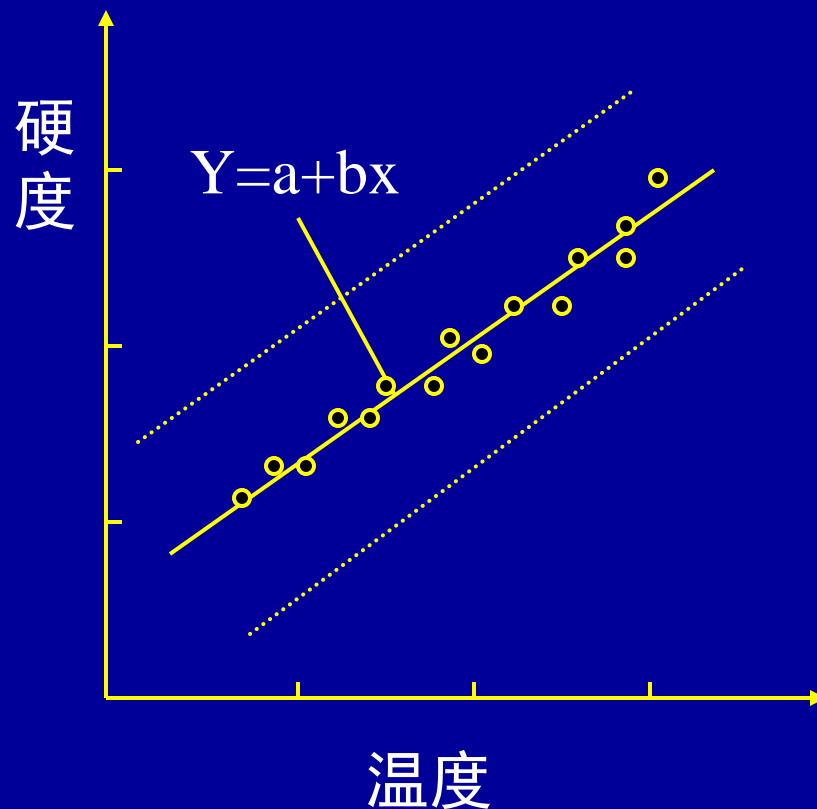
3.4 质量控制

5. 散布图

(1) 散布图是用来分析两个非确定性变量之间相关关系的方法。

(2) 散布图类型：

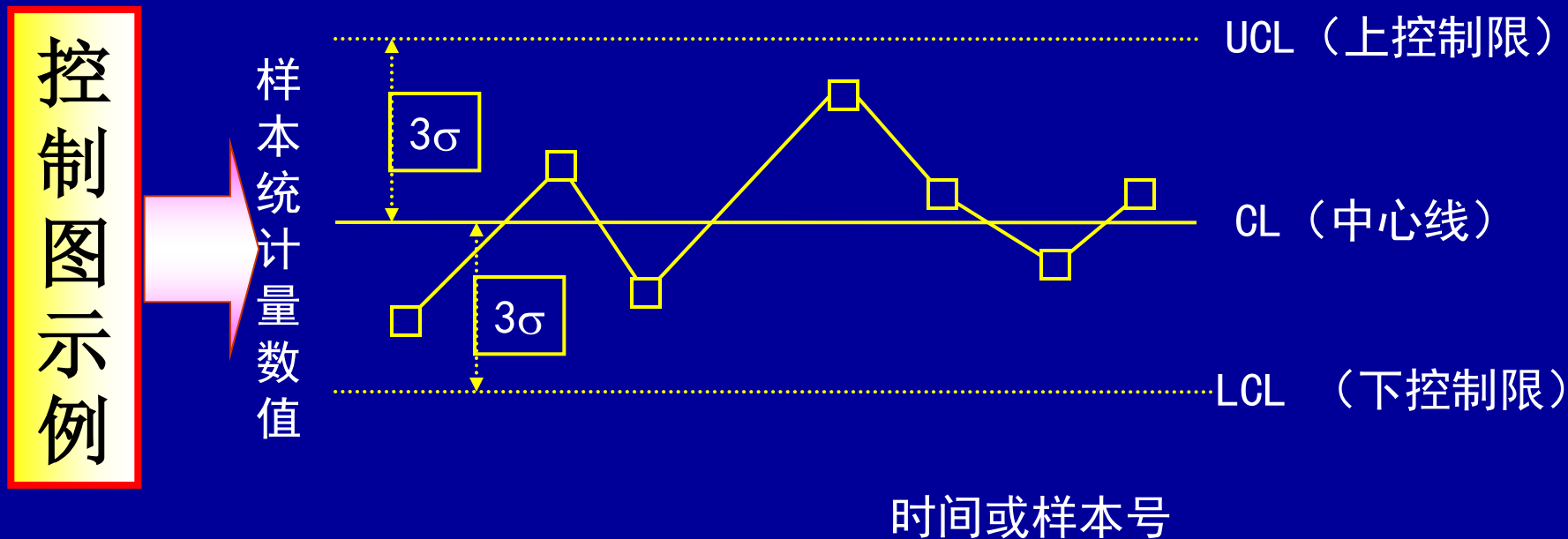
- 强相关
- 弱相关
- 不相关



3.4 质量控制

6. 控制图

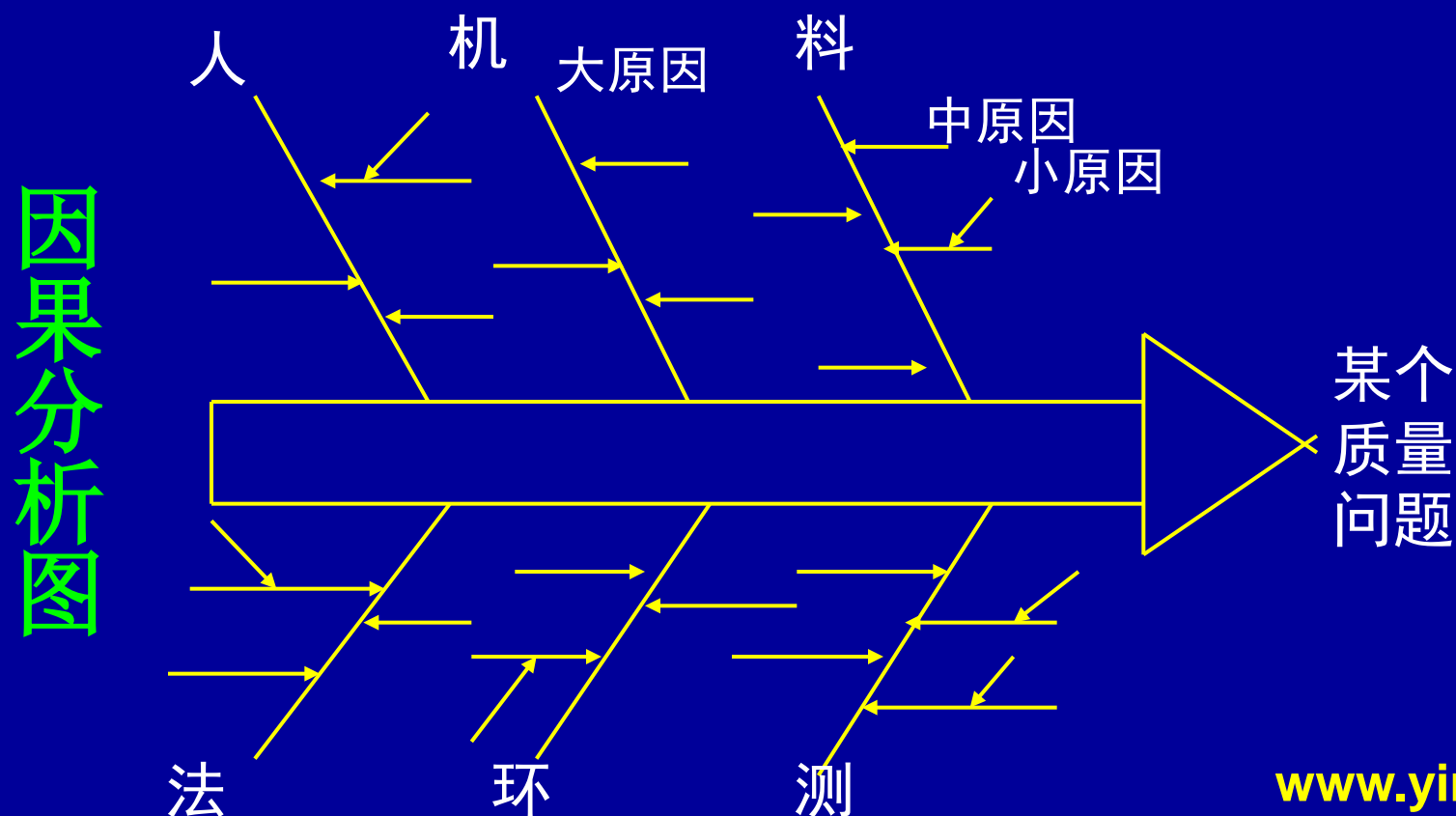
控制图是对过程质量特性值进行测量、记录、评估和监察过程是否处于统计控制状态的一种用统计方法设计的图。



3.4 质量控制

7. 因果图

- (1) 原因分析要从5M1E出发
- (2) 要集思广益
- (3) 可用排列图检查效果



质量控制的输出元素

1. 质量改进。
2. 接受的决策。
3. 返工。
4. 完成的清单。
5. 过程调整。

四、软件质量

4.1 软件质量的定义

4.2 软件质量特性

4.3 软件质量模型

4.1 软件质量的定义

- ANSI/IEEE Std 729-1983定义软件质量为“与软件产品满足规定的和隐含的需求的能力有关的特征或特性的全体”。
- M. J. Fisher 定义软件质量为“所有描述计算机软件优秀程度的特性的组合”。

4.1 软件质量的定义

软件质量的内容

产品质量

是人们实践产物的属性和行为，是可以认识，可以科学地描述的。并且可以通过一些方法和人类活动，来改进质量。

质量模型：McCall 模型, Boehm 模型, ISO 9126 模型

过程质量

软件能力成熟度模型 CMM (Capability Maturity Model).

国际标准过程模型 ISO 9000

软件过程改进和能力决断 SPICE (Software Process Improvement and Capability dEtermination)

在商业过程中有关的质量内容：

培训、成品制作、宣传、客户、风险、成本、业务等

www.yima.org.cn

四、软件质量

4.1 软件质量的定义

4.2 软件质量特性

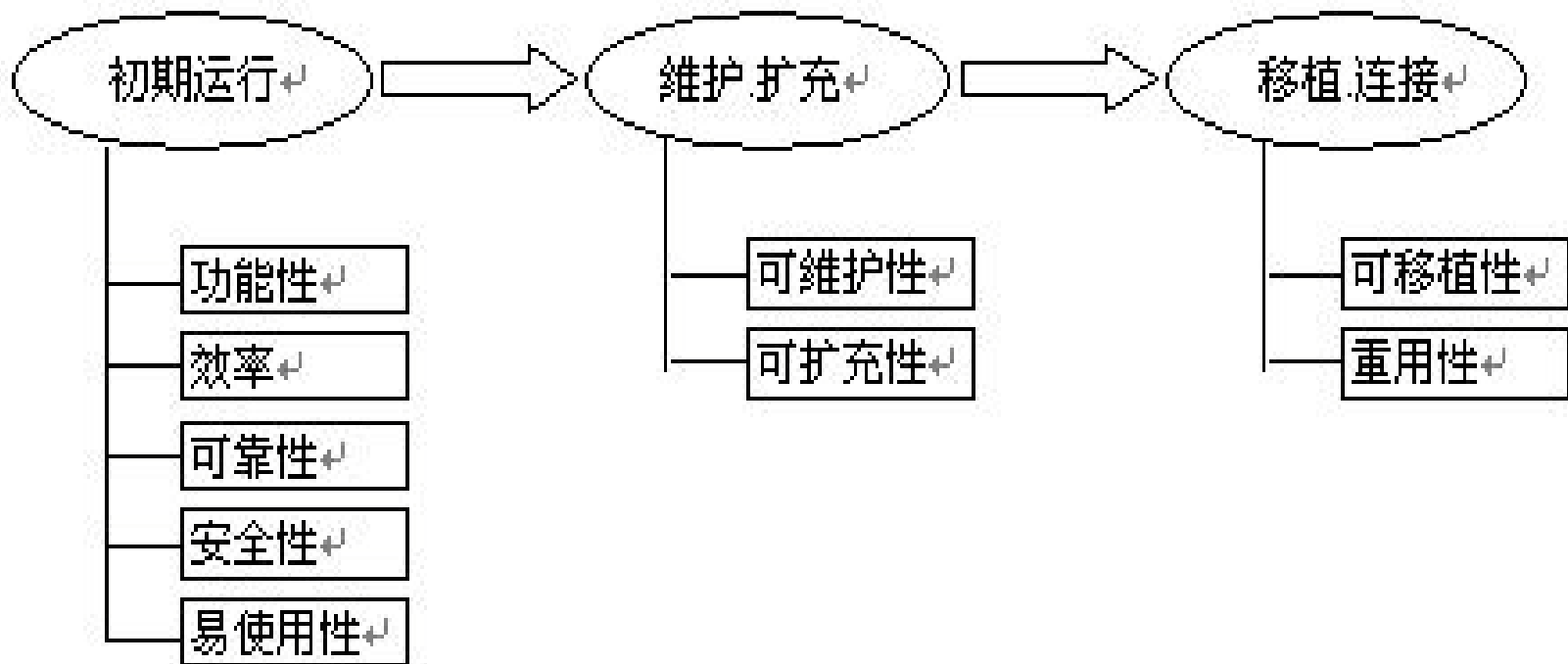
4.3 软件质量模型

4.2 软件质量特性

软件质量特性，反映了软件的本质。讨论一个软件的质量，问题最终要归结到定义软件的质量特性。而定义一个软件的质量，就等价于为该软件定义一系列质量特性。

4.2 软件质量特性

对于软件所需求的质量特性，在软件生存期的不同阶段中情况各有不同，要求也不一样，这可由下图说明。



通常，软件质量可由以下主要特性来定义：

- (1) 功能性
- (2) 效率
- (3) 可靠性
- (4) 安全性
- (5) 可维护性
- (6) 可扩充性
- (7) 可移植性
- (8) 重用性

■ 软件生存期与质量特性

从用户的角度看，软件的生存期可分为如下三个阶段：

1. 初期运用：运行新开发的软件产品。
2. 维护与扩充：在运行过程中修改缺欠的内容；而且，为了进一步的使用，需根据运行环境（主要指应用环境和技术环境）的变化做功能上和性能上的扩充。
3. 移植和连接：把在原有平台上运行的软件向其它新的运行环境转移、或者组成软件包以便重用、或与其它软件进行连接。

四、软件质量

4.1 软件质量的定义

4.2 软件质量特性

4.3 软件质量模型

4.3 软件质量模型

所谓质量模型是指提供声明质量需求和评价质量基础的特性以及特性之间关系的集合。换句话说质量模型是用来描述质量需求以及对质量进行评价的理论基础。

- 1976年 Boehm质量模型
- 1979年 McCall质量模型
- 1985年 ISO质量模型

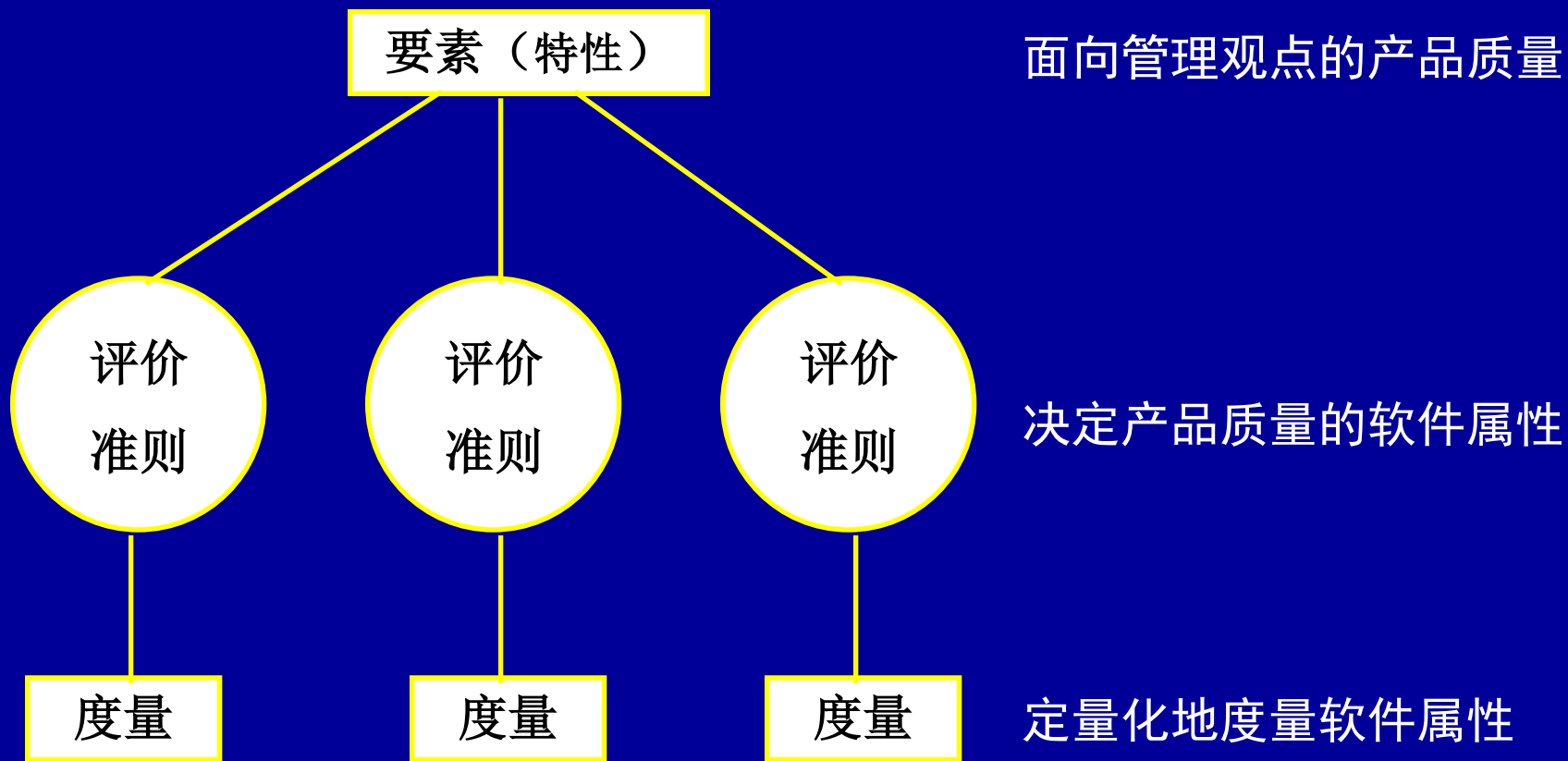


4.3 软件质量模型

Boehm软件质量模型

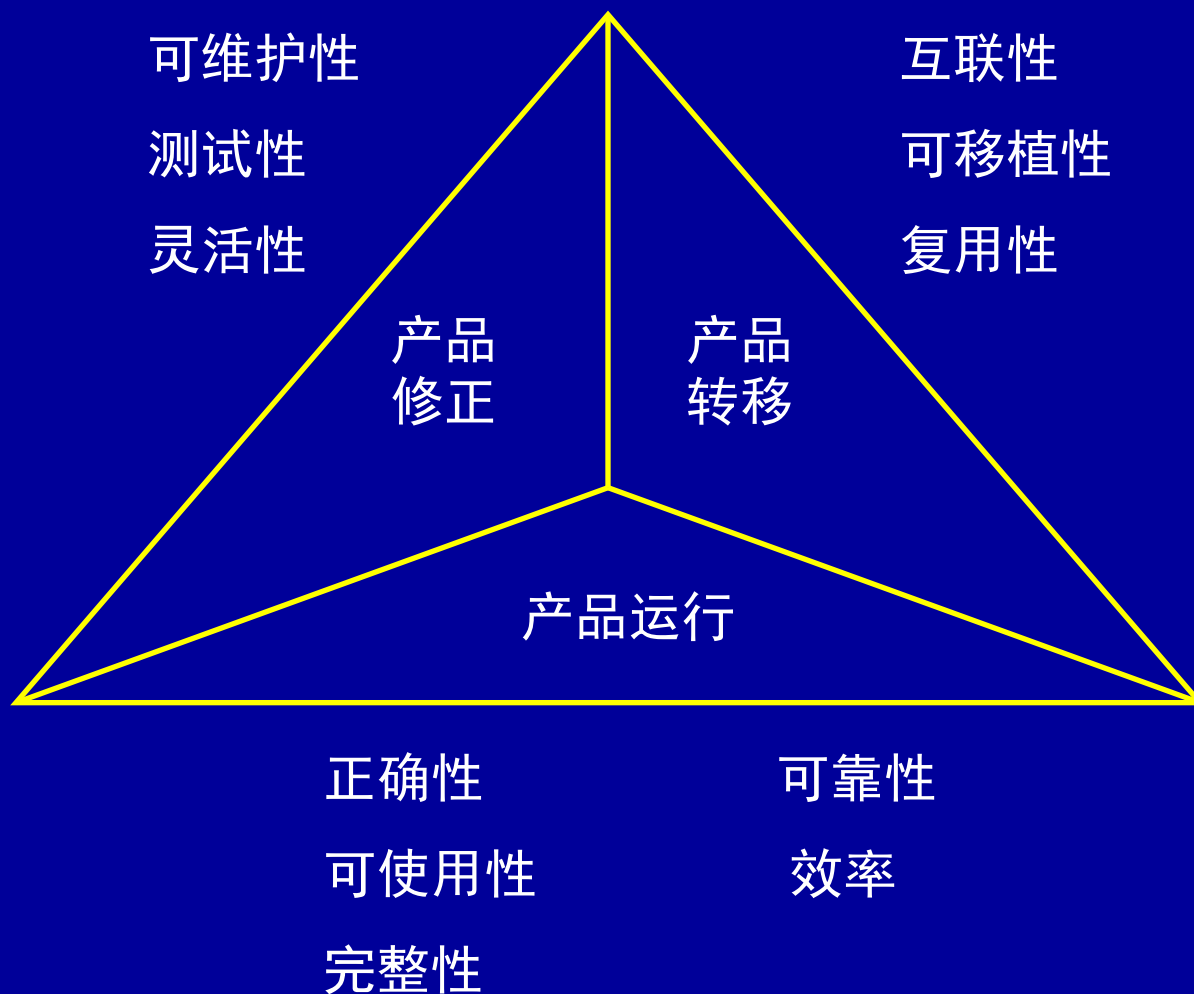


4.3 软件质量模型



McCall 质量度量模型

4.3 软件质量模型



McCall 软件质量要素评价准则

McCall 定义的软件质量要素评价准则共21种，它们是：

- | | |
|----------|-------------|
| 1. 可审查性 | 12. 硬件独立性 |
| 2. 准确性 | 13. 检测性 |
| 3. 通信通用性 | 14. 模块化 |
| 4. 完全性 | 15. 可操作性 |
| 5. 简明性 | 16. 安全性 |
| 6. 一致性 | 17. 自文档化 |
| 7. 数据通用性 | 18. 简单性 |
| 8. 容错性 | 19. 软件系统独立性 |
| 9. 执行效率 | 20. 可追踪性 |
| 10. 可扩充性 | 21. 易培训性 |
| 11. 通用性 | |

4.3 软件质量模型

计算软件质量要素

软件质量要素 F_j 的值可用下式计算

$$F_j = \sum_{k=1}^L C_{jk} M_k \quad j=1,2, \dots, 11.$$

其中， M_k 是软件质量要素， F_j 对第 k 种评价准则的测量值， C_{jk} 是相应的加权系数。□



4.3 软件质量模型

质量要素与评价准则 (1/2)

要素 关系 准则	正确性	可靠性	有效性	完整性	可维护	可测试	可移植	可重用	互操作	可用性	灵活性
可审查性				V		V					
准确性		V									
通信通用性									V		
完全性	V										
简明性			V		V						V
一致性	V	V			V						V
数据通用性									V		
容错性		V									
执行效率			V								
可扩充性											V
通用性							V	V	V		V

4.3 软件质量模型

质量要素与评价准则 (2/2)

要素 关系 准则	正确性	可靠性	有效性	完整性	可维护	可测试	可移植	可重用	互操作	可用性	灵活性
硬件独立性							V	V	V		V
检测性				V	V	V					
模块化		V			V	V	V	V	V		V
可操作性			V							V	
安全性					V						
自文档化					V	V	V	V			V
简单性		V			V	V					V
软件独立性							V	V			
可追踪性	V										
易培训性										V	

4.3 软件质量模型

ISO 软件质量度量模型

1985年国际标准化组织(ISO)建议, 软件质量度量模型由三层组成。

- 高层称软件质量需求评价准则(SQRC)
- 中层称软件质量设计评价准则(SQDC)
- 低层称软件质量度量评价准则(SQMC)

