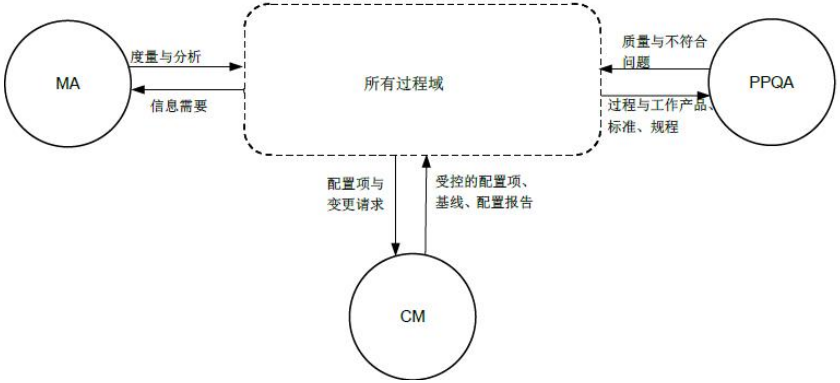
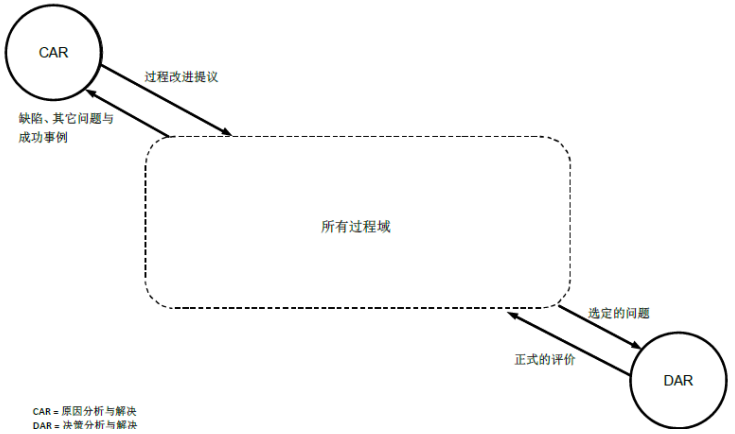
第3章 支持过程

- 3.1 软件质量保证SQA
- 3.2 配置管理CM
- 3.3 度量与分析MA
- 3.4 原因分析与解决CAR
- 3.5 决策分析与解决DAR



基础过程域

应对所有过程域都使用到的基本支持功能，这些过程域都依赖于其它过程域作为输入；



高级过程域

为项目与组织提供改进了的支持能力。这些过程域都依赖于来自其它过程域的具体输入或实践。

3.1 软件质量保证SQA

软件产品是软件企业的生命，高质量、高效率、低成本开发软件产品是企业管理追求的目标；

产品质量是产品的核心竞争力，是现代企业的核心价值观，是企业赖以生存的基石和生命线；

质量是一种态度，质量是一种意识，质量更是一种文化和理念的传递。

3.1 软件质量保证SQA

—— 什么是软件质量

软件质量是软件产品、过程的一组固有特性，反映的是满足客户和其他相关方的要求的程度。

软件质量就是“软件与明确的和隐含定义的需求相一致的程度”。具体地说，软件质量是软件符合明确叙述的功能和性能需求、文档中明确描述的开发标准、以及所有专业开发的软件都应具有的隐含特征的程度。