分别对应McCall等人的要素、评价准则和度量。

4.3 软件质量模型

ISO的质量要素与评价准则(1/2)

要素 关系 准则	正确性	可容性	有效性	安全性	可用性	可维护	灵活性	互操作
可追踪性	V							
完全性	V							
一致性	V	V				V		
准确性		V						
容错性		V						
简单性		V				V		
模块化						V	V	
通用性							V	
可扩充性							V	
检测性						V		
自描述性						V	V	

4.3 软件质量模型

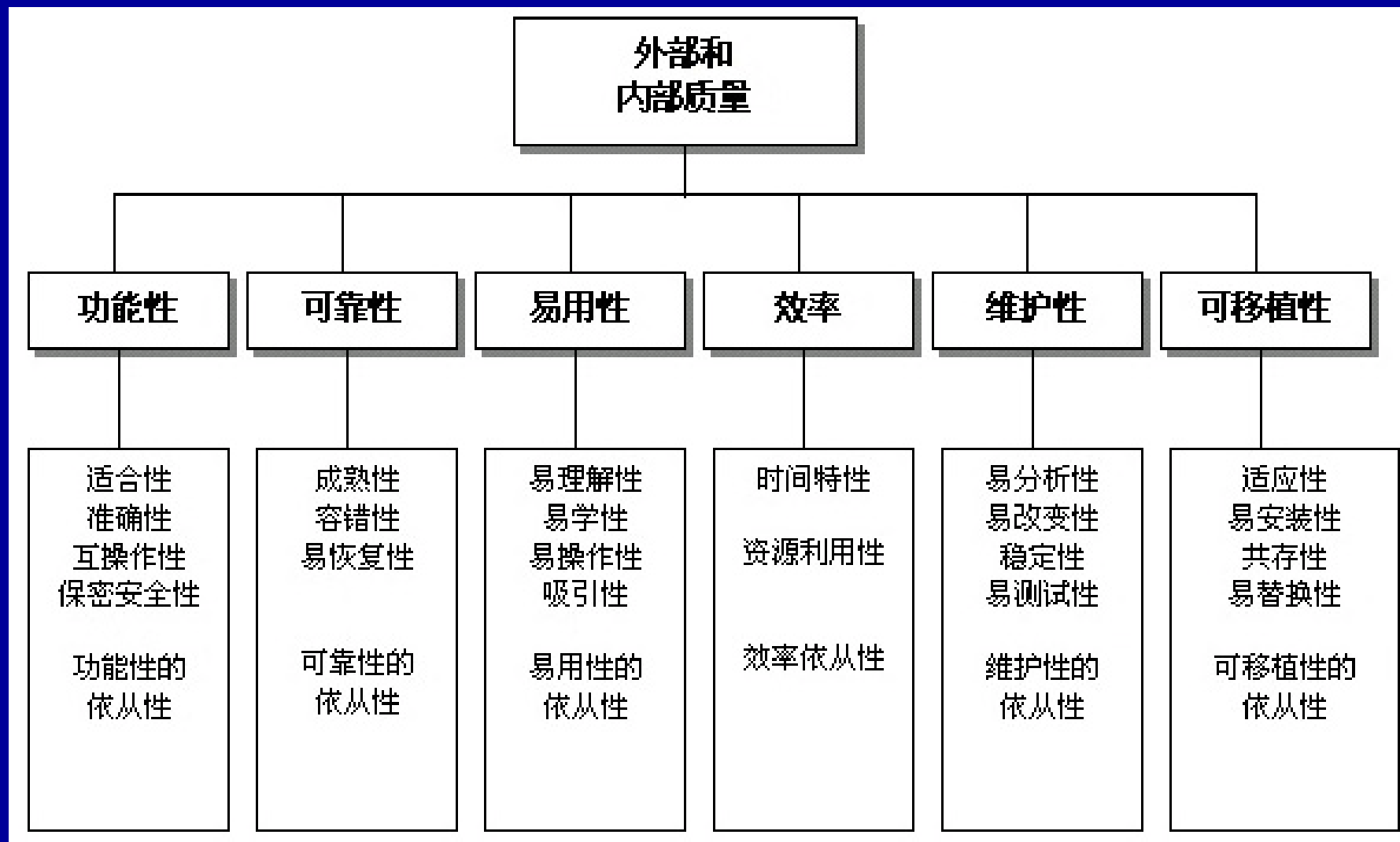
ISO的质量要素与评价准则 (2/2)

要素 关系 准则	正确 性	可容 性	有效 性	安全 性	可用 性	可维 护	灵活 性	互操 作
执行效率			V					
存储效率			V					
存取控制				V				
存取审查				V				
可操作性					V			
易培训性					V			
通信性					V			
软件独立性							V	
硬件独立性							V	
通信通用性								V
数据通用性								V
简明性						V		

1991年，ISO发布了ISO/IEC9126质量特性的国际标准，将质量特性降为6个，即功能性、可靠性、可维护性、效率、可使用性、可移植性，并定义了21个子特性。

1991年发布的ISO/IEC9126标准现在被分为了两部，ISO/IEC9126(软件产品质量)和ISO/IEC14598(软件产品评价)。

4.3 软件质量模型



软件质量特性之间的竞争

在软件的质量特性与质量特性之间、质量特性与质量量子特性之间存在着有利的影响和不利的影晌。例如，由于效率的要求，应尽可能采用汇编语言。但是用汇编语言编制出的程序，可靠性、可移植性以及可维护性都很差。

因此在系统设计过程中应根据具体情况对各种要素的要求进行折衷，以便得到在总体上用户和系统开发人员都满意的质量标准。

质量要素之间的关系表

4.3 软件质量模型

软件质量特性之间的竞争

关 系 要 素	要 素	正 确 性	可 靠 性	有 效 性	完 整 性	可 用 性	可 维 护 性	可 测 试 性	灵 活 性	可 移 植 性	可 重 用 性	可 互 操 作 性
正确性												
可靠性		△										
有效性												
完整性				▼								
可用性		△	△	▼	△							
可维护性		△	△	▼		△						
可测试性		△	△	▼		△	△					
灵活性		△	△	▼	▼	△	△	△				
可移植性				▼			△	△				
可重用性			▼	▼	▼		△	△	△	△		
可互操作性				▼	▼					△	△	△

五、软件质量保证

5.1 什么是软件质量保证

5.2 软件质量保证的主要任务

5.3 软件质量保证体系

5.4 SQA活动通用框架

5.5 SQA的组织活动

五、软件质量保证

5.1 什么是软件质量保证

软件质量保证就是向用户及社会提供满意的高质量的产品。

软件的质量保证活动也和一般的质量保证活动一样，是确保软件产品从诞生到消亡为止的所有阶段的质量的活动。即为了确定、达到和维护需要的软件质量而进行的所有有计划、有系统的管理活动。

质量保证与质量检验

- 其一是切实搞好开发阶段的管理，检查各开发阶段的质量保证活动开展得如何；
- 其二是预先防止软件差错给用户造成损失。
- 为了确保每个开发过程的质量，防止把软件差错传递到下一个过程，必须进行质量检验。

五、软件质量保证

5.1 什么是软件质量保证

5.2 软件质量保证的主要任务

5.3 软件质量保证体系

5.4 SQA活动通用框架

5.5 SQA的组织活动

五、软件质量保证

5.2 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
2. 力争不重复劳动
3. 掌握开发新软件的方法
4. 组织外部力量协作的方法
5. 排除无效劳动
6. 发挥每个开发者的能力
7. 提高软件开发的工程能力
8. 提高计划和管理质量能力

五、软件质量保证

5.1 什么是软件质量保证

5.2 软件质量保证的主要任务

5.3 软件质量保证体系

5.4 SQA活动通用框架

5.5 SQA的组织活动

五、软件质量保证

5.3 软件质量保证体系

软件的质量保证活动，是涉及各个部门的部门间的活动。

为了顺利开展这些活动，事先明确部门间的质量保证业务，确立部门间的联合与协作的机构十分重要，这个机构就是质量保证体系。

软件质量保证规程和技术准则

- 规定在项目的哪个阶段进行评审及如何评审；
- 规定在项目的哪个阶段应当产生哪些报告和计划；
- 规定产品各方面测试应达到的水平。
- 在每次评审和测试中发现的错误如何修正；
- 描述希望得到的质量度量；
- 说明各种软件人员的职责，规定为了达到质量目标他们必须进行哪些活动。
- 建立
 - 在各阶段中执行质量评价的质量评价和质量检查系统
 - 有效运用质量信息的质量信息系统，并使其运行。

质量保证的实施

- 软件质量保证的实施需要从纵向和横向两个方面展开。
 - 要求所有与软件生存期有关的人员都要参加
 - 要求对产品形成的全过程进行质量管理
- 这要求整个软件部门齐心协力，不断完善软件的开发环境。此外还需要与用户共同合作。

质量目标与度量

为了开发高质量的软件，需要明确软件的功能，明确软件应达到什么样的质量标准，即质量目标。

为了达到这个目标，在开发过程中的各个阶段进行检查和评价。

在做质量评价时，需要有对质量进行度量的准则和方法。

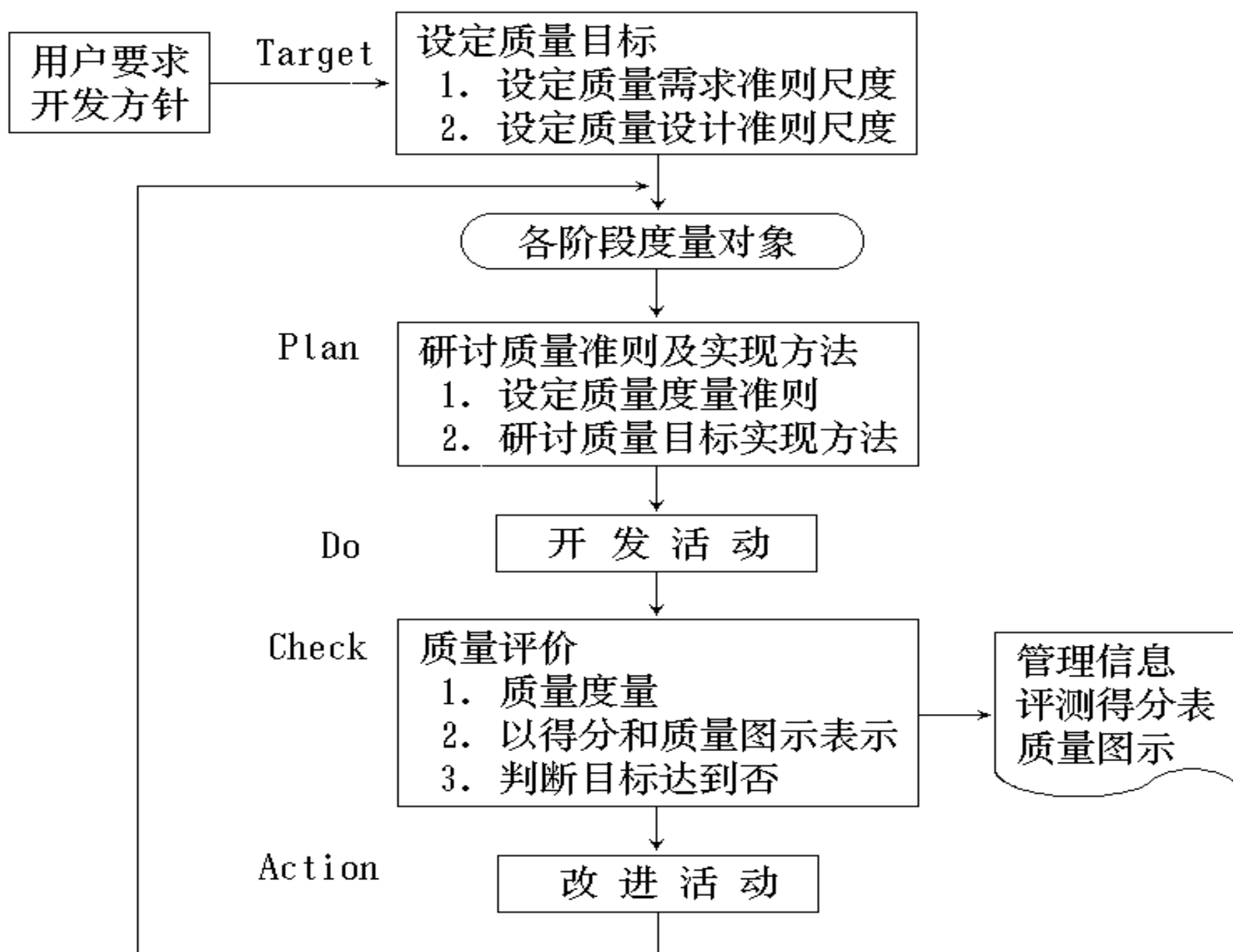
需要有在软件生存期中如何使用这些准则和方法的质量保证步骤，以及提高该项作业效率的工具。

软件质量度量和保证的条件

- 适应性：适应各种用户、软件类型
- 易学性：不需要特殊技术，易掌握
- 可靠性：同个软件的评价结果一致
- 针对性：设计阶段就确立质量目标，在各个阶段实施落实。
- 客观性
- 经济性

质量保证活动的实施步骤

- Target: 以用户要求和开发方针为依据, 对质量需求准则、质量设计准则的各质量特性设定质量目标。
- Plan: 设定适合于被开发软件的评测检查项目(质量评价准则)。研讨实现质量目标的方法或手段。
- Do: 制作高质量的规格说明和程序。在接受质量检查前先做自我检查。
- Check: 以Plan阶段设定的质量评价准则进行评价。计算结果用质量图的形式表示出来。比较评价结果的质量得分和质量目标, 看其是否合格。
- Action: 对评价发现的问题进行改进活动, 如果实现并达到了质量目标就转入下一个工程阶段。这样重复“Plan”到“Action”的过程, 直到整个开发项目完成。



质量设计准则评价得分

系统 ID [SI]

度量者 [宋明华]

工程名 [设计工程]

度量日 [97/7/1]

度量单位 [PROL]

设计质量 [0.769]

质量需求准则	质量设计准则	得分	图 示
正确性	可追踪性	0.916	<div></div>
	完备性	0.916	<div></div>
	一致性	0.872	<div></div>
可靠性	一致性	0.872	<div></div>
	简单性	0.812	<div></div>
	容错性	0.500	<div></div>
	准确性	0.616	<div></div>
可维护性	一致性	0.872	<div></div>
	模块性	0.806	<div></div>

质量设计准则评价得分

系统 ID [SI]

度量者 [宋明华]

工程名 [设计工程]

度量日 [97/7/1]

度量单位 [PROL]

设计质量 [0.769]

正确性 90%

互连性
不适用

可靠性 71%

保密安全性
30%

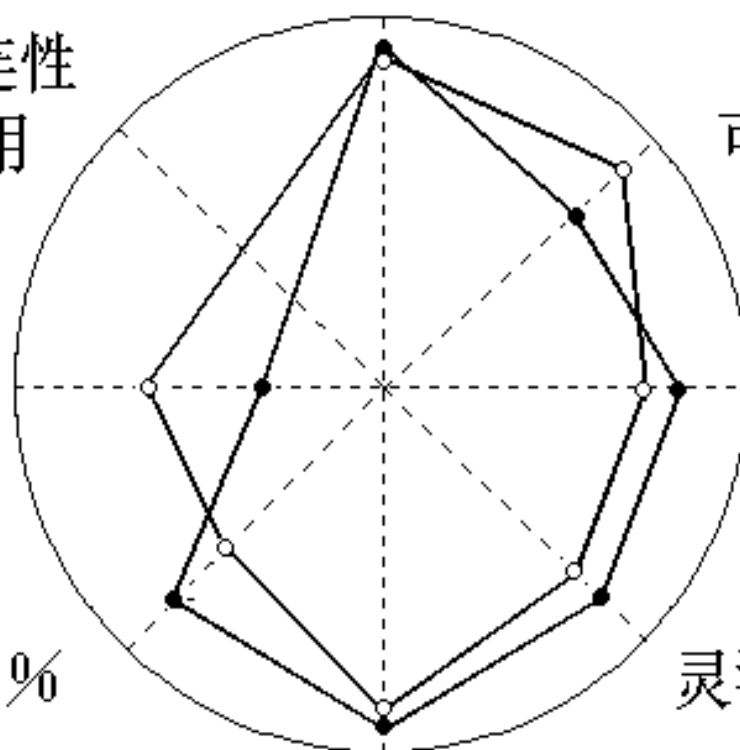
可维护性
85%

效率 85%

灵活性 85%

可使用性 93%

○—○ 目标
●—● 实际



软件的质量设计

- 质量特性转换为软件的内部结构。
- 在软件定义阶段，必须定义对软件的质量需求。即确定软件的质量特性及必需的评价准则，并定量地设定其必须达到的质量水平。
- 在以后软件开发的每一阶段结束时，要算出评价的分数，然后与目标值加以对照，以评估在这一阶段开发的软件质量是否达到要求。
- 为了实现规定的质量特性，就需要把这些质量特性转换为软件的内部结构的特性。

软件的结构特性与评价标准

- 结构特性：逻辑数据层次
- 评价标准：
 - 全部数据元素定义完毕
 - 所有层次的操作符定义完毕

- 结构特性：功能层次
- 评价标准：
 - 全部功能元素定义完毕
 - 所有层次的操作符定义完毕

- 结构特性：逻辑数据与功能的对应关系
- 评价标准：
 - 所有数据都与功能对应
 - 所有功能元素都与数据对应
 - 逻辑数据与功能的相互关系个数（局部）
- 结构特性：物理数据层次
- 评价标准：
 - 全部数据元素定义完毕
 - 物理数据之间的所有指针定义完毕
 - 上述指针都具有层次性

- 结构特性：模块层次
 - 评价标准：
 - 所有模块定义完毕
 - 模块之间所有控制关系定义完毕
 - 上述关系都是标准过程调用形式
 - 各层次上的模块大小适当
 - 结构特性：物理数据与模块的对应关系
 - 评价标准：
 - 所有物理数据都与模块对应
 - 所有模块都与物理数据对应
 - 对应于一个物理数据的模块数（以一对一为好）
- www.yima.org.cn

- 结构特性：逻辑数据与物理数据的对应关系
- 评价标准：
 - 所有逻辑数据都与物理数据对应
 - 对应于一个物理数据的逻辑数据数（以一对一为好）
- 结构特性：功能与模块的对应关系
- 评价标准：
 - 所有功能都与模块对应
 - 对应模块的功能个数（以一对一为好）

软件可靠性

- 软件生存期与软件寿命的关系
- 在软件工程中常用的定义
- 软件可靠性定义
- 测试中的可靠性分析
- 测试精确度和测试覆盖度的评价

软件故障产生原因

- 支持软件工作的基本条件(除硬件外的操作系统、数据库管理系统、编译程序、微代码等)的缺陷
- 软件设计不当
- 加入了允许范围之外的输入

软件可靠性的定义

软件可靠性是软件在给定的时间间隔及给定的环境条件下，按设计要求，成功地运行程序的概率。

环境条件一指指的是软件的使用环境。无论是什么软件，如果不对它的使用环境加以限制，都是会失效的。这种失效的数据，不能用来度量软件的可靠性。

规定的时间—在定义中，一般采用“运行时间” t 作为时间的尺度。因

- 具体要处理的问题是多种多样的
- 其对应的输入环境是随机
- 程序中相应程序路径的选取也是随机的
- 软件的失效也是随机的

应当把运行时间 t 当作随机变量来考虑。

规定的功能—在考虑软件可靠性时，首先应当明确软件的功能是什么，哪些功能是主要的，哪些功能是次要的。一般从软件需求分析说明书和设计说明书中可以了解这些情况。

由于功能不同，失效带来的损失就不一样。因此，还要明确哪些失效是致命的，哪些失效是非致命的，哪些又是容易修复的。此外，还要明确，怎样才算是完成了一个规定的功能。

成功地运行程序一是指不仅程序能正确地运行，满足用户对它的功能要求，而且当程序一旦受到意外的伤害，或系统故障时，能尽快恢复，仍能正常地运行。

1. SQA审计与评审
2. SQA报告
3. 处理不符合问题
4. 实施

五、软件质量保证

- 5.1 什么是软件质量保证
- 5.2 软件质量保证的主要任务
- 5.3 软件质量保证体系
- 5.4 SQA活动通用框架
- 5.5 SQA的组织活动

五、软件质量保证

5.4 SQA活动通用框架

1. 提出软件质量需求
2. 确定开发方案
3. 阶段评审
4. 测试管理
5. 文档化管理
6. 验证产品与相应文档和标准的一致性
7. 建立测量机制
8. 记录并生成报告

五、软件质量保证

- 5.1 什么是软件质量保证
- 5.2 软件质量保证的主要任务
- 5.3 软件质量保证体系
- 5.4 SQA活动通用框架
- 5.5 SQA的组织活动

五、软件质量保证

5.5 SQA的组织活动

5.5.1 软件质量管理的组织介绍

5.5.2 SQA组织结构和角色

5.5.3 SQA组织的目标和责任

5.5.4 SQA人员的要求和培养

5.5.5 员工的培训与认证

5.5 SQA的组织活动

5.5.1 软件质量管理的组织介绍

1. 基本的软件质量组织

- 软件测试部门
- 软件质量保证组织

2. 其他的软件质量组织

- SEPG (Software Engineering Process Group)
- SPIN (Software Process Improvement Network)
- QAI (Quality Assurance Institute)

5.5 SQA的组织活动

5.5.2 SQA组织结构和角色

1. SQA组织的建立
2. 常用的组织结构模型
3. 角色的分类和职能
4. 各角色之间的关系

2. 常用的组织结构模型

(1) 独立的SQA部门



优点：

- 保护SQA工程师的独立性和客观性
- 有利于资源的共享

缺点：

- 难于深入项目并发现关键问题
- SQA工程师发现的问题不能及时解决

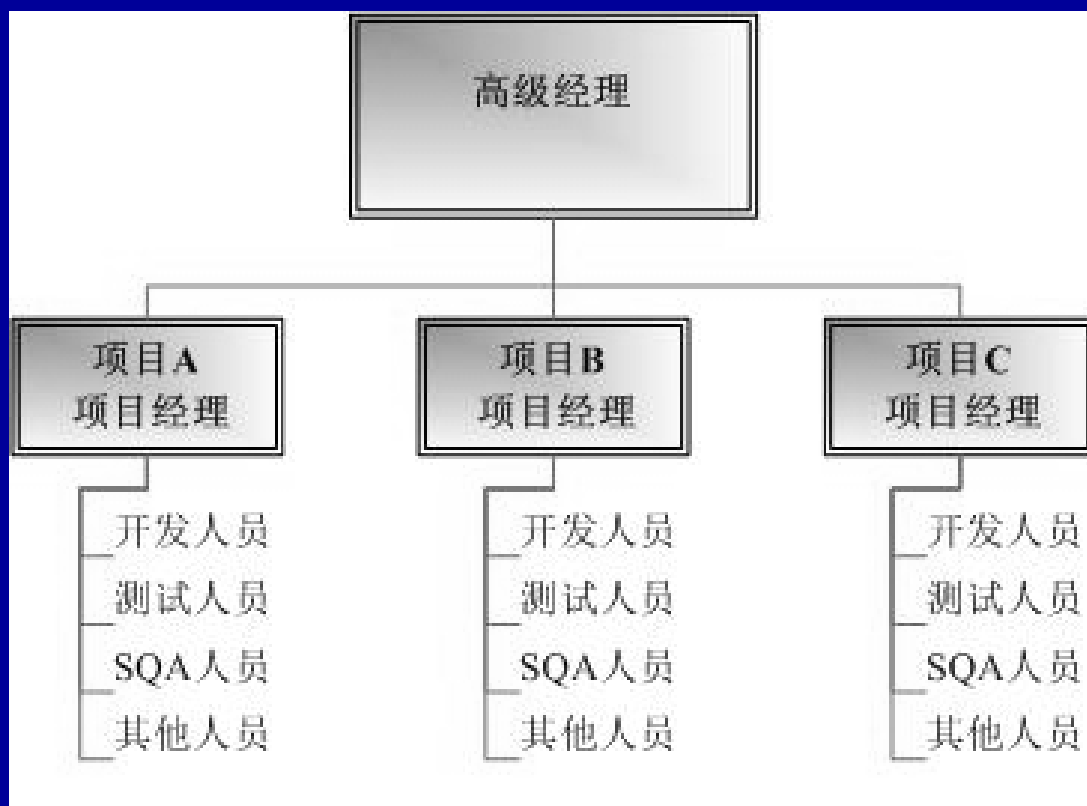
(2) 独立的SQA工程师（非独立SQA小组）

优点：

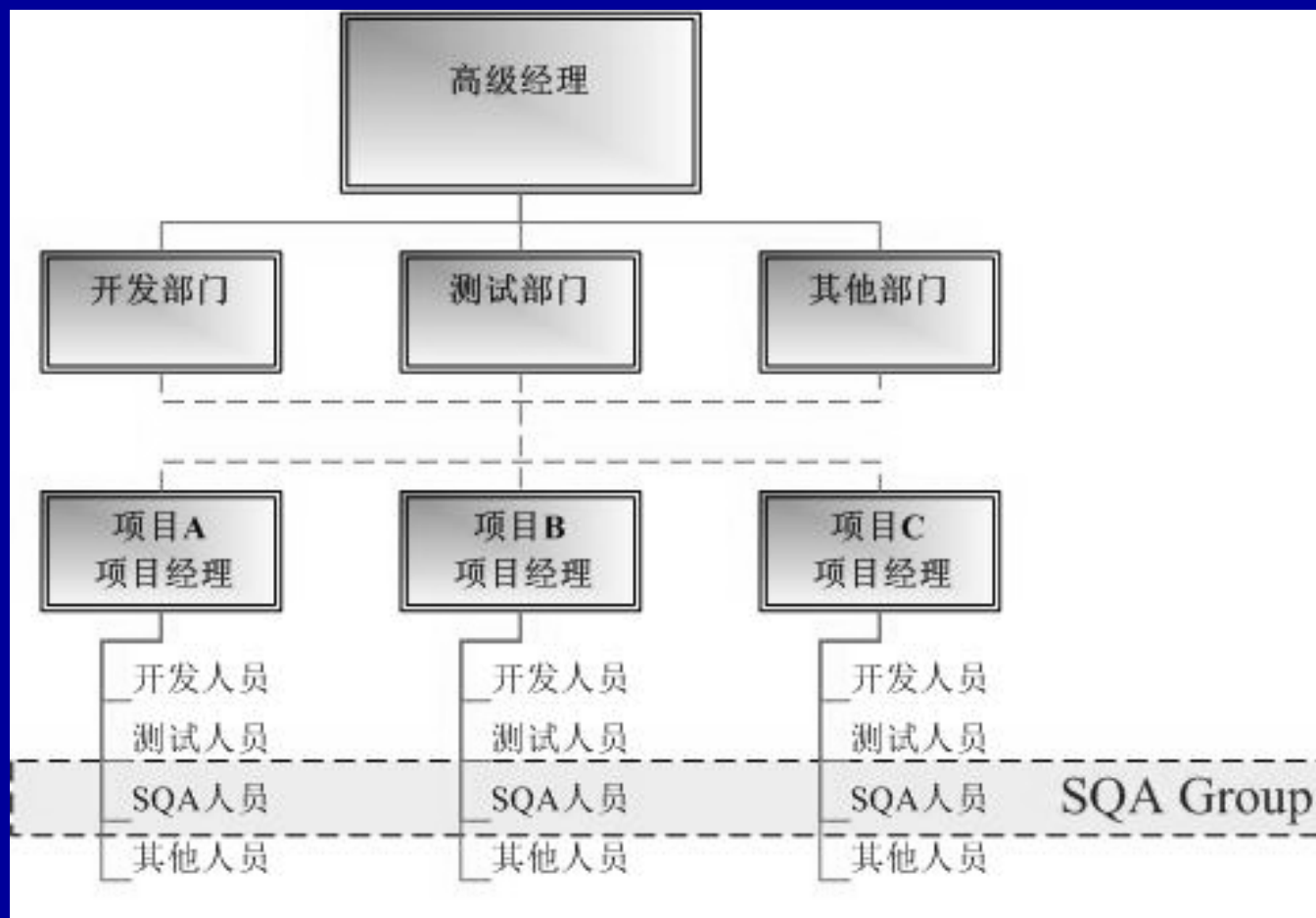
- 能够深入项目发现实质性问题
- SQA工程师发现的问题能够及时解决

缺点：

- SQA工程师之间的沟通和交流
- 独立性和客观性不足



(3) 独立的SQA工程师（独立的SQA小组）



3. 角色的分类和职能

(1) 非全职的QA

非全职SQA是指在组织结构中有自己的本职工作，在完成本职工作之外，还需要兼职完成SQA的任务的相关人员。

- 项目经理
- 开发工程师
- 测试工程师

(2) 全职的QA

专职的SQA人员承担了大部分的SQA任务，对质量保证目标的实现起着非常重要的作用。

- SQA经理
- SQA工程师

SQA是整个企业，整个组织的责任，而不仅仅是某个部门或某几个人的责任。

4. 各角色之间的关系

1. SQA和项目经理

SQA和项目经理之间是合作的关系，帮助项目经理了解项目中过程的执行情况、过程的质量、产品的质量、产品的完成情况等。

2. SQA和开发工程师

SQA和开发人员应该保持良好的沟通和合作，任何对立和挑衅都可能导致质量保证这个大目标失败。

3. SQA和测试工程师

SQA和测试人员都充当着第三方检查人员的角色。但是SQA主要对流程进行监督和控制，而测试人员则是针对产品本身进行测试。

5.5 SQA的组织活动

5.5.3 SQA组织的目标 and 责任

1. SQA计划
2. 评审和审核
3. SQA报告
4. SQA度量
5. SQA评估任务

1. SQA计划

SQA计划的实施步骤：

- (1) 了解项目的需求，明确项目SQA计划的要求和范围
- (2) 选择SQA任务
- (3) 估计SQA的工作量和资源
- (4) 安排SQA任务和日程
- (5) 形成SQA计划
- (6) 协商、评审SQA计划
- (7) 批准SQA计划
- (8) 执行SQA计划

5.5 SQA的组织活动

SQA计划包含的内容：

- 目的—SQA计划的目的是范围
- 参考文件—该SQA计划参考的文件列表
- 管理—组织，任务，责任
- 文档—列出所有相关的文档，如程序员手册，测试计划，配置管理计划，……
- 标准定义—文档标准，逻辑结构标准，代码编写标准，注释标准，……
- 评审/审核
- 配置管理—配置定义，配置控制，配置评审，……
- 问题报告和处理
- 工具，技术，方法
- 代码控制
- 事故/灾难控制—包括火灾，水灾，病毒，……

2. 评审和审核

评审 (Review) —— 对执行过程的人在过程进行时, 对过程的检查; SQA的角色在于确保当执行工程活动时各项计划所规定的过程得到遵循。评审通常通过评审会的方式进行。

审核 (Audit) —— 在软件工作产品生成时, 对其进行的检查; SQA的角色在于确保开发工作产品中各项计划所规定的过程得到遵循; 审核通常通过对工作产品的审查来执行。

主要的评审和审核：

- 软件需求评审 software requirements review
- 概要设计评审 preliminary design review
- 详细设计评审 detailed design review
- 软件验证与确认评审 software verification and validation review
- 功能审核 functional audit
- 物理审核 physical audit
- 综合检查 comprehensive audit
- 管理评审 management reviews

3. SQA报告

SQA活动的一个重要内容就是报告对软件产品或软件过程评估的结果，并提出改进建议。

在完成SQA报告的过程中要注意下面两个问题：

1. SQA报告失去原有的价值
2. 明确报告原则

4. SQA度量

(1) 软件产品评估度量

软件产品评估	页数	评估耗时	报告耗时
软件需求说明	20页	3小时	1小时

(2) 软件产品质量度量

(3) 软件过程审核度量

被审核的软件过程	审核准备耗时	评估耗时	报告耗时
错误纠正过程	2小时	2小时	1小时

5. SQA评估任务

(1) 软件工具评估

SQA需要对软件开发和支持正在使用以及计划使用的软件工具进行评估，其目的主要是保证项目组能够采用合适的技术和工具。

(2) 项目设施评估

项目设施评估的内容非常单一，仅仅是检查是否为软件开发和支持提供了所需要的设备和空间。

5.5 SQA的组织活动

5.5.4 SQA人员的要求和培养

SQA人员的要求：

- 扎实的技术基础和背景
- 良好的沟通能力
- 敏锐性和客观性
- 积极的工作态度
- 独立工作的能力

5.5 SQA的组织活动

5.5.4 SQA人员的要求和培养

目前，优秀的SQA人员非常缺乏，因此，SQA人员的培养变得非常重要。

根据SQA人员的要求，SQA人员的培养主要集中于两个方面：

- 技术培养
- 素质培养

5.5.5 员工的培训与认证

1. 培训的目标和过程
2. 认证
3. 六西格玛角色和职责
4. 六西格玛培训

1. 培训的目标和过程

SQA培训的目标：

- 使员工掌握更多的**技术和知识**，在工作中能够更有效地完成分配的任务。
- 传授**规则、流程**等方面的知识和概念，以确保开发的产品符合企业的标准。
- 了解和熟悉**SQA流程**。
- 确保每个职位上的员工都是**合格的**。

2. 认证

CSQA (Certified Software Quality Analyst)

软件质量分析师是QAI (Quality Assurance Institute) 对质量人员的认证, 获得该认证表明了在保证理论和实践方面的专业能力。

CSQE (Certified Software Quality Engineer)

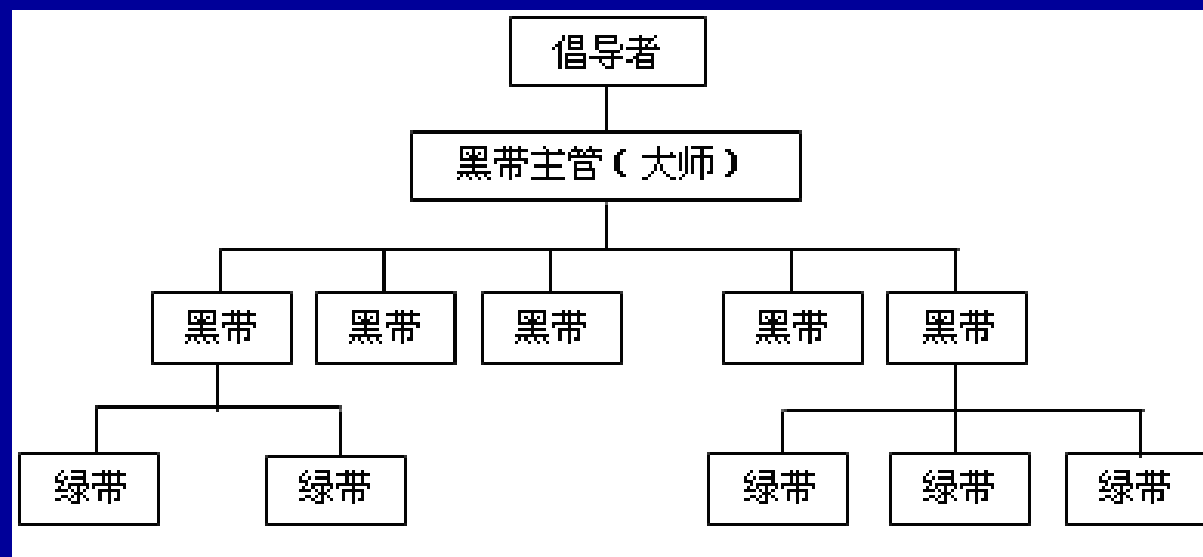
注册软件质量工程师是指充分理解软件质量的发展和实施, 对于软件的检验、测试、验证可以实现软件的开发和维护步骤及方法的专业人员。

企业内部认证

许多大型的软件企业根据自身需求, 设置了企业内部的认证。

3. 六西格玛角色和职责

- 倡导者
- 黑带大师
- 黑带
- 绿带



以黑带团队为基础的6sigma组织是领导职能推进6SIGMA方法的基础。它的重点在于建立和应用一些展开计划、报告系统和实施过程来支持PFSS（6sigma过程）和DFSS（策划）。

4. 六西格玛培训

(1) 高层管理和倡导者的培训

因为六西格玛管理实际上是自上而下的管理模式，高级管理层必须对六西格玛有正确、清晰的认识，并在整个实施过程中给与大力支持。

(2) 黑带/黑带大师和绿带培训

在六西格玛项目中，真正的执行人员是黑带和绿带。因此黑带和绿带需要通过培训掌握六西格玛基本概念，基本工具的使用等。

(3) 全体培训

在整个企业内部推行六西格玛文化，这是一个循序渐进的过程。

六、软件配置管理

6.1 什么是软件配置管理

6.2 软件配置管理的任务

6.3 版本控制

6.4 变更控制

6.5 配置状态报告

6.6 配置审计

6.1 什么是软件配置管理

协调软件开发使得混乱减到最小的技术叫做配置管理。

配置管理是一组标识、组织和控制修改的活动，目的是使错误达到最小并最有效地提高生产率。

6.1 什么是软件配置管理

软件配置管理，简称SCM，是一种“保护伞”活动，它应用于整个软件工程过程。

SCM活动的目标是为了

- (1) 标识变更；
- (2) 控制变更；
- (3) 确保变更正确地实现；
- (4) 向其他有关的人报告变更。

6.1 什么是软件配置管理

在软件工程过程中产生的所有信息项（文档、报告、程序、表格、数据）构成了软件配置。软件配置是软件的具体形态在某一时刻的瞬时影像。

随着软件工程过程的进展，软件配置项(SCI)数目快速增加。系统规格说明可繁衍出软件项目实施计划和软件需求规格说明。它们又依次繁衍出建立信息层次的其它文档。

6.1 什么是软件配置管理

基线 (Baseline)

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。

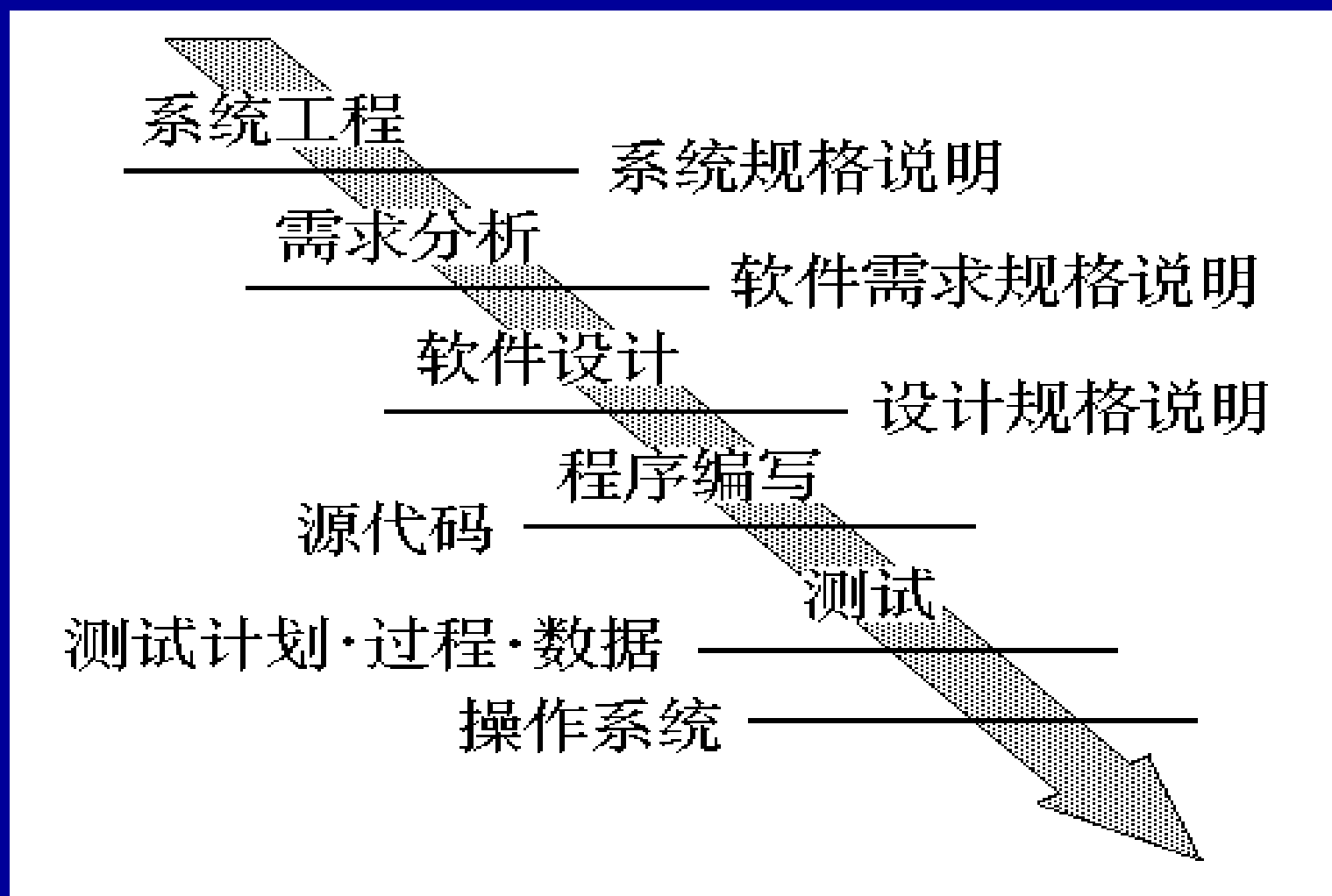
由正式的技术评审而得到的SCI协议和软件配置的正式文本才能成为基线。

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。

6.1 什么是软件配置管理



软件开发各阶段的基线



6.1 什么是软件配置管理

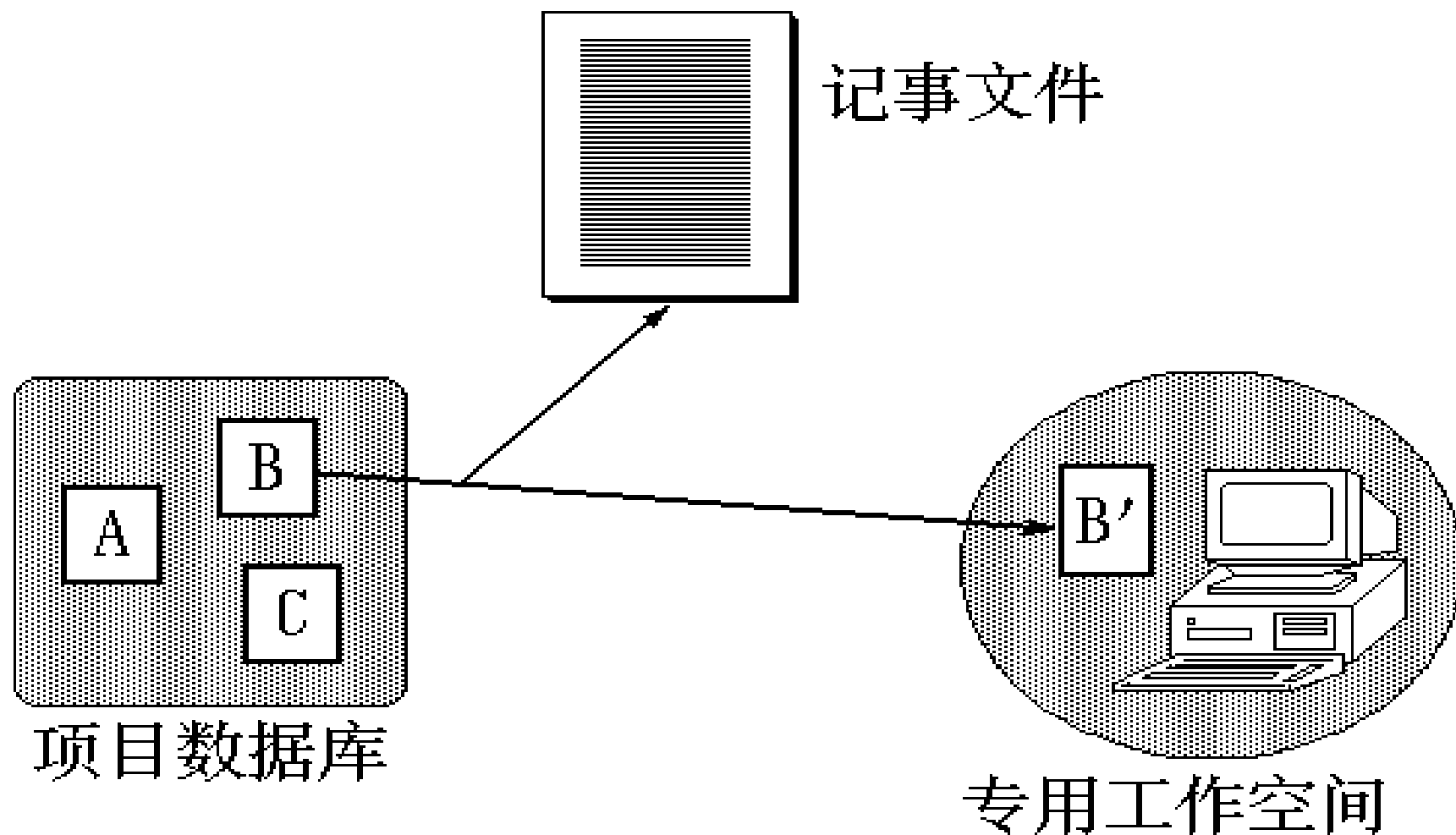
项目数据库

一旦一个SCI成为基线，就把它存放到项目数据库中。

当软件组织成员想要对基线SCI进行修改时，把它从项目数据库中复制到该工程师的专用工作区中。

6.1 什么是软件配置管理

基线SCI和项目数据库



软件配置项 SCI

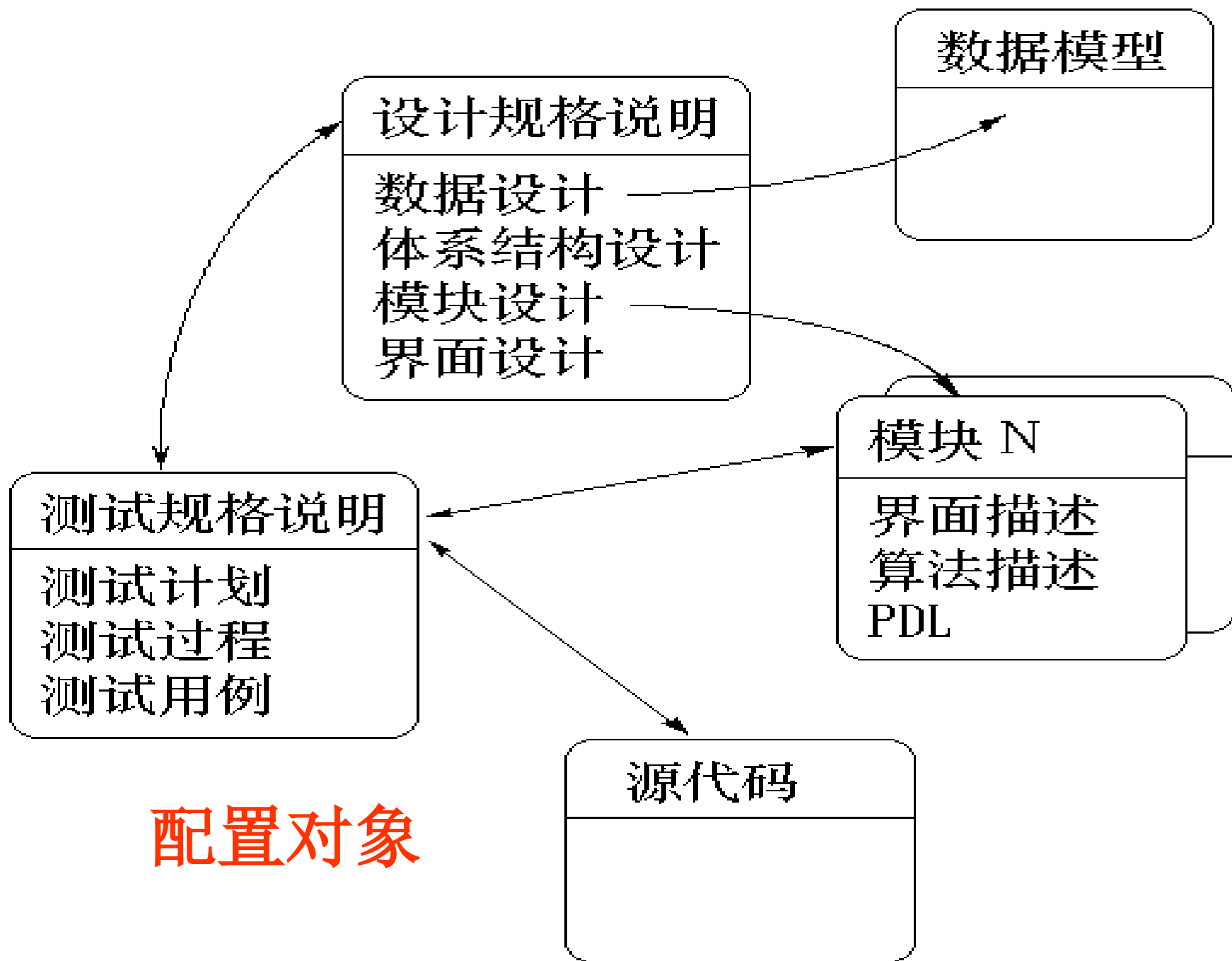
软件配置管理的对象就是软件配置项SCI。

配置对象

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。

每个对象与其它对象的联系用箭头表示。箭头指明了一种构造关系。



六、软件配置管理

6.1 什么是软件配置管理

6.2 软件配置管理的任务

6.3 版本控制

6.4 变更控制

6.5 配置状态报告

6.6 配置审计

六、软件配置管理

6.2 软件配置管理的任务

软件配置管理（SCM）的任务是：

- 标识单个的SCI
- 标识和管理软件各种版本
- 控制变更
- 审查软件配置
- 报告所有加在配置上的变更

配置标识

一方面随着软件生存期的向前推进，SCI的数量不断增多。

整个软件生存期的软件配置就象一部不断演变

的电影，而某一时刻的配置就是这部电影的一个片段。

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：是由软件工程师在分析、设计、编码和测试时所建立的文本单元。