3.1 软件质量保证SQA

—— 什么是软件质量

●软件需求是进行“质量”度量的基础。与需求不符就是质量不高。

●制定的标准定义了一组指导软件开发的准则。如果不能遵守这些准则，就极有可能导致质量不高。

●通常有一组“隐含需求”是不被提及的（如对维护性的需求）。如果软件符合了明确的需求却没有满足隐含需求，软件质量仍然值得怀疑。

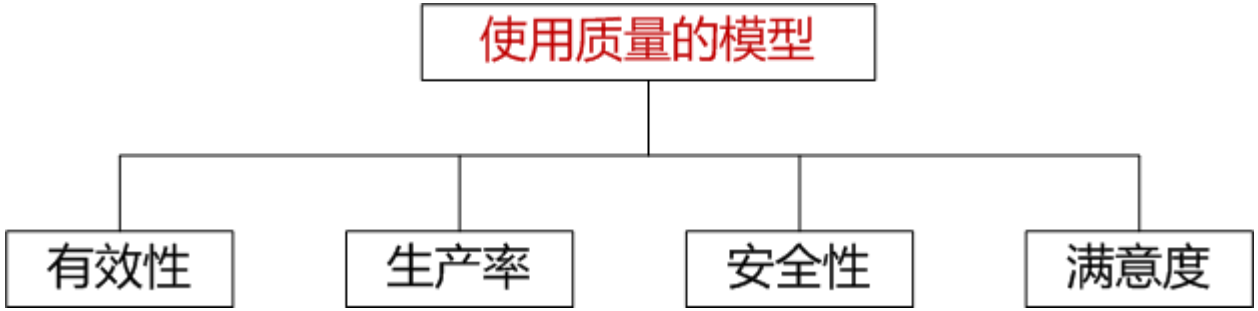
使用质量

使用质量是用户在规定的环境下使用产品实现规定目标的全部特性，包括有效性、生产率、安全性、满意度四个方面。

使用质量是用户对软件及其使用环境下的使用结果的质量观点，不是软件本身的特性。

3.1 软件质量保证SQA

—— 软件质量的层次



有效性	软件产品使用户在规定的使用环境中能正确、完全实现规定目标的能力
生产率	软件产品使用户能在规定的使用环境中实现有效性时消耗适当资源（包括完成的任务数、时间、工作量、资源量）的能力
安全性	软件产品在规定的使用环境中实现在人员、经营、软件、数据、财产或环境损伤方面可接受风险的能力
满意度	软件产品在规定的使用环境中满足用户的能力，包括用户与产品的易理解、易交互、易操作使用等方面

外部质量与内部质量

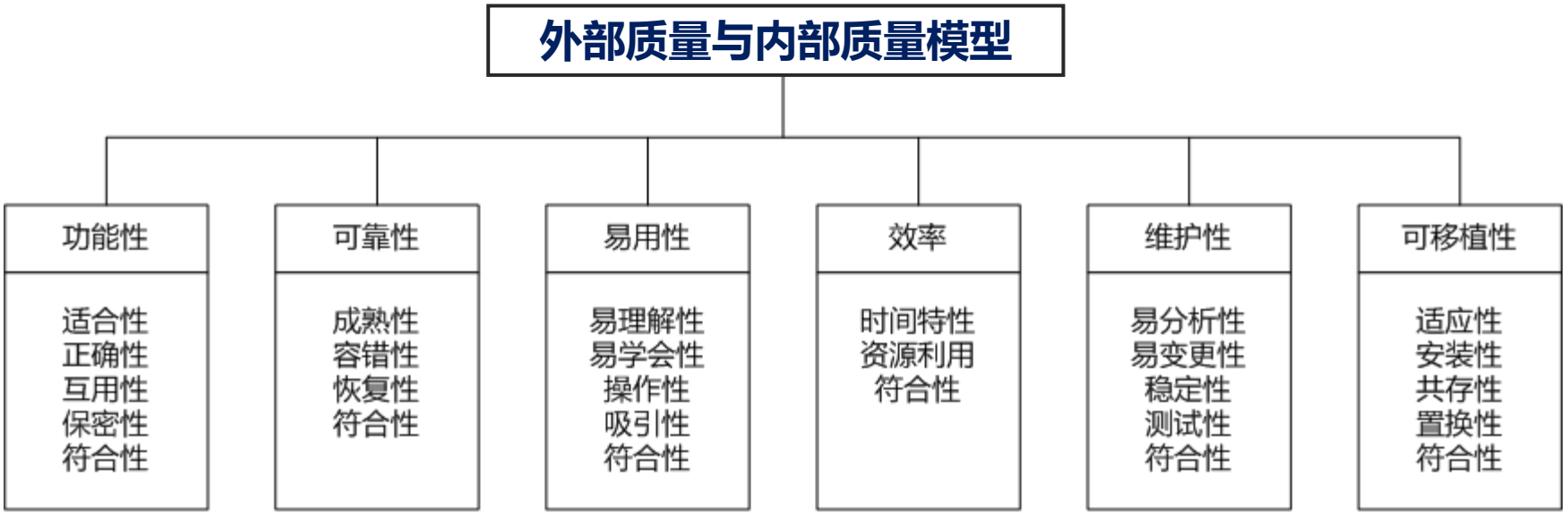
外部质量是从外部观点看软件产品的质量特性，可以简单理解为测试验证组发现的设计或代码缺陷的问题集合，通常是确认测试、系统测试、验收测试、三方测试、验收发现的问题集合。