复合对象：是基本对象或其它复合对象的一个收集。

对象标识：

（名字、描述、资源、实现）

对象的名字明确地标识对象。

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。

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。

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。

命名对象之间的联系

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。

E-R diagram 1.4 < is part of > data model;
data model < is part of > Design Specification;
就可以建立SCI的一个层次。

对象的相互关联关系：对象跨越对象层次的分支相互关联。这些交叉的结构联系表达方式如下：

data model <interrelated> data flow model;
(两个复合对象之间的相互联系)

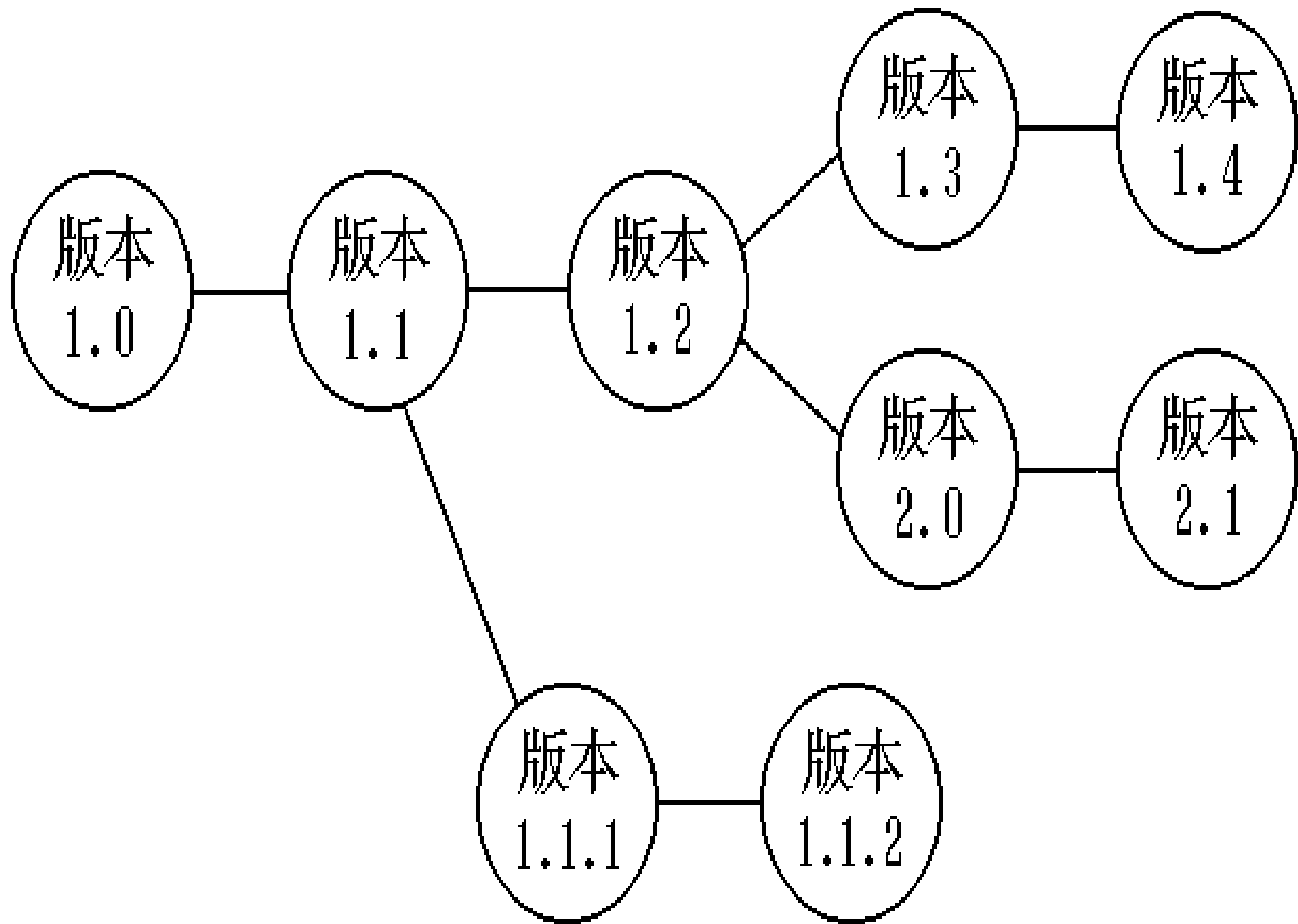
data model <interrelated> test case class m;
(一个复合对象与一个特定的基本对象之间的相互联系)

演变图

整个软件工程过程中所涉及的软件对象都必须加以标识。

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。

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。



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。

六、软件配置管理

6.1 什么是软件配置管理

6.2 软件配置管理的主要任务

6.3 版本控制

6.4 变更控制

6.5 配置状态报告

6.6 配置审计

6.3 版本控制

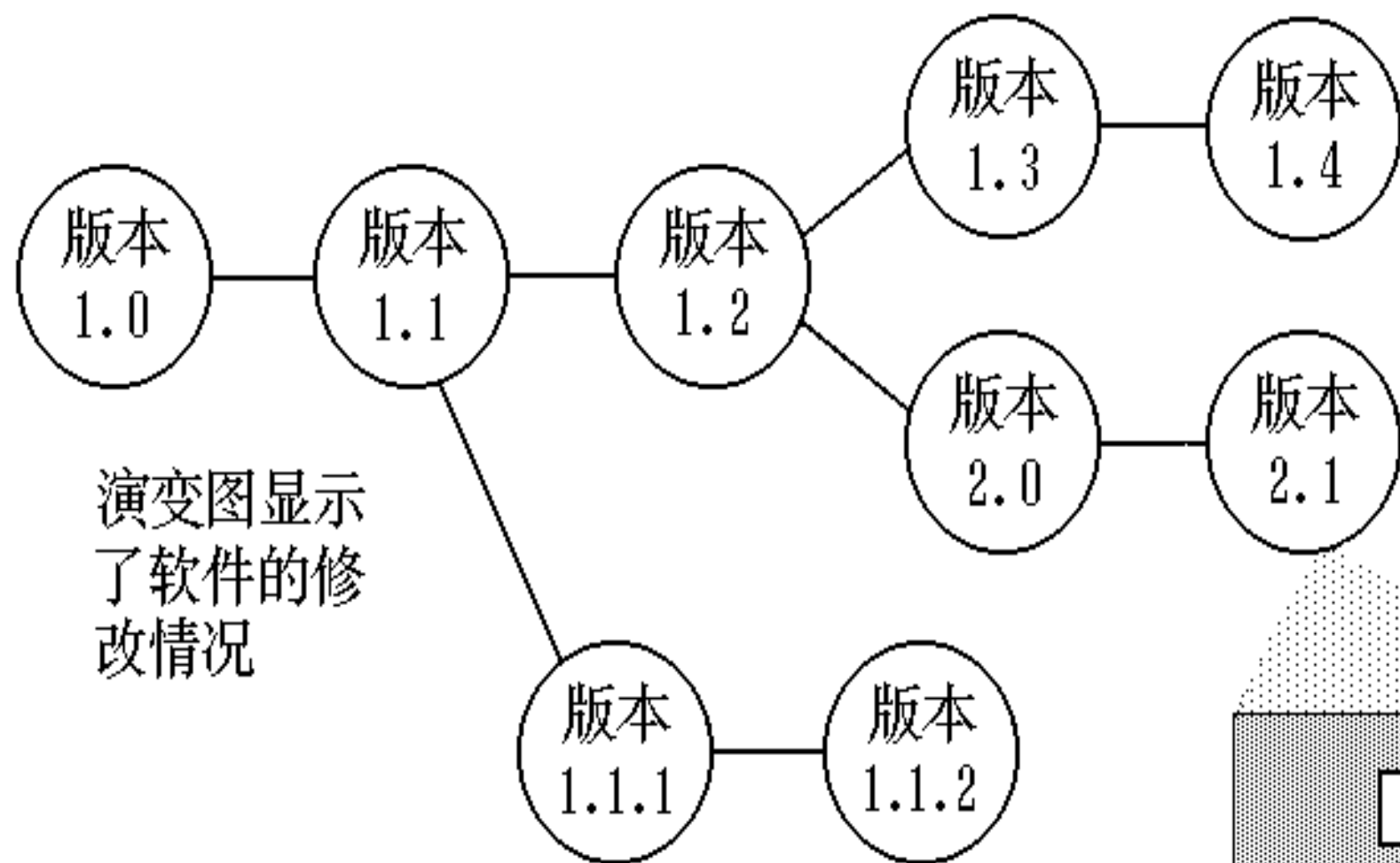
版本控制是SCM的基础，它管理并保护开发者的软件资源。

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。

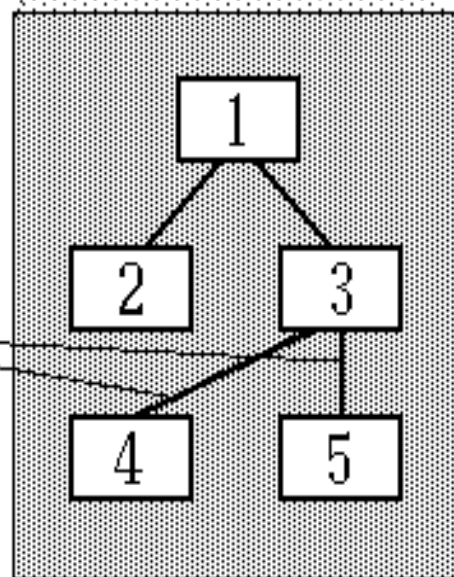
版本管理可以把一些属性结合到各个软件版本上。

通过描述所希望的属性集合来确定（或构造）所需要的配置。

使用演变图来表示系统的不同版本。



变种



版本管理的主要任务

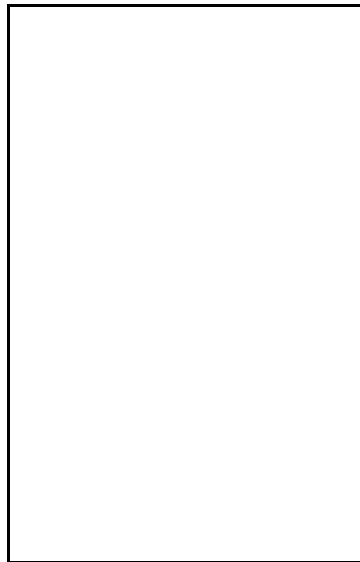
集中管理档案，安全授权机制：

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。

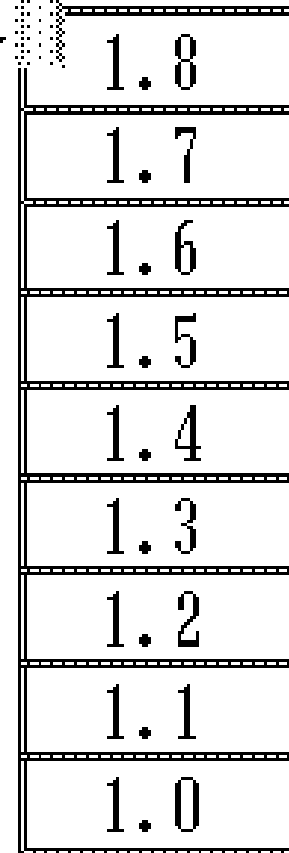
Workstation

LAN

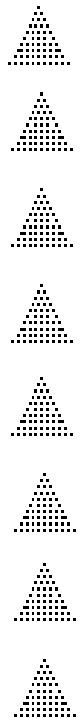


filename.cpp

GET
check in



1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1.0



filename.cpv

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。

UNIX

DOS

OS/2

1.5
1.4
1.3
1.2
1.1
1.0

主干

1.2.1.4
1.2.1.3
1.2.1.2
1.2.1.1
1.2.1.0

分支

1.2.1.1.1.3
1.2.1.1.1.2
1.2.1.1.1.1
1.2.1.1.1.0

分支

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。

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。

六、软件配置管理

6.1 什么是软件配置管理

6.2 软件配置管理的主要任务

6.3 版本控制

6.4 变更控制

6.5 配置状态报告

6.6 配置审计

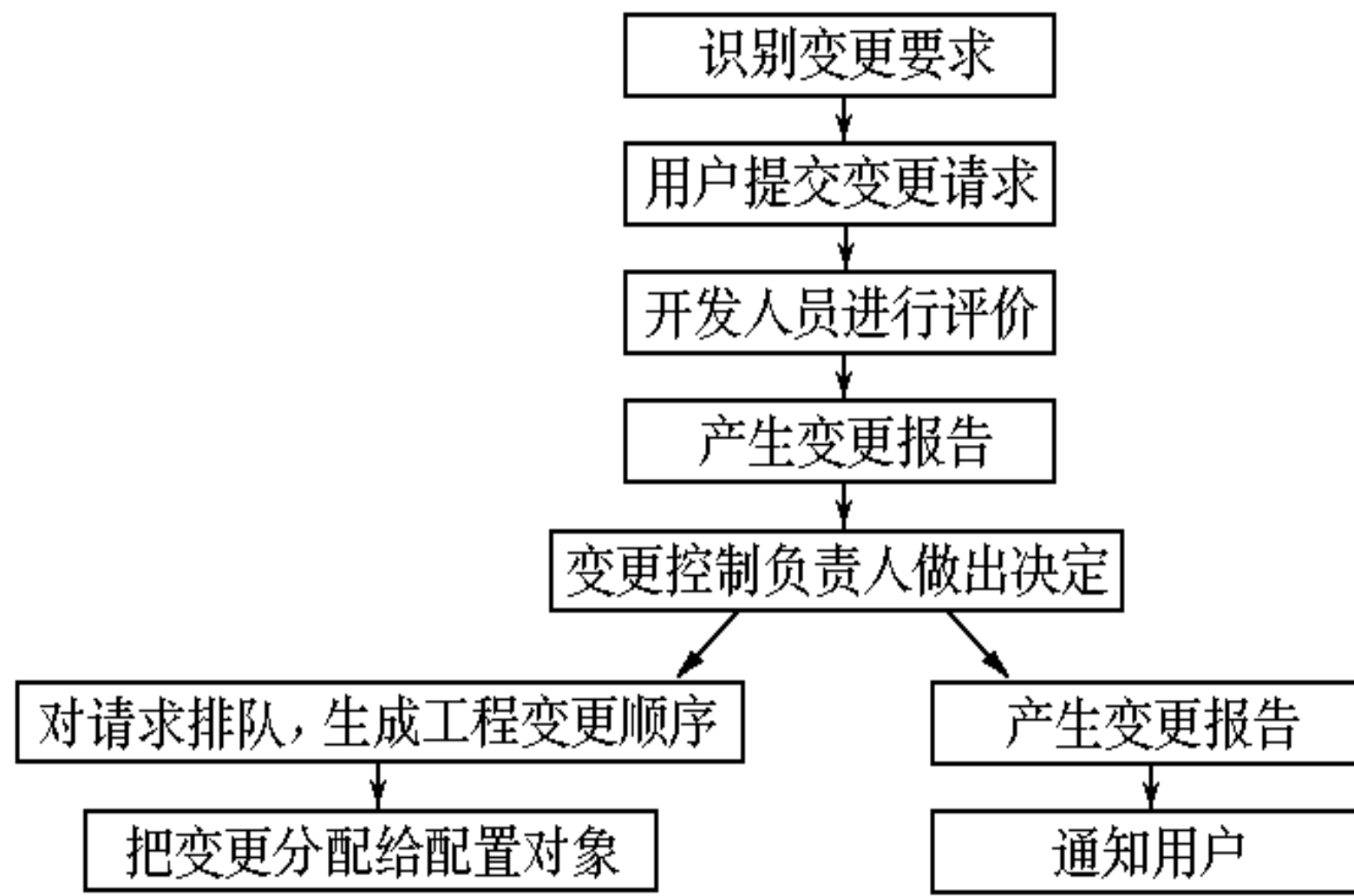
6.4 变更控制

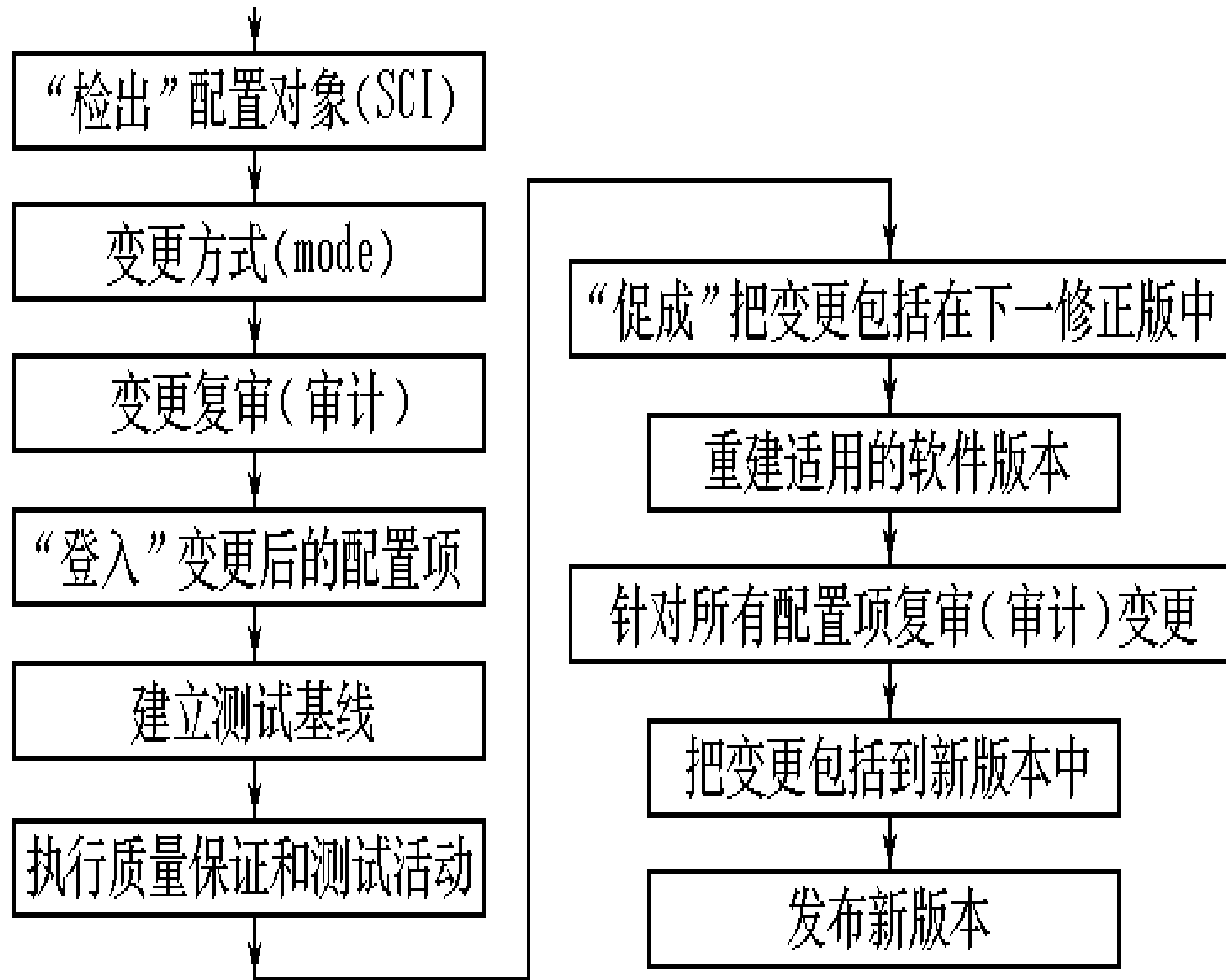
软件生存期内全部的软件配置是软件产品的真正代表，必须使其保持精确。

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。

变更控制包括建立控制点和建立报告与审查制度。

变更控制过程





软件变更有两类不同情况：

为改正小错误需要的变更。它是必须进行的，通常不需要从管理角度对这类变更进行审查和批准。但是，如果发现错误的阶段在造成错误的阶段的后面，例如在实现阶段发现了设计错误，则必须遵照标准的变更控制过程，把这个变更正式记入文档，把所有受这个变更影响的文档都做相应的修改。

为了增加或者删掉某些功能、或者为了改变完成某个功能的方法而需要的变更。这类变更必须经过某种正式的变更评价过程，以估计变更需要的成本和它对软件系统其它部分的影响。

如果变更的代价比较小且对软件系统其它部分没有影响，或影响很小，通常应批准这个变更。

如果变更的代价比较高，或者影响比较大，则必须权衡利弊，以决定是否进行这种变更。

如果同意这种变更，需要进一步确定由谁来支付变更所需要的费用。如果是用户要求的变更，则用户应支付这笔费用；否则，必须完成某种成本 / 效益分析，以确定是否值得做这种变更。

这种变更报告和审查制度，对变更控制来说起了一个安全保证作用。

在一个SCI成为基线之前，可以对所有合理的项目和技术申请进行非正式的变更；

一旦某个SCI经过正式的技术评审并得到批准，它就成了基线。以后如果需要对它变更，就必须得到项目负责人的批准，或者必须得到变更控制负责人的批准。

六、软件配置管理

6.1 什么是软件配置管理

6.2 软件配置管理的主要任务

6.3 版本控制

6.4 变更控制

6.5 配置状态报告

6.6 配置审计

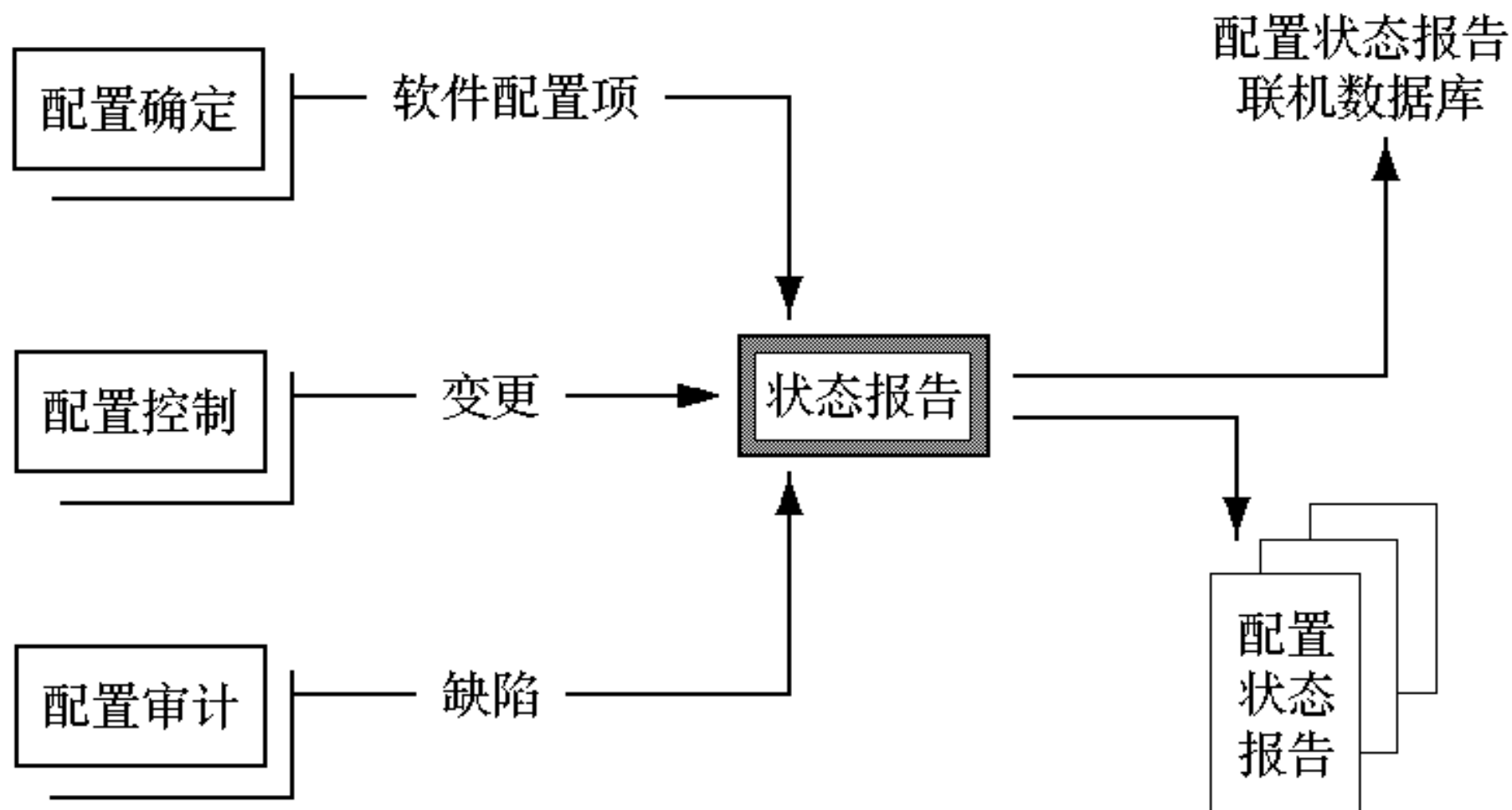
6.5 配置状态报告

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的任务。

登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。

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？

配置状态报告信息流



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。

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。

六、软件配置管理

6.1 什么是软件配置管理

6.2 软件配置管理的主要任务

6.3 版本控制

6.4 变更控制

6.5 配置状态报告

6.6 配置审计

6.6 配置审计

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。

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。

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
- 软件配置审计

正式的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，

评审者评价SCI，决定它与其它SCI的一致性，是否有遗漏或可能引起的副作用。

正式技术评审应对所有的变更进行，除了那些最无价值的变更之外。

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。

七、软件质量度量

7.1 软件质量的度量和评价

7.2 软件产品的质量度量

7.3 软件过程的质量度量

七、软件质量度量

7.1 软件质量的度量和评价

软件质量的度量

是软件属性的量化，是经验关系系统到数值关系系统的一种映射。软件质量特性度量有两类：预测型和验收型。

- 预测度量是利用定量的或定性的方法，对软件质量的评价值进行估计，以得到软件质量的比较精确的估算值。它是用在软件开发过程中的。
 - 尺度度量：这是一种定量度量。它适用于一些能够直接度量的特性，一般它作为相对量进行度量。如可靠性度量、复杂度度量、缺陷度量和规模度量等。
 - 二元度量：这是一种定性度量。它适用于一些只能间接度量的特性。
- 验收度量是在软件开发各阶段的检查点，对软件的要求质量进行确认性检查的具体评价，它可以看成是对预测度量的一种确认，是对开发过程中的预测进行评价。

7.1 软件质量的度量和评价

软件质量度量的实施

在确定要对一个软件（系统）进行度量之后，一般，采取以下几个步骤，来实施对该软件的度量：

- （1）确定软件质量需求
- （2）确定直接度量
- （3）分析度量结果
- （4）确认质量度量

软件质量评价

定量地评价软件的质量，目前还不能精确地做到。一般采用由若干（6~10）位软件专家进行打分来评价。这些软件专家应是富有实际经验的项目带头人。软件质量评价分两步走。

（1）评分

（2）分析结果

四层模型

- 软件质量的度量关键是要定出评定质量的指标和评定优劣的标准。有一种四层模型，第一层是软件产品质量(质量需求)，第二层是质量特性，第三层是质量子特性，第四层是度量指标。
- 软件产品质量层(质量需求)：是产品必须满足的质量需求。它是用用户术语描述的，主要有四点：
 - (1) 产品将在用户所在组织当前使用的平台和操作系统上运行。
 - (2) 产品将是可靠的并能防止数据丢失的机制。
 - (3) 产品将提供完成某些任务所必需的功能。
 - (4) 产品将易于使用。

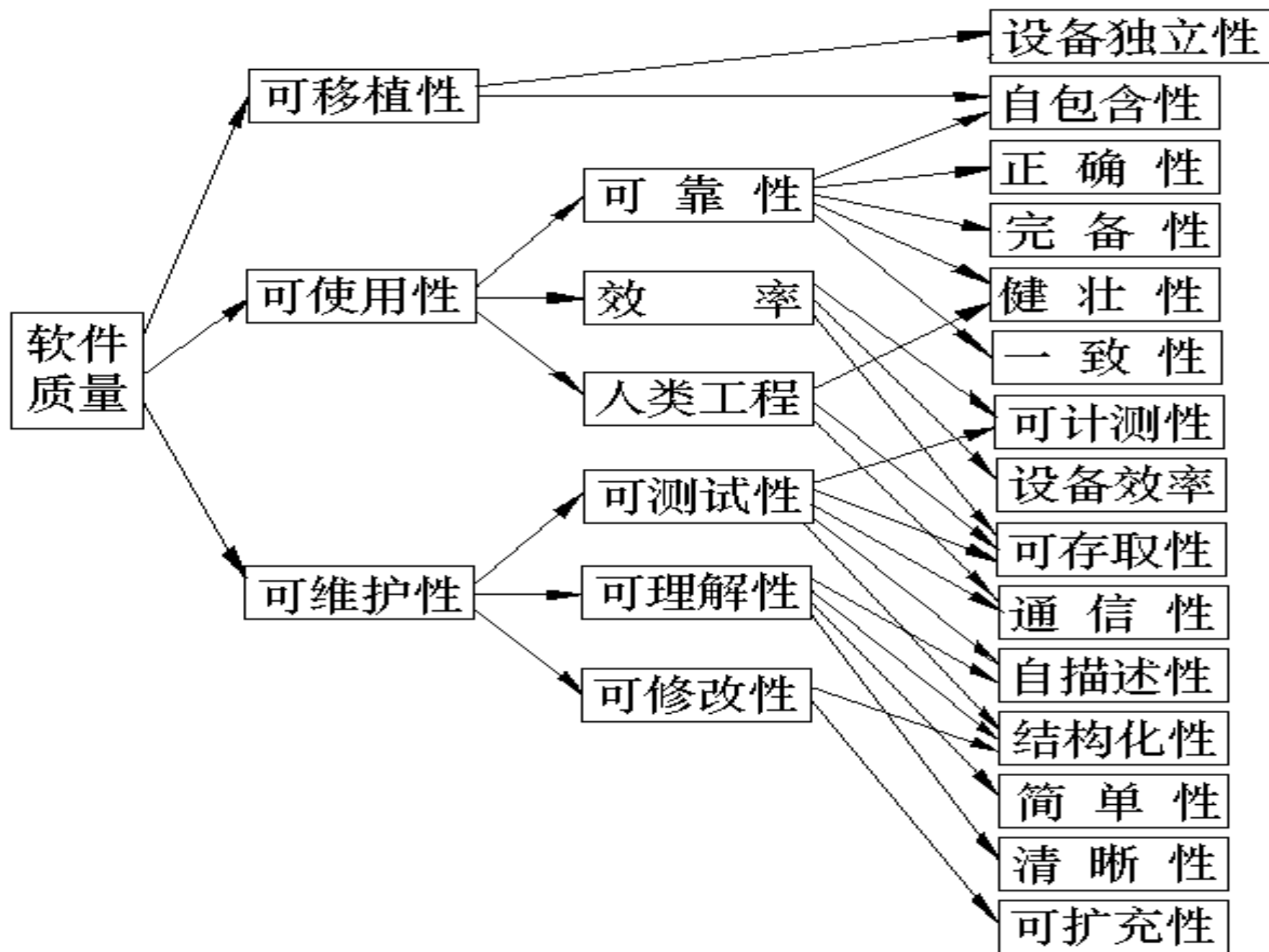
- 质量特性：表示与整个质量需求有关的特殊质量特性，它代表了用户的质量需求。它采用从用户角度考虑的立场，把软件质量分解成四类质量特性，这四个质量特性是软件的基本特征。IEEE的四个质量特性是：可移植性、可靠性、功能性、可使用性。
 - 可靠性：在规定的时间和条件下，软件所能维持其性能水平的程度。可靠性对某些软件是重要的质量要求，它除了反映软件满足用户需求正常运行的程度，且反映了在故障发生时能继续运行的程度。
 - 功能性：软件所实现的功能满足用户需求的程度。功能性反映了所开发的软件满足用户指明的或隐含的需求的程度，即用户要求的功能是否全部实现了。
 - 可移植性：从一个计算机系统或环境转移到另一个计算机系统或环境的容易程度。
 - 易使用性：对于一个软件，用户学习、操作、准备输入和理解输出时，所做努力的程度。易使用性反映了与用户的友善性，即用户在使用本软件时是否方便。

四层模型

质量需求	质量特性	质量子特性	直接度量	度量描述（例子）
产品将在多平台和当前用户正在使用的操作系统上运行	可移植性	硬件独立性	硬件依赖性	计算硬件的依赖性
		软件独立性	软件依赖性	计算软件的依赖性
		易安装性	安装时间	测量安装时间
		可重用性	能够用于其他应用软件中	计算能够或已经应用于其他软件系统的模块数量
产品将是可靠的并能提供防止数据丢失的机制	可靠性	无缺陷性	测试覆盖	测量测试覆盖度
			审查覆盖	计算已做过的代码审查模块
		容错性	数据完整性	统计用户数据被破坏情况
			数据恢复	测量恢复被破坏的数据的能力
		可用性	软件可用的百分比	软件可用时间除以总的软件使用时间

7.1 软件质量的度量和评价

质量需求	质量特性	质量子特性	直接度量	度量描述（例子）
产品将提供完成某些任务所必需的功能	功能性	完备性	测试覆盖	计算调用或分支测量覆盖
		正确性	缺陷密度	计算每一版本发布前的缺陷
		安全性	数据安全性	统计用户数据被破坏的情况
			用户安全性	没有被阻止的非法用户入侵数
		兼容性	环境变化	软件安装后必须修改的环境变量数量
		互操作性	混合应用环境下软件的可操作性	混合应用环境下可正确运行的数量
产品将易于使用	可使用性	易理解性	学习所用时间	新用户学习软件特性所花费的时间
		易学性	学习所用时间	新用户学会操作软件提供的基本功能所花费的时间
		易操作性	人的因素	新用户基于人类工程学对软件消极方面的评价数量
		沟通性	人的因素	新用户基于人类工程学对软件消极方面的评价数量



ISO的软件质量评价模型

按照ISO/TC97/SC7/WG3/1985-1-30/N382，软件质量度量模型由三层组成

- 软件质量需求评价准则（SQRC）
- 软件质量设计评价准则（SQDC）
- 软件质量度量评价准则（SQMC）

高层和中层建立国际标准，低层可由各使用单位视实际情况制定

可维护性(Maintainability)

可测试性(Testability)

灵活性(Flexibility)

互连性(Interoperability)

可移植性(Portability)

复用性(Reusability)

PRODUCT

PRODUCT

REVISION

TRANSITION

产品修正

产品转移

产品运行

PRODUCT OPERATIONS

正确性(Correctness)

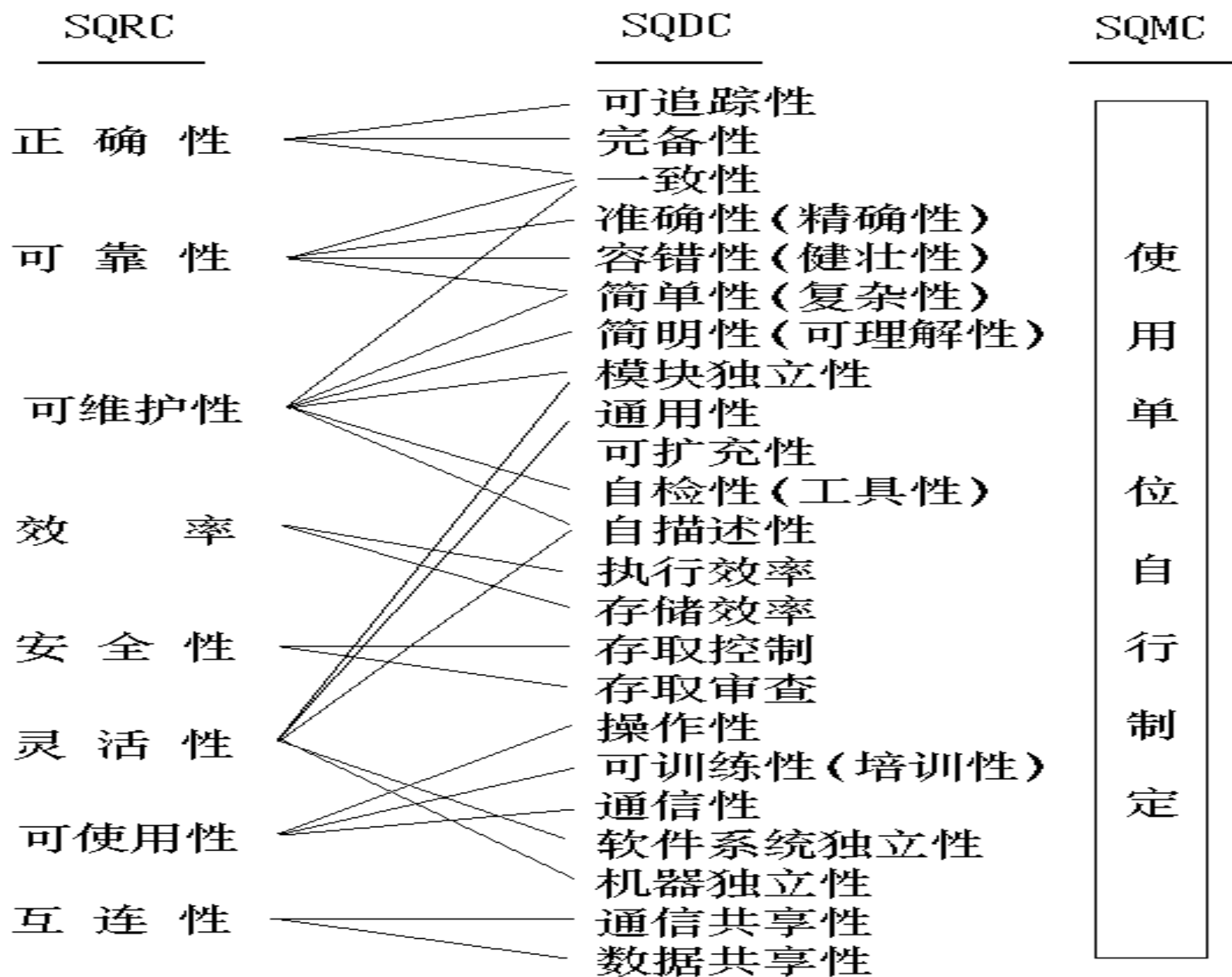
可使用性(Usability)

完整性(Integrity)

可靠性(Reliability)

效率(Efficiency)

Boehm质量模型



1991年 ISO质量特性国际标准（ISO/IEC9126）

质量特性：功能性、可靠性、可维护性、效率、可使用性、可移植性。

推荐21个子特性：适合性、准确性、互用性、依从性、安全性、成熟性、容错性、可恢复性、可理解性、易学习性、操作性、时间特性、资源特性、可分析性、稳定性、可变更性、可测试性、可安装性、可替换性、适应性、一致性。

	功能性	可靠性	可使用性	效 率	可维护性	可移植性
功 能 性		△			△	
可 靠 性				▽		△
可使用性				▽	△	△
效 率		▽			▽	▽
可维护性		△		▽		△
可移植性		▽		▽		

其中，△表示有利影响，▽表示不利影响。

软件质量特性度量有两类：预测型和验收型。

预测度量是利用定量或定性的方法，估算软件质量的评价值，以得到软件质量的比较精确的估算值。

验收度量是在软件开发各阶段的检查点，对软件的要求质量进行确认性检查的具体评价值，它是对开发过程中的预测进行评价。

预测度量有两种。

第一种叫做尺度度量，这是一种定量度量。它适用于一些能够直接度量的特性，例如，出错率定义为：错误数 / KLOC / 单位时间。

第二种叫做二元度量，这是一种定性度量。它适用于一些只能间接度量的特性，例如，可使用性、灵活性等等。

尺度度量检查表

评价 准则	度 量	需 求		设 计		编 码	
		是 / 否	值	是 / 否	值	是 / 否	值
程序 复杂性	每一模块的复杂性度量(McCabe)		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>
	$\text{系统复杂性度量} = \frac{\text{各模块复杂性度量之和}}{\text{系统模块数}}$		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>

二元度量检查表

评价 准则	度 量	需 求		设 计		编 码	
		是 / 否	值	是 / 否	值	是 / 否	值
设计 文档 的 完 备 性	(1) 无二义性引用(输入 / 功能 / 输出)	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	
	(2) 所有数据引用都可以从一个外部源定义、计算和取得	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	
	(3) 所有定义的功能都被使用	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	
	(4) 所有使用的功能都被定义	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	
	(5) 对每一个判定点, 所有的条件和处理都被定义	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	
	(6) 所有被定义、被引用的调用序列的参数一致	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	

7.2 软件产品的质量度量

1. 软件复杂性的度量
2. 软件缺陷度量
3. 顾客满意度度量

7.2 软件产品的质量度量

1. 软件复杂性的度量

McCabe环形计算复杂度

复杂度计算公式为：

$$M = V(G) = e - n + 2p$$

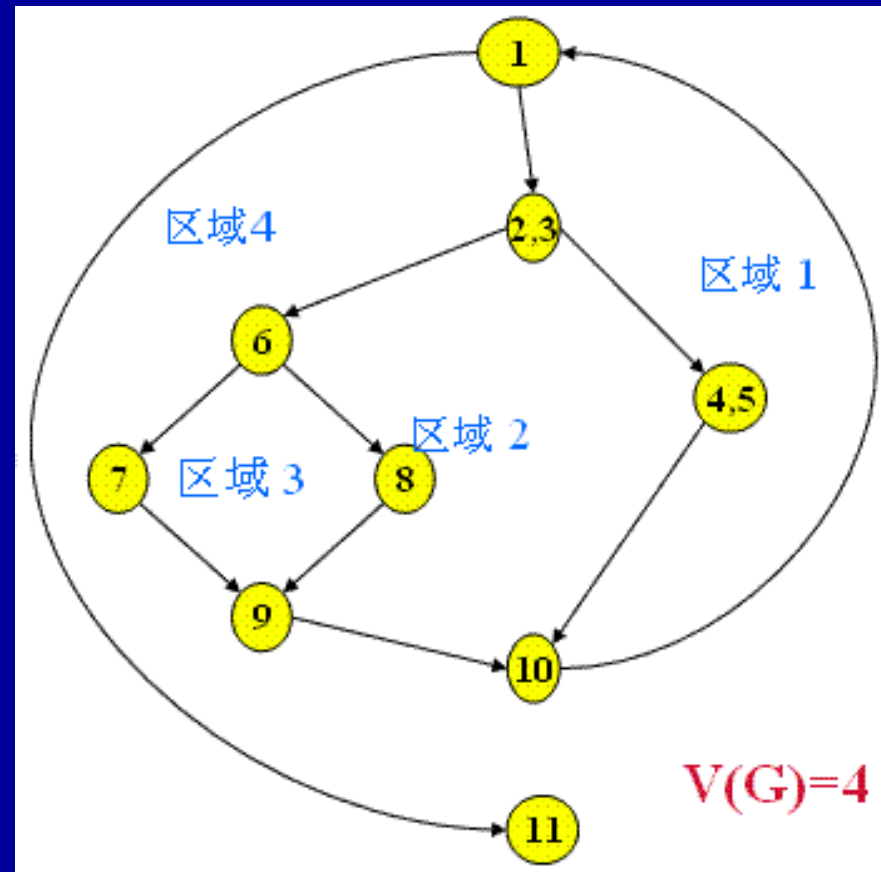
其中：

$V(G)$ 路径图的环形数目

e = 边的数目

n = 节点数目

p = 图中没有连接部分的数目



语法构造方法

基本思路是根据程序中可执行代码行的操作符和操作数的数量来计算程序的复杂性。操作符和操作数的量越大，程序结构就越复杂。

语法构造方法可以揭示程序中单独的语法构造和缺陷率之间的关系：

缺陷率= 0.15 + 0.23 DO WHILE + 0.22
SELECT + 0.07 IF-THEN-ELSE

结构度量方法

Henry给出的复杂性定义：

$$C_p = (\text{扇入} \times \text{扇出})^2$$

其中：

扇入 — 调用外部模块的模块数

扇出 — 被外部模块调用的次数

Card/Glass给出的度量模型：

$$C_t = S_t + D_t$$

其中：

S_t — 结构（模块间）复杂性

D_t — 数据（模块间）复杂性

2. 软件缺陷度量

缺陷密度——软件缺陷在规模上的分布

缺陷率——缺陷在时间上的分布

整体缺陷清除率

在软件开发过程中发现的所有缺陷数 / 发现的总缺陷数

阶段性缺陷清除率

$$\frac{\text{开发阶段清除的缺陷数}}{\text{产品中潜伏的缺陷数}} \times 100\%$$

3. 顾客满意度度量

软件组织的顾客满意度要素及其内容

顾客满意度要素	顾客满意度要素的内容
技术解决方案	质量、可靠性、有效性、易用性、价格、安装、新技术
支持与维护	灵活性、易达性、产品知识
市场营销	解决方案、接触点、信息
管理	购买流程、请求手续、保证期限、注意事项
交付	准时、准确、交付后过程
企业形象	技术领导、财务稳定性、执行印象

软件项目的顾客满意度要素及其内容

顾客满意度要素	顾客满意度度量内容
软件产品	功能性、可靠性、易用性、效率性、可维护性、可移植性
开发文档	文档的构成、质量、外观、图表以及索引、用语
项目进度以及交期	交期的根据、进度迟延情况下的应对、进展报告
技术水平	项目组的技术水平、项目组的提案能力、项目组的问题解决能力
沟通能力	事件记录、格式确认、问题解答
运用维护	支持、问题发生时的应对速度、问题解决能力

7.3 软件过程的质量度量

1. 软件需求过程的质量度量
2. 软件过程生产率的度量
3. 测试阶段的过程质量度量
4. 维护阶段的过程质量度量

7.3 软件过程的质量度量

1. 软件需求过程的质量度量

需求一致性度量

$$Q1 = n_{ui} / n_r$$

n_{ui} 是所有复审者都有相同解释的需求数目

n_r 是需求说明书中需求的个数，包含功能和非功能需求

需求完整性度量

$$Q2 = n_u / (n_i \times n_s)$$

n_u 是唯一功能需求的数目

n_i 是由需求规格定义或包含的输入的个数

n_s 是被表示的状态的个数。

需求确认程度度量

$$Q3 = n_c / (n_c + n_{nv})$$

n_c 是已经确认为正确的需求的个数

n_{nv} 是尚未被确认的需求的个数

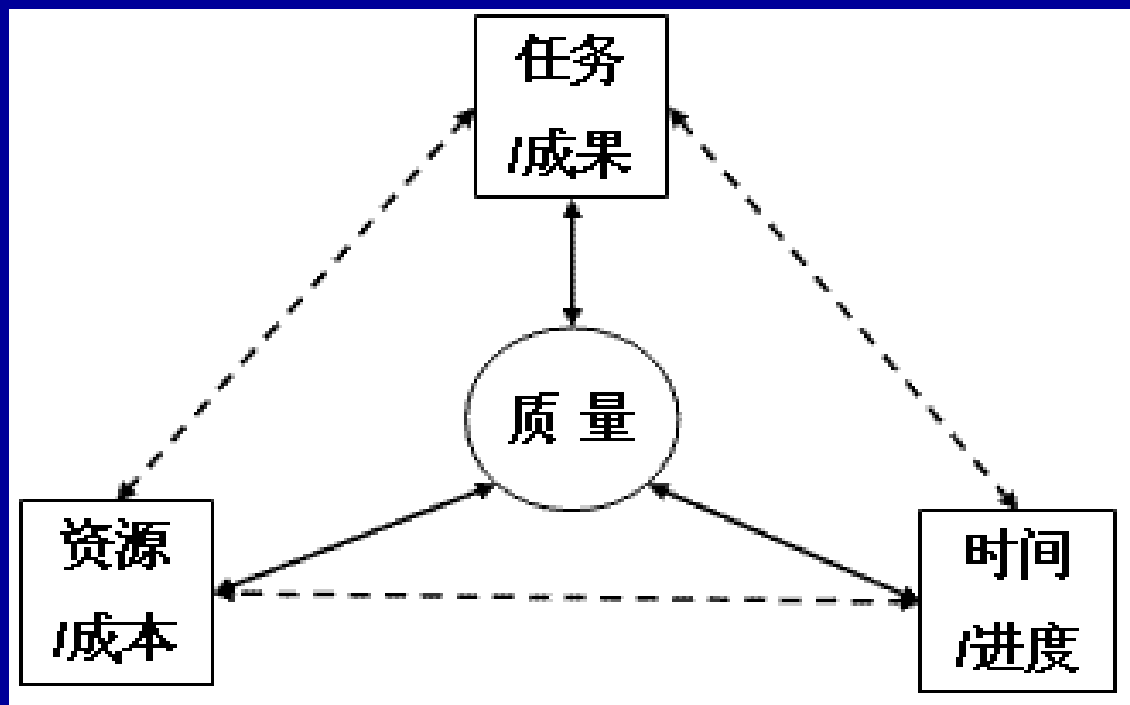
需求稳定性度量

需求稳定性度量是通过需求稳定因子RSI来表示：

$$RSI = \frac{\text{（所有确定的需求数} - \text{累计的需求变化请求数）}}{\text{所有确定的需求数}}$$

$$\text{所有确定的需求数} = \text{初始需求请求列表数} + \text{接受的需求变化请求数}$$

2. 软件过程生产率的度量



软件生产率的三维关系

度量量

代码行

功能点

类

测试用例

度量单位

人时 (man-hour)