内部质量是从内部观点看软件产品的质量特性，可以简单理解为项目研发组发现的代码或设计缺陷的问题集合，通常是软件评审、单元测试、软件集成、集成测试发现的问题集合。

外部质量和内部质量采用相同的质量模型。

3.1 软件质量保证SQA

—— 软件质量的层次



3.1 软件质量保证SQA

—— 软件质量的层次

1、功能性	
适应性	与规定任务或用户能否提供一组功能以及这组功能的适合程度有关的软件属性
准确性	与能否得到正确或相符的结果或效果有关的软件属性
互用性	与同其他指定系统进行交互的能力有关的软件属性
保密性	与防止对程序及数据的非授权的、故意的、意外的访问的能力有关的软件属性
符合性	使软件产品遵循与功能性有关的标准、约定、法规及类似规定的软件属性
2、可靠性	
成熟性	与由软件故障引起失效的频度有关的软件属性
容错性	与在误操作、软件故障、违反指定接口的情况下，维持规定的性能水平、失效安全的能力有关的软件属性
恢复性	与在失效发生后，重建其性能水平并恢复直接受影响数据的能力以及为达此目的所需的时间和努力有关的软件属性
符合性	使软件产品遵循与可靠性有关的标准、约定、法规及类似规定的软件属性

3.1 软件质量保证SQA

—— 软件质量的层次

3、易用性	
易理解性	与用户为认识逻辑概念及其应用范围所需努力有关的软件属性
易学会性	与用户为学习软件应用所需努力有关的软件属性
易操作性	与用户为操作和运行控制所需努力有关的软件属性
吸引性	与软件产品吸引用户能力有关的软件属性
符合性	使软件产品遵循与易用性有关的标准、约定、法规及类似规定的软件属性
4、效率	
时间特性	与软件执行其功能时响应和处理处理时间以及吞吐量有关的软件属性
资源特性	与在软件执行其功能时所使用的资源数量及其使用时间有关的软件属性
符合性	使软件产品遵循与效率有关的标准、约定、法规及类似规定的软件属性

3.1 软件质量保证SQA

—— 软件质量的层次

5、维护性	
易分析性	与为诊断缺陷或失效原因及为判定待修改的部分所需努力有关的软件属性
易变更性	与进行修改、排除错误或适应环境变化所需努力有关的软件属性
稳定性	与修改所造成的未预料结果的风险有关的软件属性
易测试性	与确认已修改软件所需努力有关的软件属性
符合性	使软件产品遵循与维护性有关的标准、约定、法规及类似规定的软件属性
6、可移植性	
适应性	与软件无需采用有别于为该软件准备的活动或手段就可能适应不同的规定环境有关的软件属性
易安装性	与在指定环境下安装软件所需努力有段的软件属性
共存性	与其他软件在公共环境种共享公共资源的能力有关的软件属性
置换性	与软件在该软件环境中用来替代指定的其他软件的机会和努力有关的软件属性
符合性	使软件产品遵循与可移植性性有关的标准、约定、法规及类似规定的软件属性