人日 (man-day)

人月 (man-month)

人年 (man-year)

3. 测试阶段的过程质量度量

测试用例的深度 (TCD, Test Case Depth)

- 每KL0C的测试用例数
- 每个功能点/对象点的测试用例数

测试用例的有效性

- 每100或1000个测试用例所发现的缺陷数

测试用例的质量 (TCQ, Test Case Quality)

- 测试用例发现的缺陷数量/总的缺陷数量

测试执行的效率和质量

- 每个人日所执行的测试用例数
- 每个人日所发现的缺陷数
- 每修改的KLOC所运行的测试用例数

缺陷报告的质量

- 报告的质量不高的缺陷数/报告的总缺陷数
- 质量不高的缺陷包含：
- 1) 状态为“需要补充信息”的缺陷
 - 2) 状态为“不是缺陷”的缺陷

7.3 软件过程的质量度量

基于需求的测试覆盖

- 已执行的测试覆盖 = T_x / R_{ft}
- 成功的测试覆盖 = T_s / R_{ft}

T_x 表示已执行的测试过程数或测试用例数

T_s 是已执行的完全成功、没有缺陷的测试过程数或测试用例数

R_{ft} 是测试需求的总数

基于代码的测试覆盖

- 已执行的测试覆盖 = T_c / T_{nc}

T_c 是用代码语句、条件分支、代码路径、数据状态判定点或数据元素名表示的已执行项目数

T_{nc} (Total number of items in the code) 是代码中的项目总数

4. 维护阶段的过程质量度量

平均失效时间MTTF (mean time to failure)

基于时间缺陷（或用户问题数）的到达率

积压缺陷管理指标 (BMI)

软件成熟度指标 (SMI)

八、软件质量标准

8.1 标准与标准化

8.2 质量标准体系介绍

8.3 中国软件工程标准

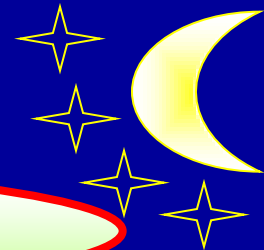
8.4 ISO9000族标准

8.5 CMM



八、软件质量标准

8.1 标准与标准化



标准

是一种文件，一种特殊文件。

为在一定的范围内获得最佳秩序，对活动或其结果规定共同的重复使用的规则、导则或特殊文件。该文件经协商一致制定并经一个公认机构的批准。

标准化

为在一定的范围内获得最佳秩序，对实际的或潜在的问题制定共同和重复使用的规则的活动。

标准的分级

国家标准、行业标准、地方标准、企业标准 www.yima.org.cn

标准的分类

(1) 根据适用范围分

国家标准：GB, GB/T, GB/Z, GSB；ANSI, BS, DIN, JIS。

行业标准：TB, TB/T；GJB, CMM/CMMI；IEEE, MIL_S, ASCE。

地方标准：DB××，DB××/T。

企业标准：Q/××× ××× - ×××；IBM, HP, NOKIA。

区域性标准：EN。

国际标准：ISO、IEC。

(2) 根据法律的约束性分

- ① 强制性标准：GB (国家技术法规)
- ② 推荐性标准：GB/T (指导性标准，自愿性文件)
- ③ 标准化指导性技术文件：GB/Z (尚未成为标准) 准标准

我国采用国际标准的程度

- 等同采用 (IDT, identical)
- 等效采用 (修改采用, MOD, modified)
- 参照采用 (非等效采用, NEQ, not equivalent)

八、软件质量标准

8.1 标准与标准化

8.2 质量标准体系介绍

8.3 中国软件工程标准

8.4 ISO9000族标准

8.5 CMM

8.2 软件质量标准体系介绍

标准体系空间

- ① 从纵向看，分为产品质量管理标准和过程质量管理标准
- ② 从横向看，分为通用标准和各个行业的质量标准
- ③ 从范围看，分为国际标准和国内标准

标准体系层次

- ① 原理标准，描述各个原理级的关键组织标准
- ② 要素标准，原理标准中的各个要素的详细性能要求的标准，必须执行；
- ③ 指南和补充，为如何把原理或要素标准应用于特定场合而提供指导性的文件

软件产品质量标准体系

	产品特性		软件产 品	产品文 档	功能规格
原理	9126-1				
要素标 准	TR 9126- 2/3/4	15026	12119	9127	14143-1/2
指南				18019	TR 14143- 3/4/5

软件过程质量标准体系

	软件过程						系统 过程
原理	12207/AMD1的过程结果						15288
要素 标准	12207 /14764	TR15846	TR16326	15939	14598	15910	15288 标准 部分
指南	TR15271	ISO9000-3	TR9294			18019	15288指南

八、软件质量标准

8.1 标准与标准化

8.2 质量标准体系介绍

8.3 中国软件工程标准

8.4 ISO9000族标准

8.5 CMM

八、软件质量标准

8.3 中国软件工程标准

分类	标准名称	标准号
基础标准	软件工程术语	GB/T 11457-2006
	信息处理-数据流程图、程序流程图、系统结构图、程序网络图和系统资源图的文件编辑符号及约定	GB/1526-89（ISO 5807-1985）
	软件工程标准分类法	GB/T 15538-1995
	信息处理-程序构造及其表示法的约定	GB 13502-92（ISO 8631）
	信息处理-单命中判定表规范	GB/T 15535-1995（ISO 5806）
	信息处理系统-计算机系统配置图符号及其约定	GB/T 14085-93（ISO 8790）
开发标准	信息技术 软件生存周期过程	GB/T 8566-2007
	计算机软件单元测试	GB/T 15532-95
	信息处理-按记录组处理顺序文卷的程序流程	GB（ISO 6593-1985）
	软件维护指南	GB/T 14079-93

8.3 中国软件工程标准

分类	标准名称	标准号
文档标准	软件文档管理指南	GB/T 16680-1996
	计算机软件文档编制规范	GB/T 8567-2006
	计算机软件需求说明编制指南	GB 9385-88
	计算机软件测试文件编制指南	GB 9386-88
管理标准	计算机软件配置管理计划规范	GB/T 12505-90
	信息技术 软件产品评价 质量特性及其使用指南	GB/T 16260-96
	计算机软件质量保证计划规范	GB/T 12504-90
	计算机软件可靠性和可维护性管理	GB/T 14394-93
	质量管理和质量保证标准 第三部分：在软件开发，供应和维护中的使用指南	GB/T 19000-3-94

信息技术 软件生存周期过程（GB/T 8566-2007）

- 软件过程：活动的一个集合；
- 活动：任务的一个集合；
- 任务：将一个输入转换为一个输出的操作。

按性质可分为三类过程：基本过程类

支持过程类

组织过程类

1〕基本过程类

包括：获取过程、供应过程、开发过程、运行过程、维护过程

2〕支持过程类

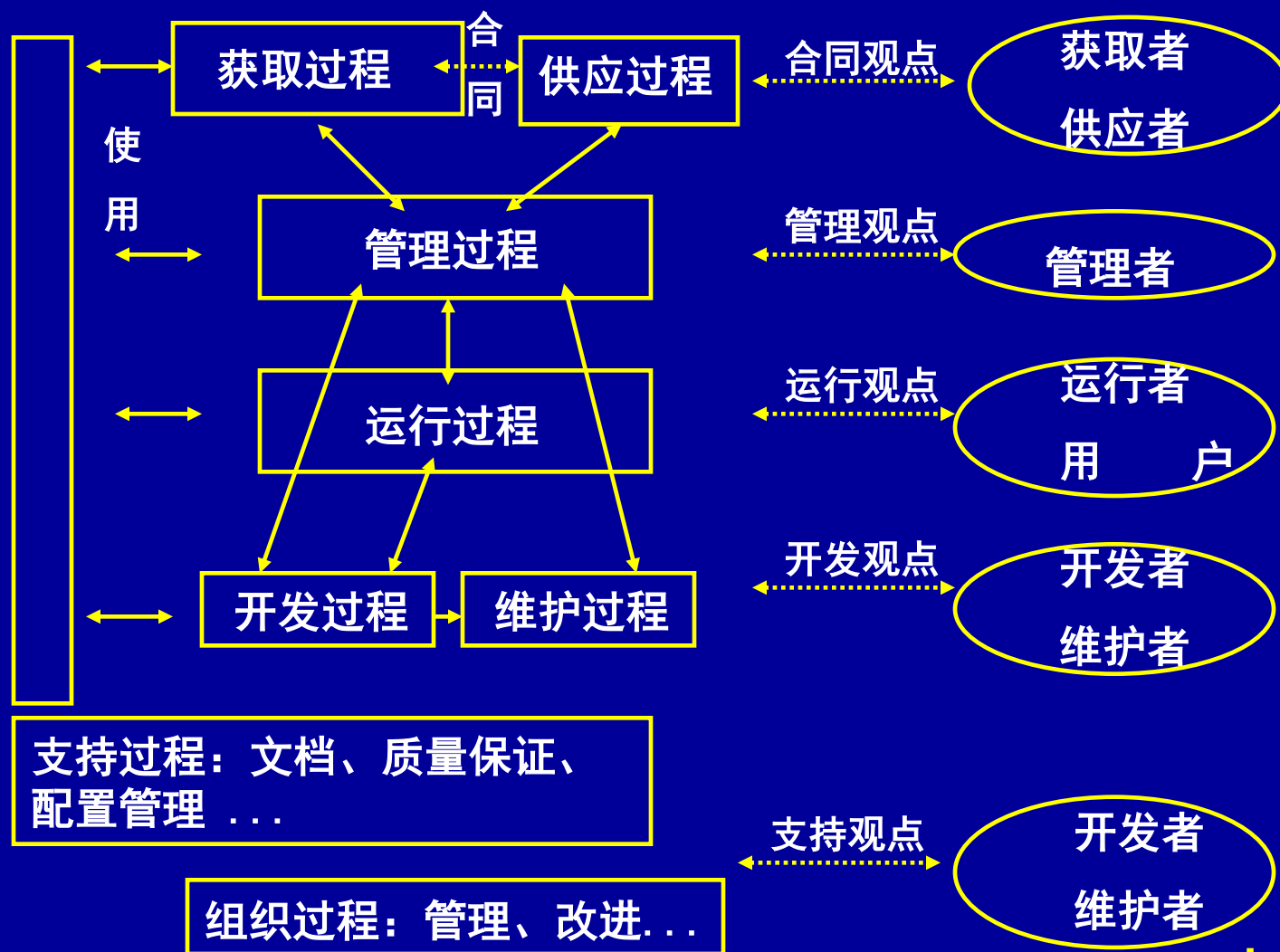
包括：文档过程、配置管理过程、质量保证过程、验证过程、确认过程、联合评审过程、审计过程、问题解决过程等。

3〕组织过程类

包括：管理过程、基础设施过程、培训过程、改进过程

剪裁过程：对软件过程和活动实施剪裁。将一选定的模型以及相关标准应用于某一领域或具体的软件项目，形成该领域的模型及标准，或该软件项目的软件过程和活动。

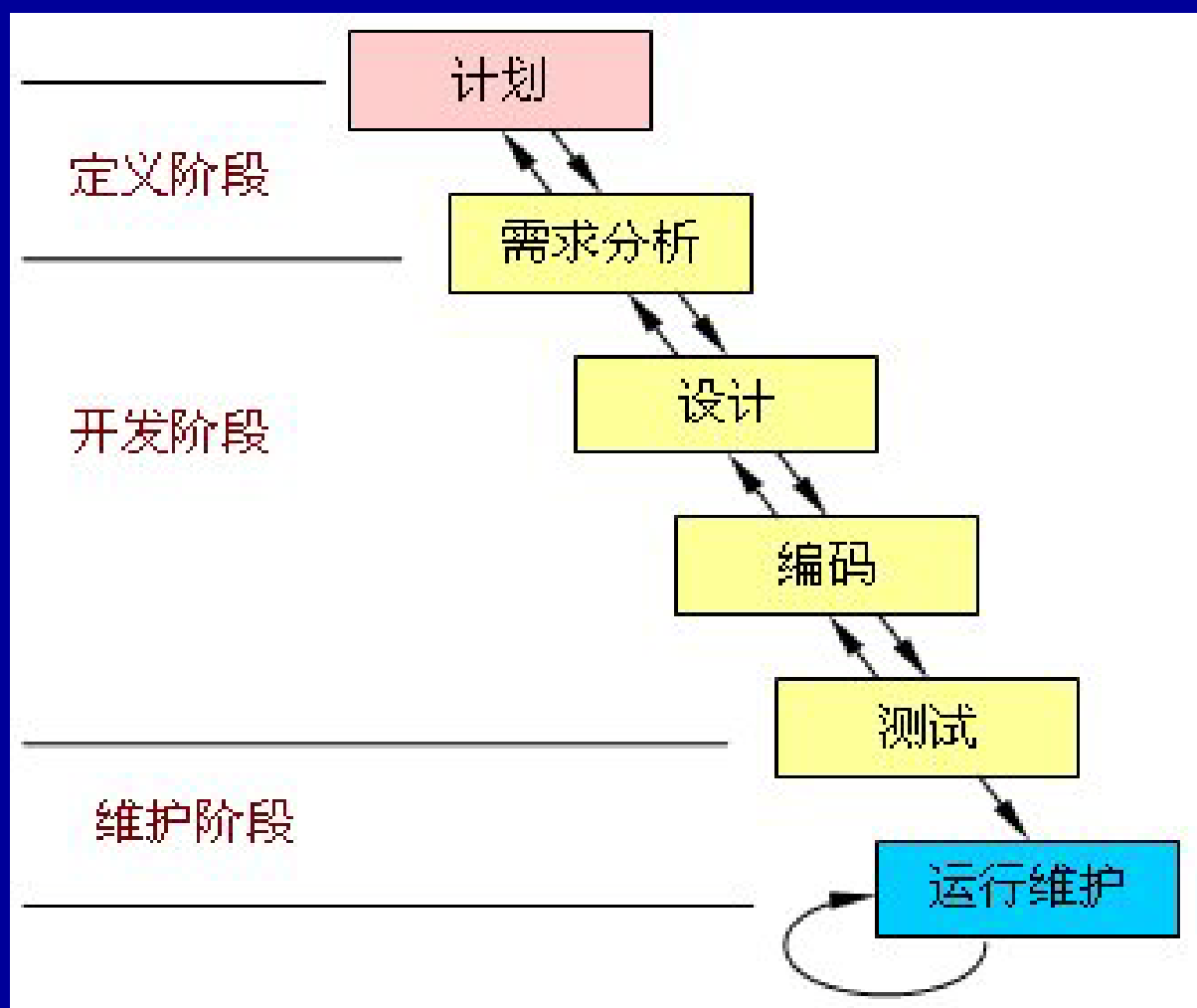
3) 软件过程之间的关系



为解决软件危机，人们提出用工程化的原则及方法来组织软件开发工作，这就是软件工程的由来。在软件工程中可将软件的生存周期分为6个阶段（传统生命周期，即瀑布模型）

- (1) 计划 (Planning)
- (2) 需求分析 (Requirement Analysis)
- (3) 设计 (Design)
- (4) 编码 (Coding)
- (5) 测试 (Testing)
- (6) 运行与维护 (Run and Maintenance)

软件开发瀑布模型



计算机软件文档编制规范（GB/T 8567-2006）

- 文档过程
- 文档编制要求
- 文档编制格式
- 面向对象软件的文档编制

八、软件质量标准

8.1 标准与标准化

8.2 质量标准体系介绍

8.3 中国软件工程标准

8.4 ISO9000族标准

8.5 CMM

8.4 ISO9000族标准

在质量管理八大原则的指导下，TC176的专家们，起草了ISO9001：2000FDI版，并在2000年获得表决通过。

■ ISO9001：2008

1. ISO9000族标准及其文件结构
2. ISO9000系列标准的内容
3. ISO9000系列标准的特点
4. ISO9000系列标准流行的原因
5. 制定与实施ISO9000系列标准的主导思想

8.4 ISO9000族标准

1.ISO9000族标准

“ISO9000”不是指一个标准，而是一族标准的统称。ISO9000族标准是ISO/TC176（第176个技术委员会，质量管理与质量保证技术委员会）制定的所有国际标准。

ISO9000族标准的基本思想

其一是控制的思想；其二是预防的思想。

ISO9000族标准的文件结构



2. ISO9000系列标准的内容

ISO9000	质量管理体系	基础和术语
ISO9001	质量管理体系	要求
ISO9004	质量管理体系	业绩改进指南

GB/T 19000 idt ISO 9000

GB/T 19001 idt ISO 9001

GB/T 19004 idt ISO 9004

注：GB，国家标准；T，推荐；idt，等同采用。

GB/T19001标准的目的

阐明满足顾客和适用的法律法规的质量管理体系的最低要求

本要求是通过满足顾客要求而使顾客满意

组织达到本要求，表明其具备了二个能力：

- 稳定提供合格产品的能力

- 可增强顾客满意的能力

ISO9000系列标准内容的高度概括：

“有章可循，有章必依，有据可查，有人负责”。

该说到的要说到，说到的要做到。

按照自己所说的去做，将自己所做的写下来。

说你所做；写你所说；做你所写。

要做到说、写、做一致。

ISO9001的内容

- 0 引言
- 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 产品实现
- 8 测量、分析和改进

8.4 ISO9000族标准

3. ISO9000系列标准的特点

- (1) 强调对各部门的职责权限进行明确划分、计划和协调；
- (2) 强调管理层的介入；
- (3) 强调纠正及预防措施；
- (4) 强调不断的审核及监督；
- (5) 强调全体员工的参与及培训；
- (6) 强调文化管理。

ISO9000族标准的特点

1. 通用性强
2. 结构简化
3. 以八项质量管理原则作为标准的基础
4. 更明确地将质量管理体系与过程联系起来，强调识别与策划
5. 更强调了最高管理者的作用
6. 突出了持续改进
7. 强调了效果评价
8. 考虑了各相关方的利益

8.4 ISO9000族标准

标准的优点

- (1) 适用于各种组织
- (2) 满足多个行业
- (3) 易于使用、易于翻译、容易理解
- (4) 减少了强制性的“程序”
- (5) 将管理体系过程联系起来
- (6) 强调了持续改进
- (7) 强调了顾客满意
- (8) 与ISO14000具有更好的兼容性
- (9) 考虑了相关方利益

ISO9001的特点和作用

- (1) 可作为供方质量保证工作的依据，也是评价供方质量体系依据。
- (2) 企业申请ISO9000族质量体系认证的依据。
- (3) 开发/设计、生产、安装和服务的质量保证模式。
- (4) 要求提供全过程严格控制的依据。
- (5) 要求供方贯彻“预防为主、检验把关相结合”的原则。

4. ISO9000系列标准流行的原因

- (1) 市场经济，特别是国际贸易的驱动。
- (2) ISO9000系列标准适用领域广阔。

5. 制定与实施ISO9000系列标准的主导思想

- (1) 强调质量并非在产品检验中得到，而是形成于生产的全过程。
- (2) 要求必须使影响产品质量的全部因素在生产全过程中始终处于受控状态。
- (3) 要求证实企业具有持续提供符合要求产品的能力。
- (4) 强调质量管理必须坚持进行质量改进。

八、软件质量标准

8.1 标准与标准化

8.2 质量标准体系介绍

8.3 中国软件工程标准

8.4 ISO9000族标准

8.5 CMM

8.5 CMM

能力成熟度模型（CMM）简介

1) 问题的提出

计算机软件的开发一直是广泛应用计算机的瓶颈。

在80年代中期，美国工业界和政府部门开始认识到，在软件开发中，关键的问题在于软件开发组织不能很好地定义和控制其软件过程。

针对这一问题：

1986年11月，美国卡内基-梅隆大学软件工程研究所（SEI）开始开发过程成熟度框架。

1987年9月，SEI发布了过程成熟度框架的简要描述和成熟度调查表。

1991年，SEI将过程成熟度框架演化为CMM 1.0版：CMU/SEI-91-TR-24、CMU/SEI-91-TR-25。

1993年，SEI根据反馈，提出CMM 1.1版：CMU/SEI-93-TR-25。目前，已经提出CMM 2.0版。

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。

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：

过程成熟度的基本概念

①软件过程能力

描述(开发组织或项目组)通过遵循其软件过程能够实现预期结果的程度。

②软件过程性能

表示(开发组织或项目组)遵循其软件过程所得到的实际结果。

③软件过程成熟度

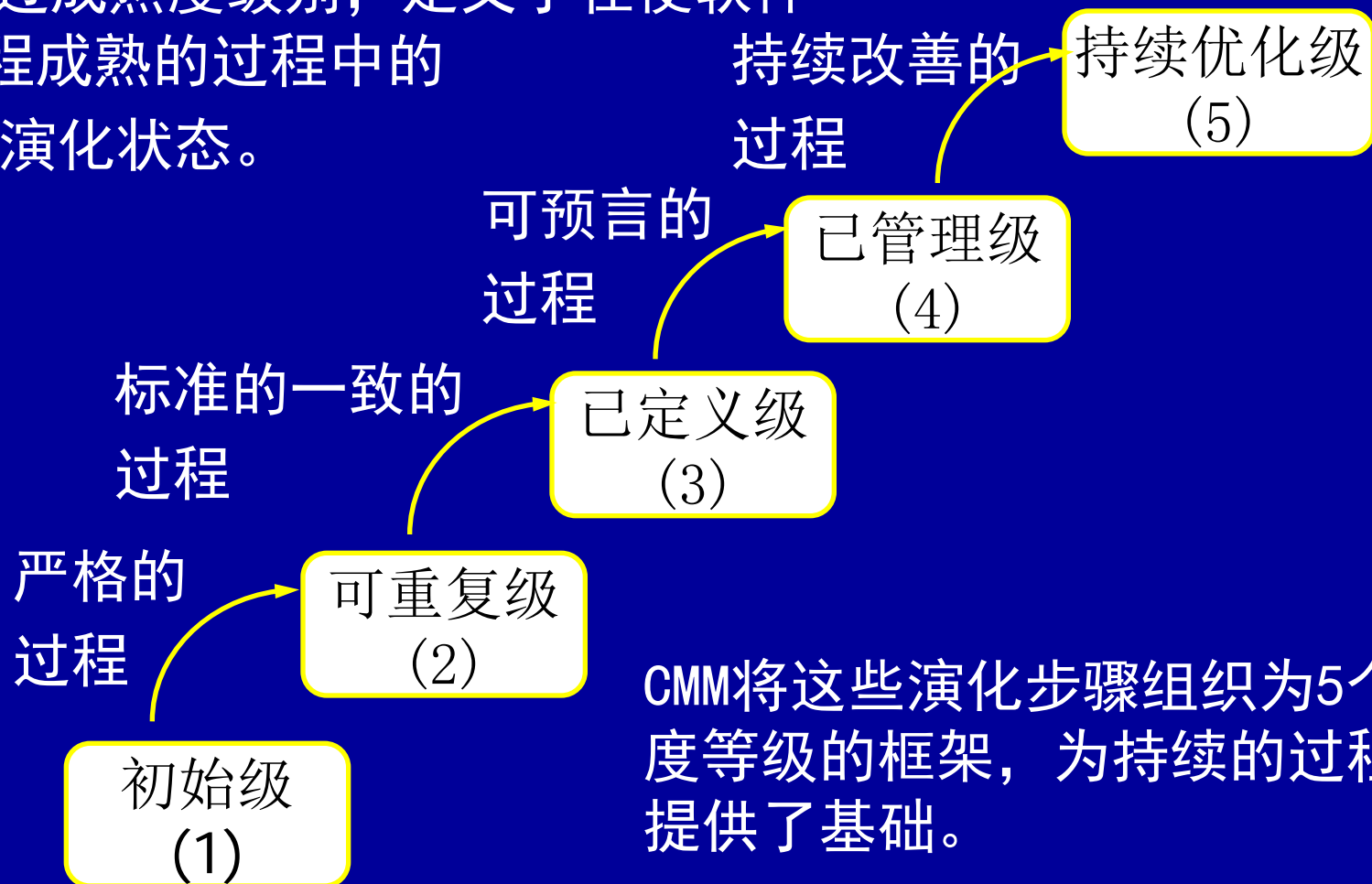
一个特定软件过程被明确和有效地定义、管理、测量和控制的程度。

④软件能力成熟度等级

软件开发组织在走向成熟的过程中,几个具有明确定义的、可以表征其软件过程能力成熟程度的“平台”。

CMM的软件过程成熟度框架

通过成熟度级别，定义了在使用软件过程成熟的过程中的演化状态。

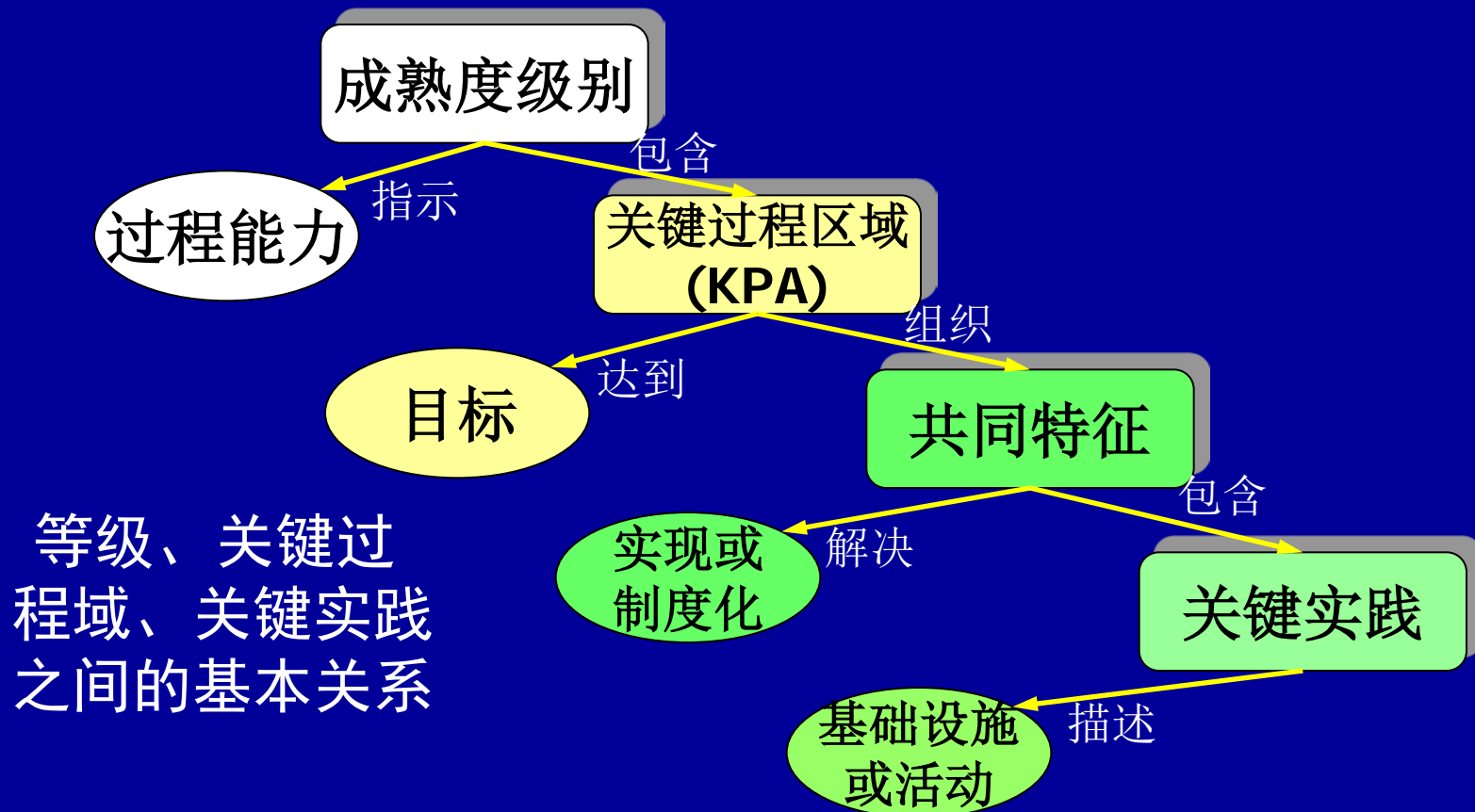


CMM将这些演化步骤组织为5个成熟度等级的框架，为持续的过程改进提供了基础。

软件过程成熟度模型

CMM的每个等级是通过三个层次加以定义的：

关键过程域 关键实践类 关键实践



例如：软件项目规划

- ① 所属等级
- ② 目的
- ③ 目标
- ④ 实现目标的关键实践

总结：各级包含的关键过程域：

可重复级：6个

软件配置管理，软件质量管理，子产品工程
项目跟踪和监督，软件项目规划，需求管理

已定义级：6+7（个）=13个

对等复审，组间协作，软件产品工程，
集成的软件管理，培训计划，组织过程定义，
组织过程焦点

已管理级：13+2（个）=15个

软件质量管理，量化的过程管理

持续优化级：15+3（个）=18个

过程变化管理，技术变化管理，缺陷预防

基本概念

过程：为达到目的而执行的所有步骤的序列。

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。

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。

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。

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。当一个企业达到了一定的软件过程成熟级别后，它将通过制定策略、建立标准和确立机构结构使它的软件过程制度化。而制度化又促使企业通过建立基础设施和企业文化来支持相关的方法、实践和过程。从而使之可以持续并维持一个良性循环。

软件机构的成熟性

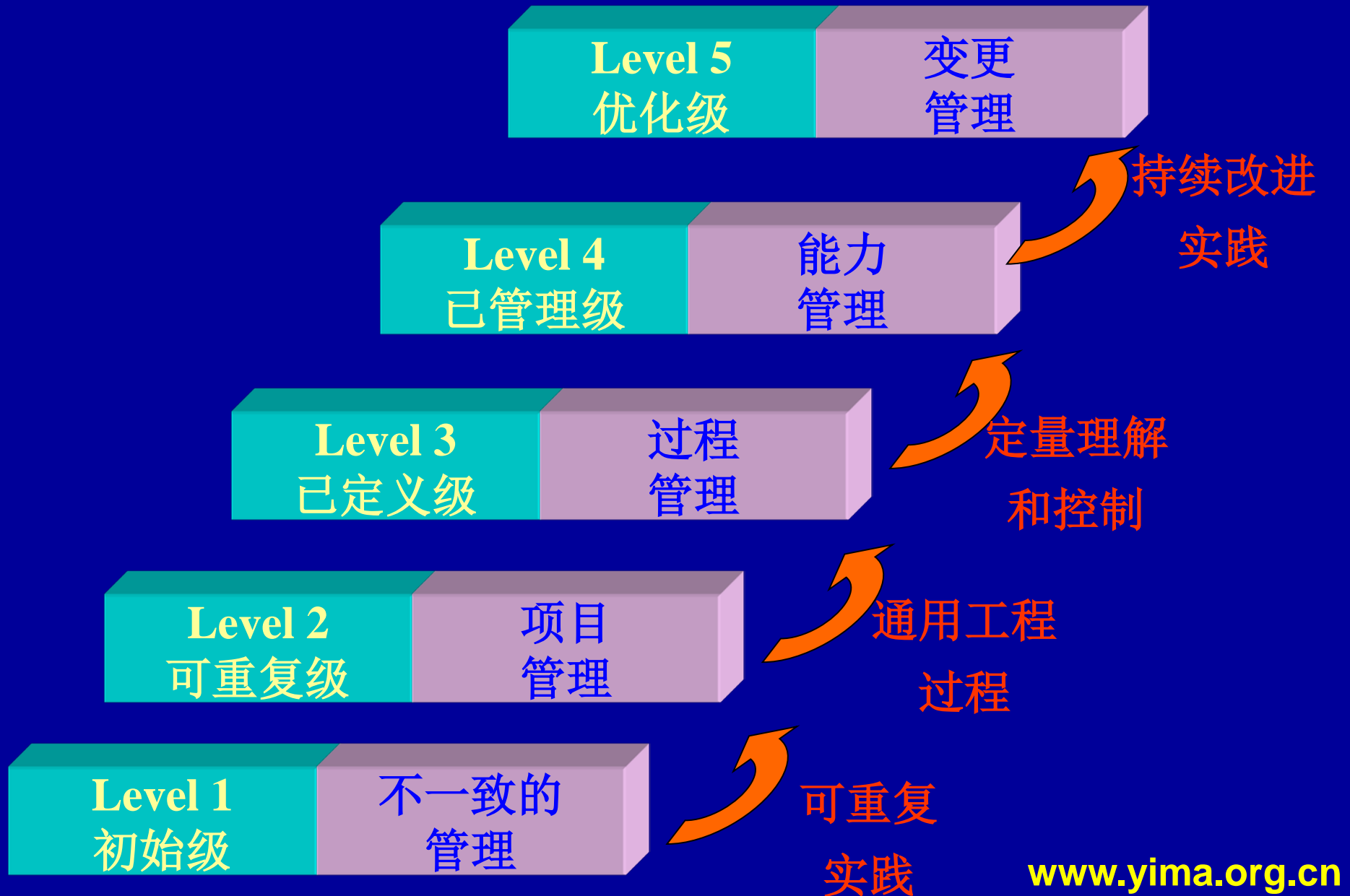
对于不同的软件开发机构，在组织人员完成软件项目中所依据的管理策略有很大的差别，因而软件项目所遵循的软件过程也有很大差别。在此，可用软件机构的成熟度（Maturity）加以区别。

不成熟软件机构的特征：

- (1) 软件过程一般在项目进行中由参与开发的人员临时确定。有时即使确定了，实际上也并不严格执行；
- (2) 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；
- (3) 项目的进度和经费预算由于估计的不切实际，所以常常突破。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；
- (4) 产品质量难以预测。质量保证活动，如质量评审、测试等，常被削弱或被取消。

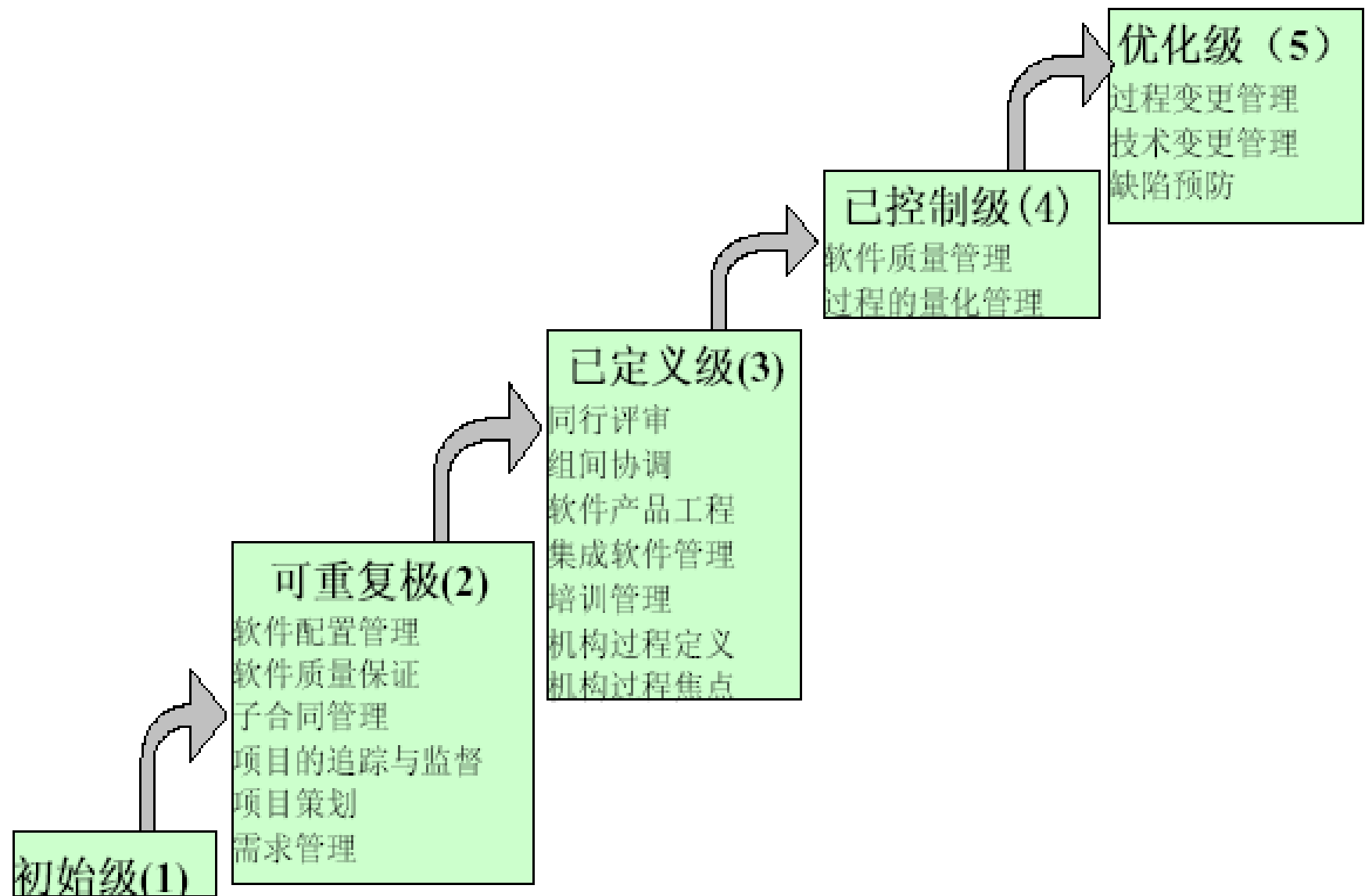
成熟软件机构具有的特征：

- (1) 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；
- (2) 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；
- (3) 软件产品的质量 and 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；
- (4) 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。



关键过程域

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，称为关键过程域（KPA Key Process Area）。每一个关键过程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域。



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。

利用CMM对软件机构进行成熟度评估

评估过程有以下几步：

- (1) 建立评估组。
- (2) 评估组准备。
- (3) 项目准备。
- (4) 进行评估。
- (5) 初评。
- (6) 讨论结果。
- (7) 给出最后的结论。

九、软件评审

9.1 为什么需要评审

9.2 软件评审的角色和职能

9.3 评审的内容

9.4 评审的方法和技术

9.5 准备评审会议

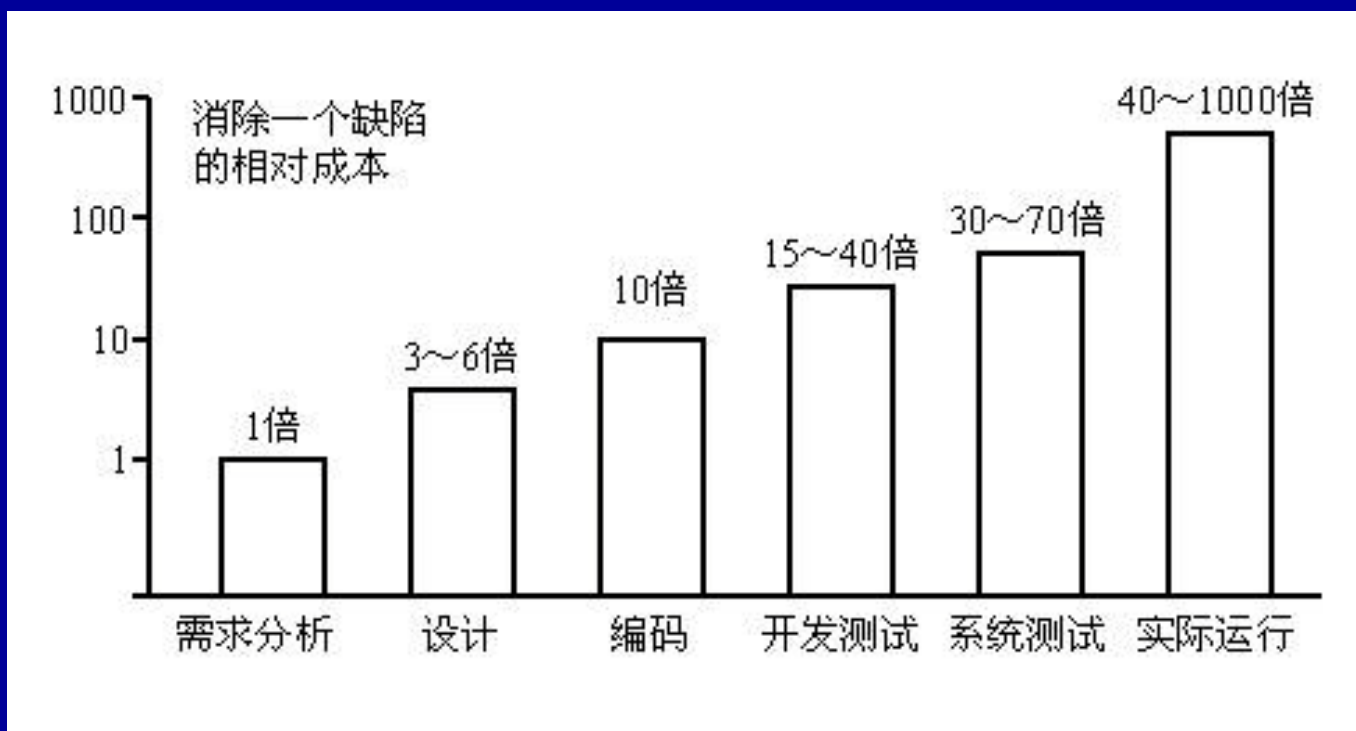
9.6 召开评审会议

9.7 跟踪和分析评审结果

9.8 如何实施成功的评审

9.1 为什么需要评审 从成本上来衡量

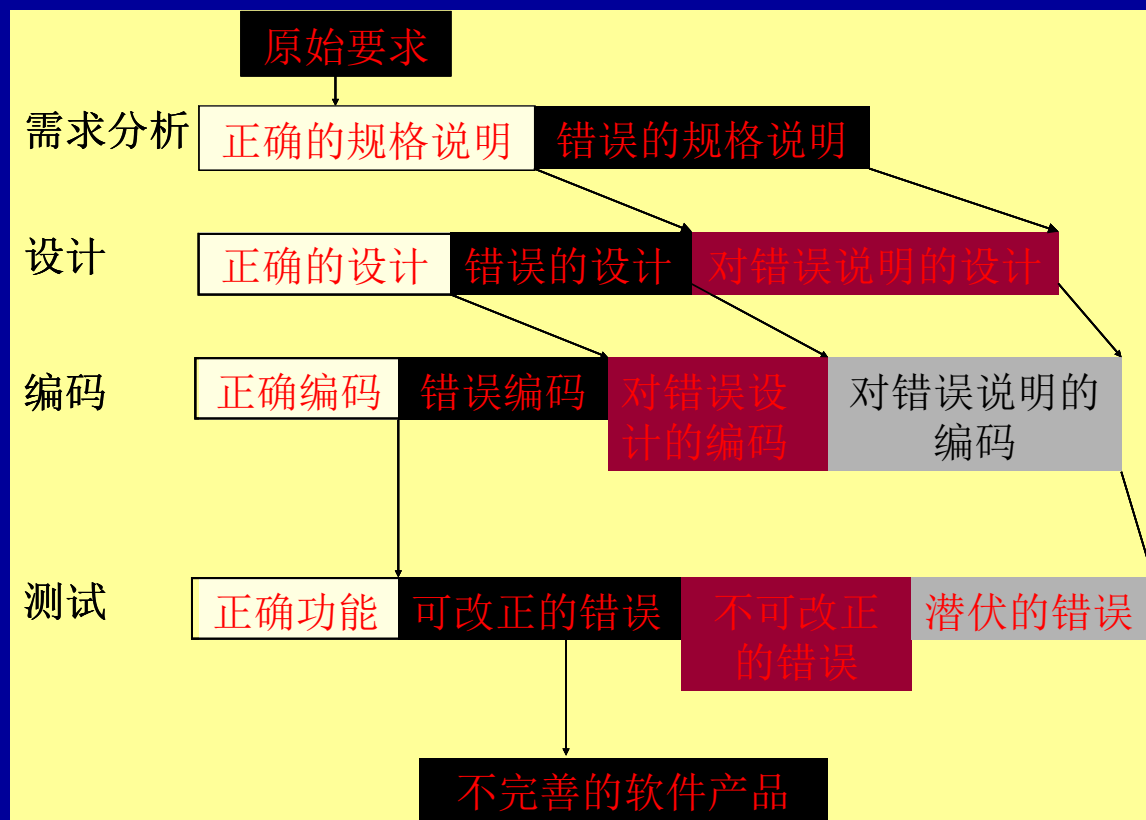
缺陷发现得越晚纠正费用越高，而软件评审的重要目的就是通过软件评审尽早的产品中的缺陷，减少大量的后期返工。



9.1 为什么需要评审

从技术上来衡量

前一阶段的错误自然会导致后一阶段的工作结果中有相应的错误，而且错误会逐渐累积，越来越多。



九、软件评审

9.1 为什么需要评审

9.2 软件评审的角色和职能

9.3 评审的内容

9.4 评审的方法和技术

9.5 准备评审会议

9.6 召开评审会议

9.7 跟踪和分析评审结果

9.8 如何实施成功的评审

9.2 软件评审的角色和职能

- 协调人
- 作者
- 评审员
- 用户代表
- 质量保证代表

九、软件评审

9.1 为什么需要评审

9.2 软件评审的角色和职能

9.3 评审的内容

9.4 评审的方法和技术

9.5 准备评审会议

9.6 召开评审会议

9.7 跟踪和分析评审结果

9.8 如何实施成功的评审

9.3 评审的内容

9.3.1 管理评审

9.3.2 技术评审

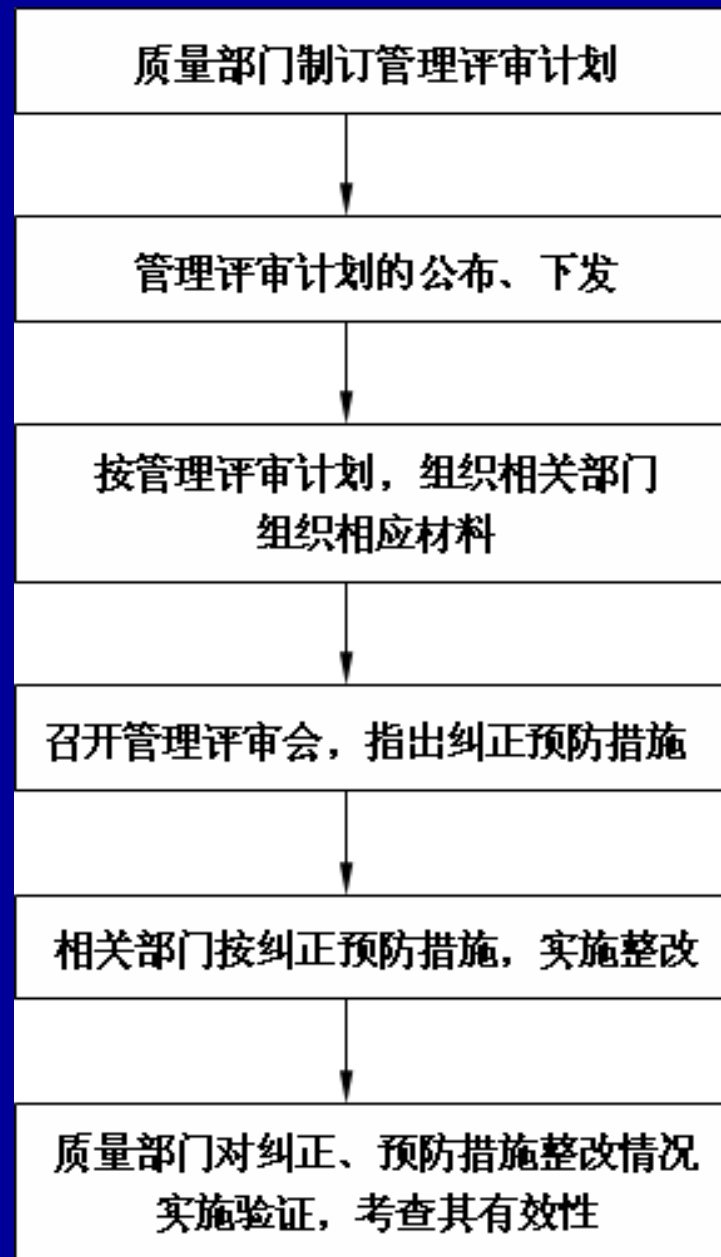
9.3.3 文档评审

9.3.4 过程评审

9.3 评审的内容

9.3.1 管理评审

“由最高管理者就质量方针和目标，对质量体系的现状和适应性进行正式评价。”