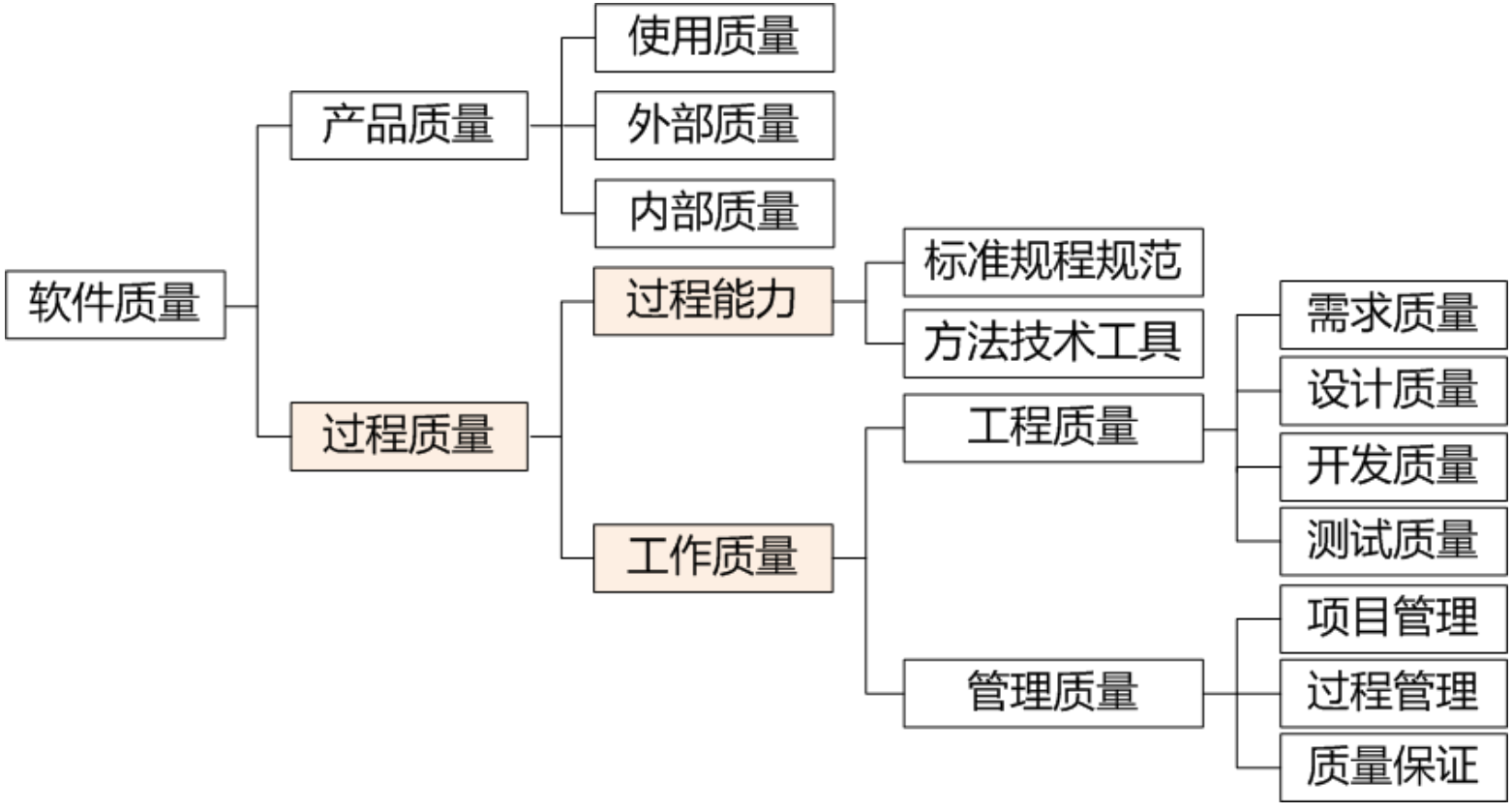
过程质量

内部质量、外部质量、使用质量强调的是产品质量，而产品质量是需要过程来保证的，过程中各项活动的质量决定了产品质量。过程质量是指过程能力与活动满足要求的程度，包括过程能力、工作质量。