9.3 评审的内容

- 质量管理体系运行状况
- 内、外部审核结果
- 改进、预防和纠正措施的状况
- 上次管理评审提出的改进措施实施情况及验证信息

输入

管理评审

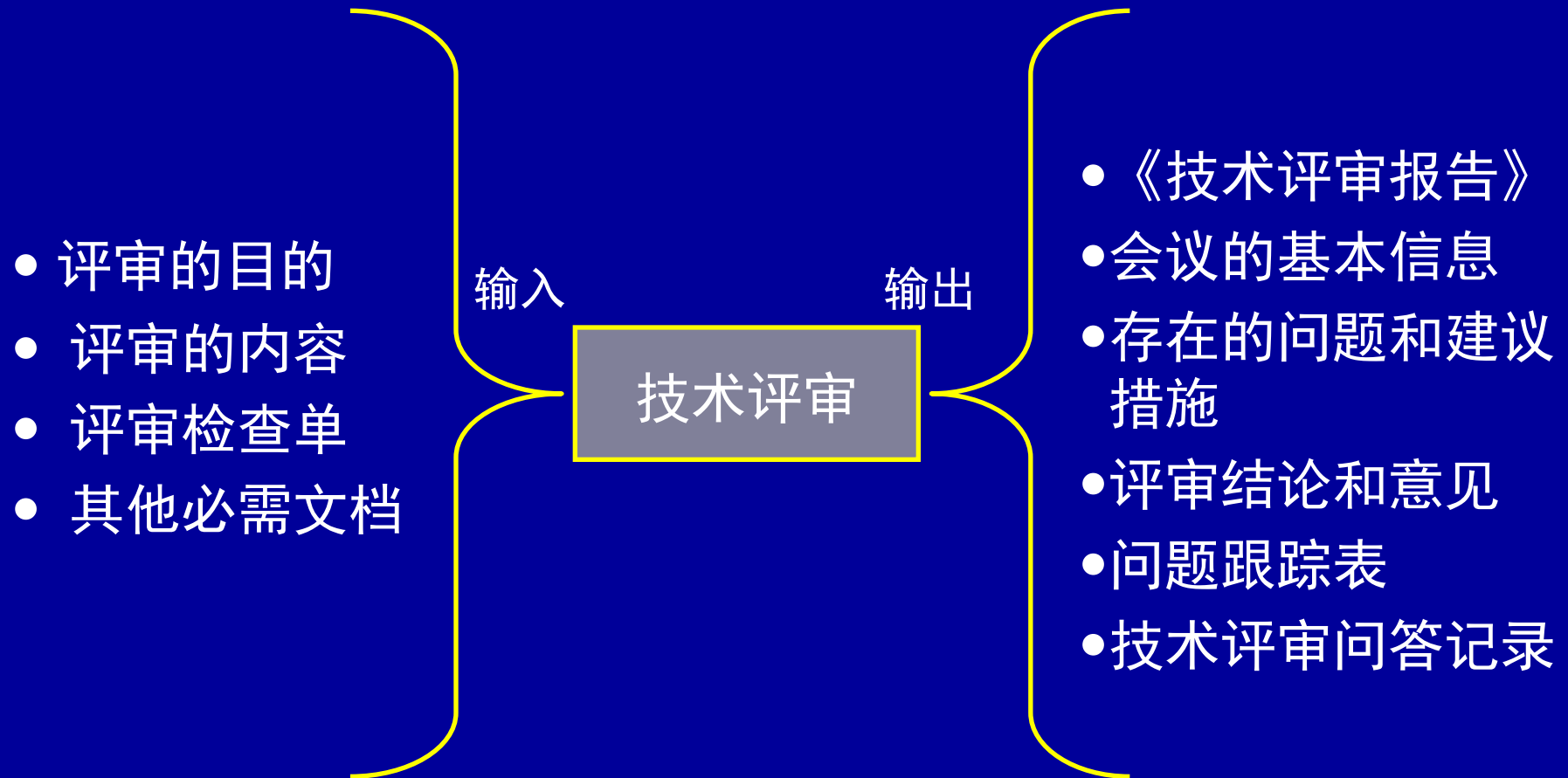
输出

- 质量体系的总体评价
- 质量管理体系及其过程的改进
- 产品是否符合要求的评价，有关产品的改进
- 新资源的需求的决定和措施

对质量体系进行回顾和总结并确保其适宜性、有效性和充分性

9.3 评审的内容

9.3.2 技术评审



9.3 评审的内容

9.3.3 文档评审

1. 正确性
2. 完整性
3. 一致性
4. 有效性
5. 易测性
6. 模块化
7. 清晰性
8. 可行性
9. 可靠性
10. 可追溯性

9.3 评审的内容

9.3.4 过程评审

过程评审的目的：

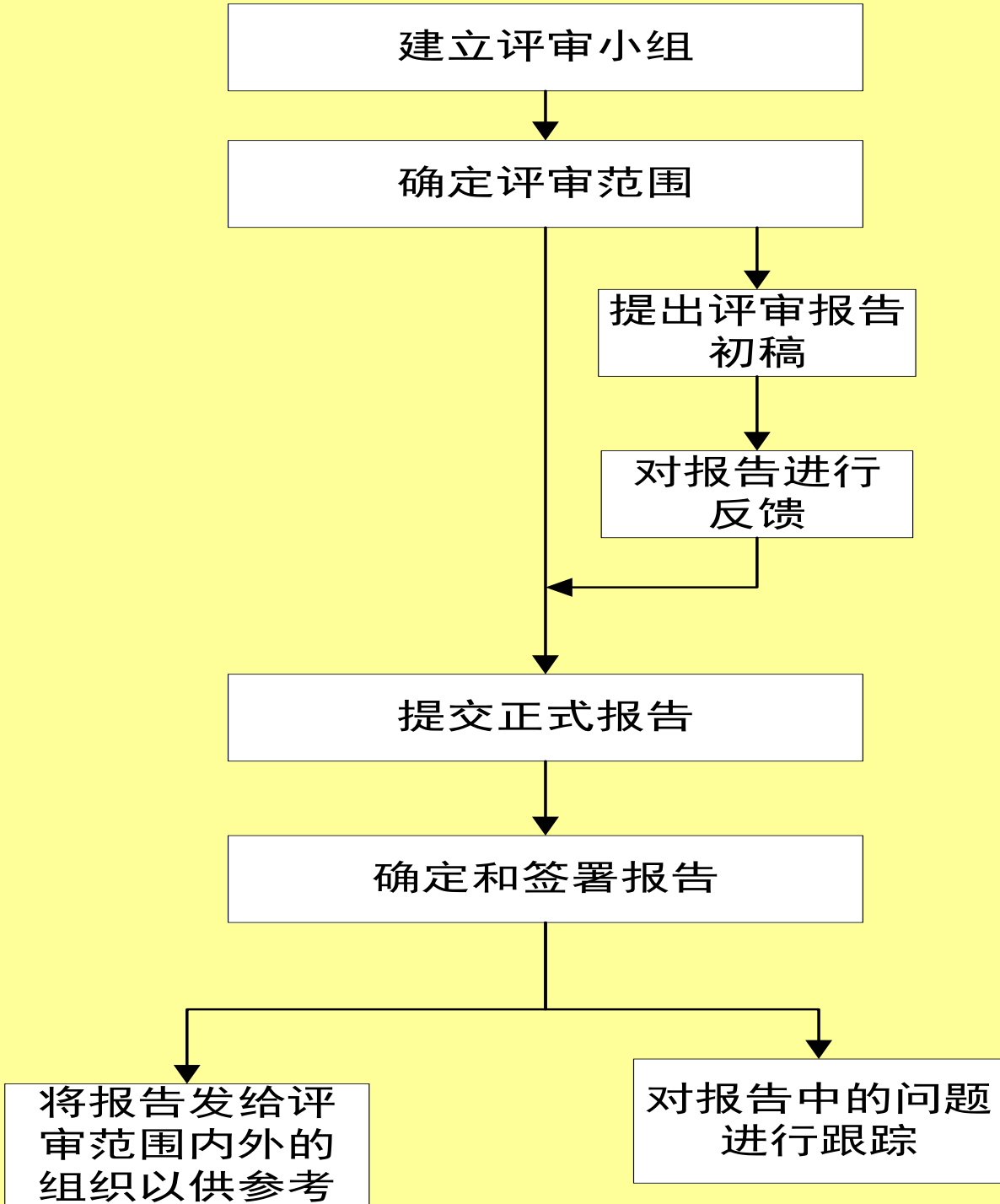
- 评估主要的质量保证流程
- 考虑如何处理/解决评审过程中发现的不符合问题
- 总结和共享好的经验
- 指出需要进一步完善和改进的地方

评审技术后，评审小组需要提交一份《评审报告》，其中包括：

- 评审记录
- 评审后，对现有流程的说明和注释
- 评审小组的建议

9.3 评审的内容

过程评审流程



九、软件评审

9.1 为什么需要评审

9.2 软件评审的角色和职能

9.3 评审的内容

9.4 评审的方法和技术

9.5 准备评审会议

9.6 召开评审会议

9.7 跟踪和分析评审结果

9.8 如何实施成功的评审

9.4 评审的方法和技术

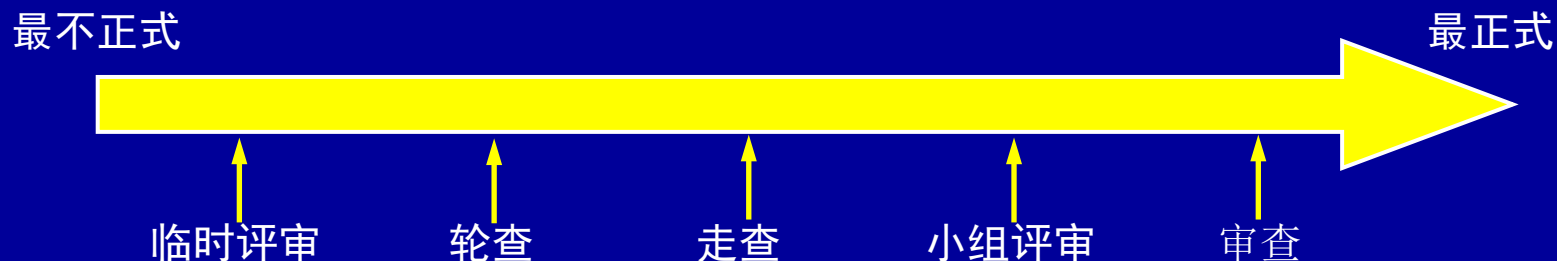
9.4.1 评审的方法

9.4.2 评审的技术

9.4 评审的方法和技术

9.4.1 评审的方法

- 临时评审 (Ad hoc review)
- 轮查 (Pass-round)
- 走查 (Walkthrough)
- 小组评审 (Group Review)
- 审查 (Inspection)



审查、小组评审和走查异同点比较表

角色/职责	审查		小组评审		走查
主持者	评审组长		评审组长或作者		作者
材料陈述者	评审者		评审组长		作者
记录员	是		是		可能
专门的评审角色	是		是		否
检查表	是		是		否
问题跟踪和分析	是		可能		否
产品评估	是		是		否
评审方法	计划	准备	会议	修正	确认
审查	有	有	有	有	有
小组评审	有	有	有	有	有
走查	是	无	有	有	无

如何选择正确的评审方法？

选择评审方法最有效的标准是：

“对于最可能产生风险的工作成果，要采用最正式的评审方法。”

9.4 评审的方法和技术

9.4.2 评审的技术

- 缺陷检查表
 - 它列出了容易出现的典型错误，是评审的一个重要组成部分。
- 规则集
 - 类似于缺陷检查表，通常是业界通用的规范或者企业自定义的各种规则的集合。
- 评审工具的使用
 - 合理的利用工具，如NASA开发的ARM（自动需求度量）
- 从不同角色理解
 - 不同的角色对产品/文档的理解是不一样的。
- 场景
 - 按照用户使用场景对产品/文档进行评审。

九、软件评审

9.1 为什么需要评审

9.2 软件评审的角色和职能

9.3 评审的内容

9.4 评审的方法和技术

9.5 准备评审会议

9.6 召开评审会议

9.7 跟踪和分析评审结果

9.8 如何实施成功的评审

九、软件评审

9.5 准备评审会议

1. 评审计划

各个阶段的《评审计划》的内容包括：各个阶段的评审时间、评审方式、评审组成员等。

SQA在其提交的《质量保证计划》中，应根据各个阶段的《评审计划》，制定相应的评审检查点。

2. 组建评审组

项目组提出评审组长和评审组成员名单的建议，质量组根据项目组的建议，与相关部门或人员（如外项负责人）进行协商确定。

选定评审组长对评审来说是非常重要的，评审组长需要和作者一起，策划和组织整个评审活动。

3. 准备评审材料

基础性和早期的文档，如需求说明和原型等
与重大决策有关的文档，如体系结构模型

对如何做没有把握的部分，如一些挑战性模块，
他们实现了不熟悉的或复杂的算法，或涉及复杂的
商业规则等

将不断被重复使用的部件

4. 发送审查包

将被审查的可交付产品/文档，其中指明了需要审查的部分

定义了可交付产品的前期文档

相关标准或其他参考文档

参与者需要的所有表格

有助于审查者发现缺陷的工具/文档：如缺陷检查表，相关规则等

用于验证可交付产品的测试文档

5. 制定活动进程表

评审会议之前，评审组长还需要制定相应的活动进度表，安排会议房间，并将活动、日期、次数和地点通知评审组成员

九、软件评审

9.1 为什么需要评审

9.2 软件评审的角色和职能

9.3 评审的内容

9.4 评审的方法和技术

9.5 准备评审会议

9.6 召开评审会议

9.7 跟踪和分析评审结果

9.8 如何实施成功的评审

9.6 召开评审会议

评审的主要步骤：

- 1) 由评审员/作者进行演示或说明。
- 2) 评审员会就不清楚或疑惑的地方与作者进行沟通。
- 3) 协调人或记录员在会议过程中完成会议记录。

评审结果：

- 接受，评审内容不存在大的缺陷，可以通过
- 有条件接受，评审内容不存在大的缺陷，修订其中的一些小缺陷后，可以通过
- 不能接受，评审内容中有较多的缺陷，作者需要对这些缺陷进行修改，并在修改之后重新进行评审。
- 评审未完成，由于某种原因，评审未能完成，还需要后续会议

评审中的注意事项：

- 人身攻击 — 在评审过程中，所有的参与人都应该将矛盾集中于评审内容本身，而不能针对特定的参与人。
- 无休止的争论 — 通常对于某些问题，评审组很难达成一致意见，这时，可以把问题记录下来，而如何认定则留给作者自己决定。
- 偏离会议中心 — 在实际会议中，会议常常会发生偏离，如转到政治话题的讨论。
- 鼓励所有人发言 — 鼓励不擅言辞的参与者就评审内容发表自己的看法，比如按照座位顺序轮流发表意见。

九、软件评审

9.1 为什么需要评审

9.2 软件评审的角色和职能

9.3 评审的内容

9.4 评审的方法和技术

9.5 准备评审会议

9.6 召开评审会议

9.7 跟踪和分析评审结果

9.8 如何实施成功的评审

九、软件评审

9.7 跟踪和分析评审结果

1. 评审结果的跟踪

评审结果为有条件接受

评审结果为不接受

2. 评审结果的分析

有效性

效率和成本

九、软件评审

9.1 为什么需要评审

9.2 软件评审的角色和职能

9.3 评审的内容

9.4 评审的方法和技术

9.5 准备评审会议

9.6 召开评审会议

9.7 跟踪和分析评审结果

9.8 如何实施成功的评审

九、软件评审

9.8 如何实施成功的评审

解决不成功评审的主观因素：

- 对所有的工程师进行评审的培训，使评审深入人心
- 预防个人冲突，尽量避免对作者有人身攻击的工程师加入评审小组
- 将评审活动加入到项目计划中，并为评审分配足够的资源
- 收集以前的评审数据，了解哪一种评审方法最为有效
- 将评审列入个人的时间表中，确保评审员有充分的时间为评审做准备和参加评审



9.8 如何实施成功的评审

解决不成功评审的客观因素：

时间/地点	相同	不同
相同	传统的评审方式	异步评审
不同	分布式评审	异步评审

- 异步评审-如共享文档、邮件评审
- 分布式评审-如视频会议

十、软件全面质量管理

10.1 全面质量管理

10.2 全面质量管理在软件开发中实施

10.3 华为公司的TQM关键品质

一切为用户着想、一切以预防为主、
一切凭数据说话和一切按PDCA循环办事；
发轫于质量运动，但非简单的质量管理；
实质是文化变革，关键在建立实施框架。



■ - Sarah Fister Gale

10.1 全面质量管理

◎定义

为了能够在最经济的水平上，并考虑到充分满足用户要求的条件下进行市场研究、设计、生产和服务，把企业内各部门研制质量、维持质量和提高质量的活动构成为一体的一种有效体系。

（费根堡姆）

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。

（ISO9000）

TQM就是全面的、全过程的、全员的和科学的
质量管理的指导思想。

全面质量管理的特点：

- 全员参加的质量管理
- 全过程的质量管理
- 管理对象的全面性
- 管理方法的全面性
- 经济效益的全面性

10.1 全面质量管理

◎ 全面质量管理的基本要求：“三全”。

1. 全过程的质量管理

全过程的质量管理就意味着全面质量管理要“始于识别顾客的需要，终于满足顾客的需要”。

2. 全员的质量管理

产品质量人人有责，人人关心产品质量，人人做好本职工作，全体参加质量管理，才能生产出顾客满意的产品。

3. 全企业的质量管理

全企业的质量管理就是要“以质量为中心，领导重视、组织落实、体系完善。”

◎ 全面质量管理的核心观点

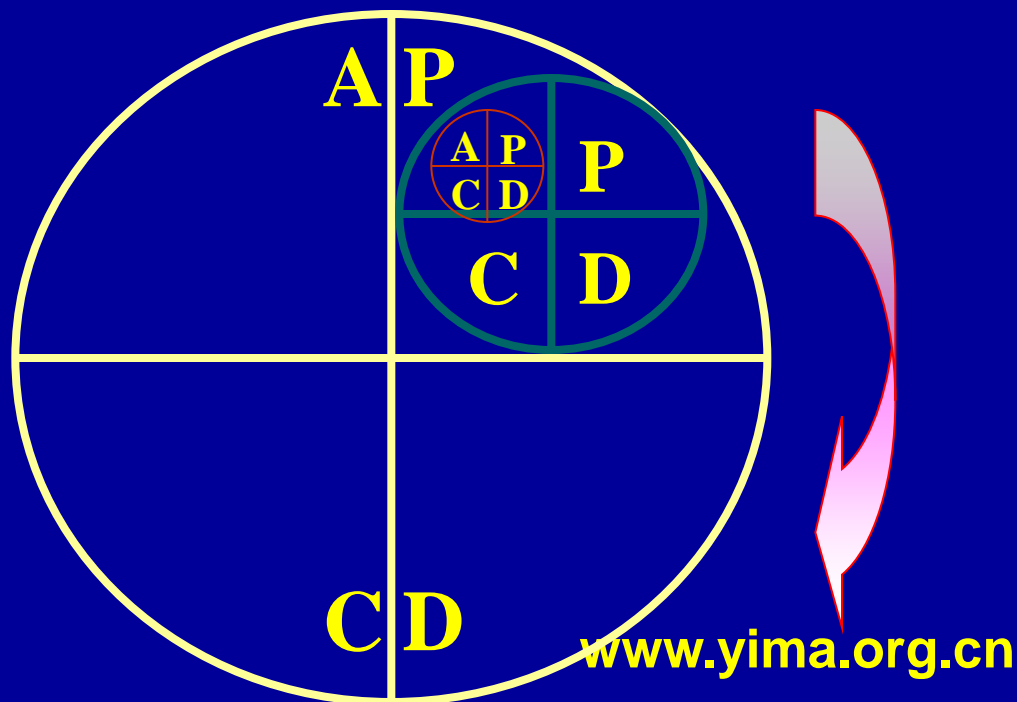
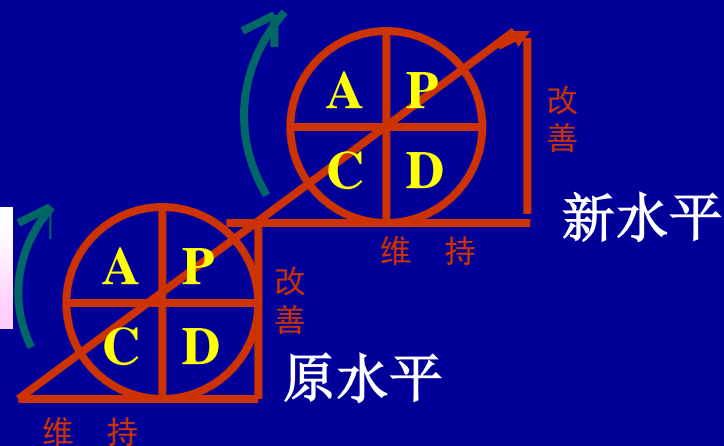
- 一切为了“顾客”
- 一切凭数据说话
- 一切以预防为主
- 一切按PDCA循环办事

◎ PDCA循环的特点

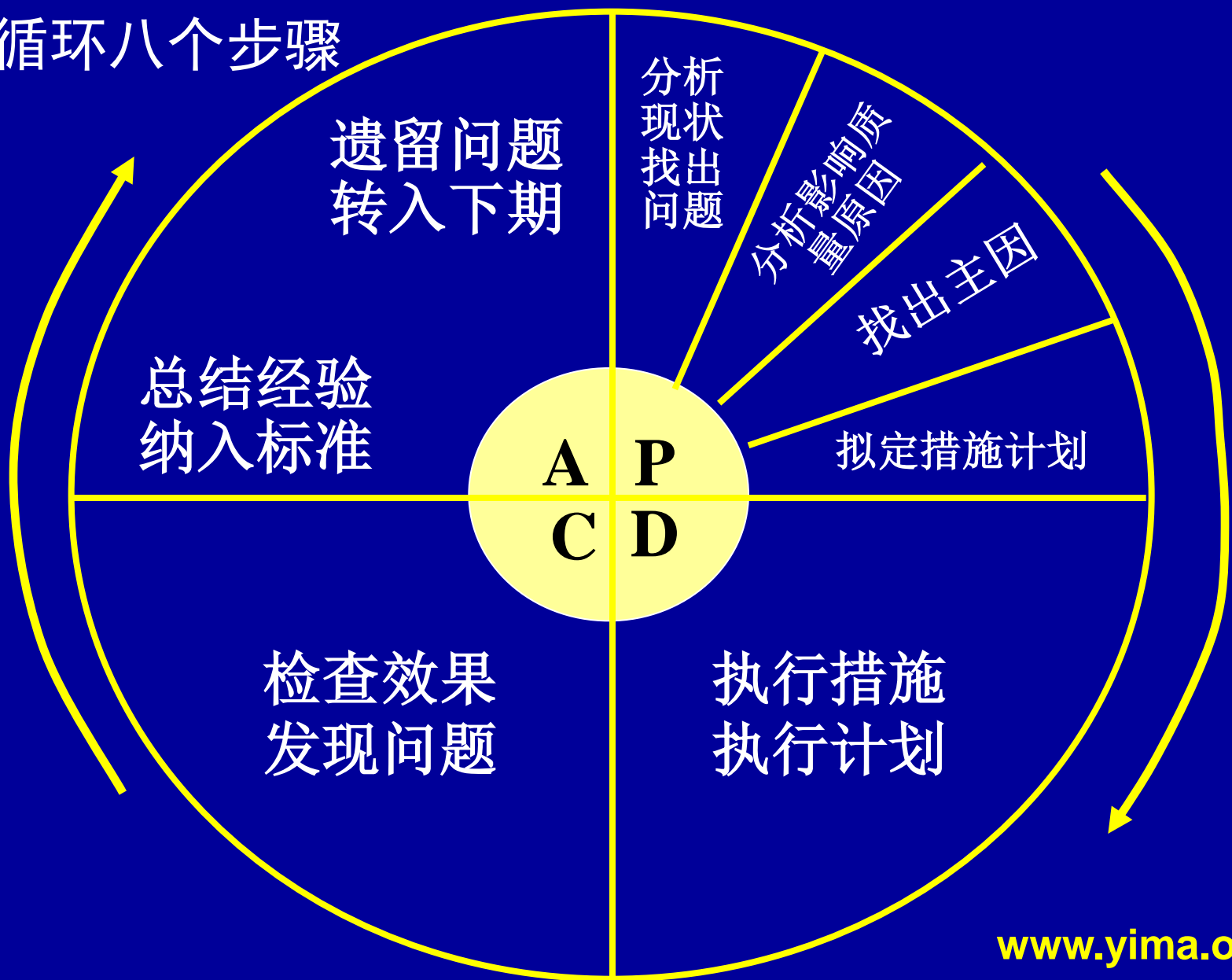
一、完整的循环

二、逐步上升的循环

三、大环套小环



PDCA循环八个步骤



10.2 全面质量管理在软件开发中实施步骤

1. 质量小组的建立和作用
2. 进行全面质量管理思想的教育
3. 了解市场、明确顾客需求
4. 建立明确的质量基准和质量评估制度
5. 建立相对完善的激励机制

10.3 华为公司的TQM关键品质

1. 华为密切留意自己的薄弱环节，公司上下都热衷质量管理。
2. 华为的质量管理方法具有很高的技术性和哲理性。
3. 注重研发、坚持以顾客为导向、拥有精密的技术流程。
4. 令人赞叹的产品可靠性记录。

谢谢

李炳森

QQ: 617205188

Email: basie@yima.org.cn

<http://weibo.com/basielee>

<http://www.yima.org.cn>