3.1 软件质量保证SQA

—— 软件质量的层次

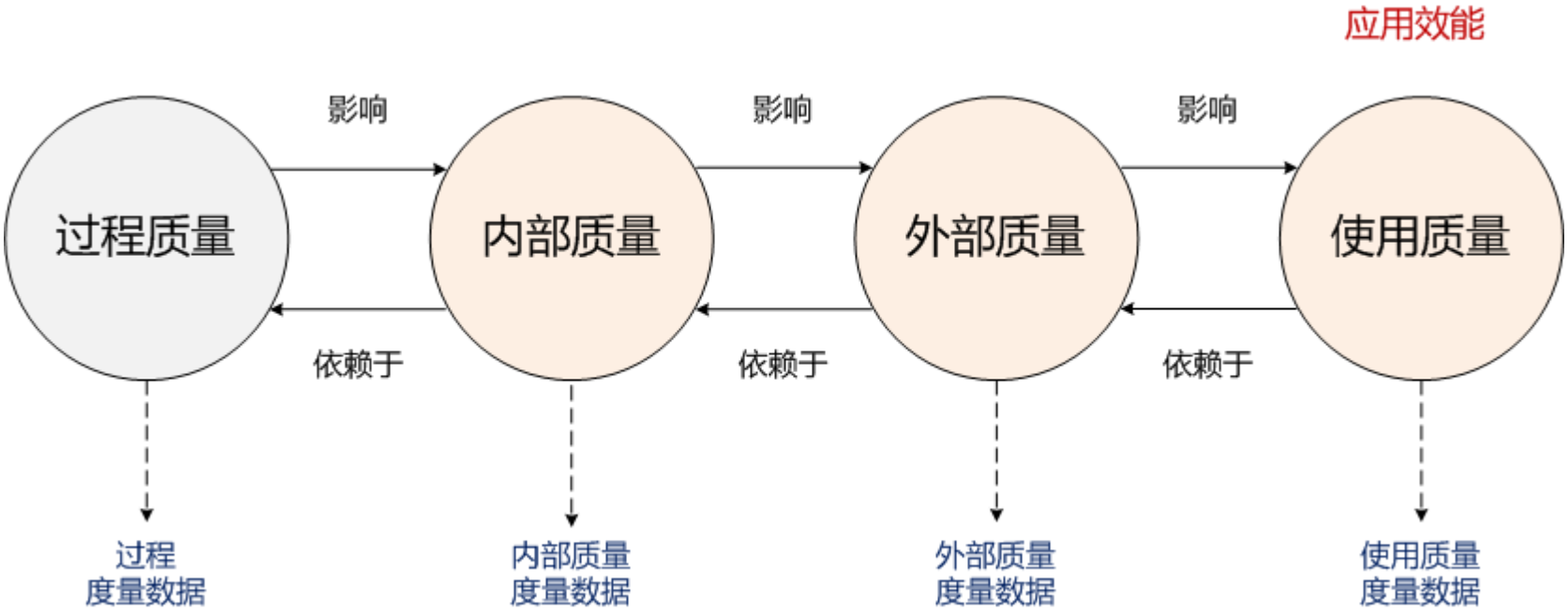
过程质量模型



3.1 软件质量保证SQA

—— 软件质量的层次

软件产品质量的层次



3.1 软件质量保证SQA

软件特性是软件质量的反映，软件特性可以用作评价准则，定量化地度量软件属性可知软件质量的优劣。

常见的软件质量模型：

Jim McCall 软件质量模型（1977 年）

Barry W. Boehm 软件质量模型（1978 年）

FURPS/FURPS+ 软件质量模型

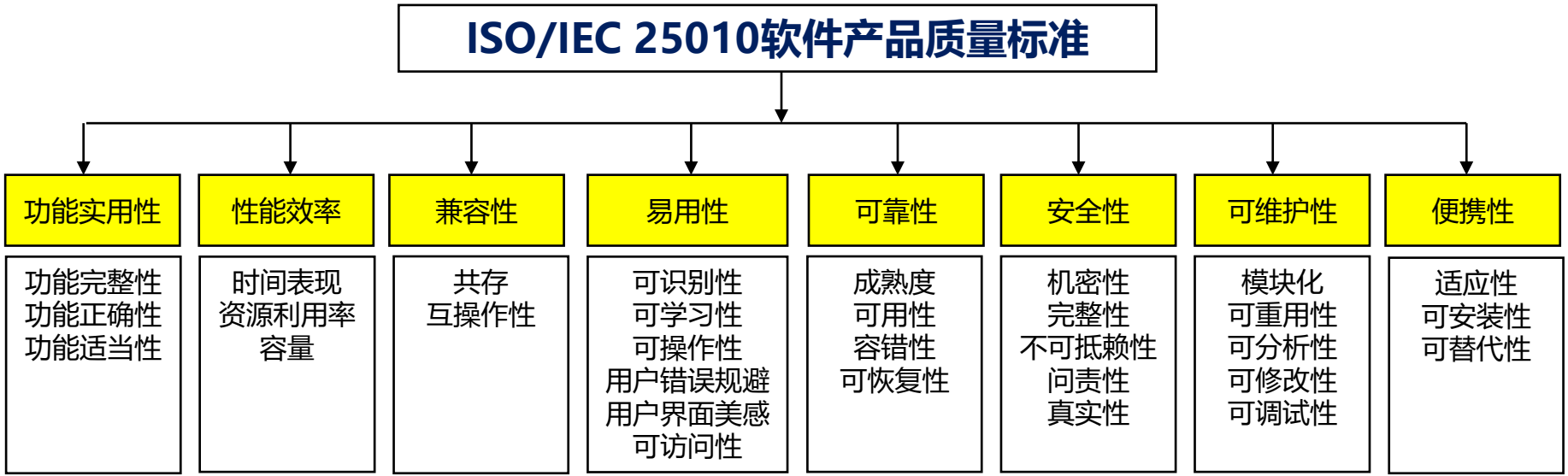
R. Geoff Dromey 软件质量模型

ISO/IEC 9126 软件质量模型（1993 年）

ISO/IEC 25010 软件质量模型（2011 年）

3.1 软件质量保证SQA

—— 软件质量模型



3.1 软件质量保证SQA

—— 软件质量模型

1、功能适用性	
功能完整性	功能涵盖所有指定任务和用户目标的程度
功能正确性	产品和系统满足精确度提供正确结果的程度
功能适当性	功能促进特定任务和目标完成的程度
2、性能效率	
时间表现	产品或系统在执行其功能时的响应和处理时间以及吞吐率达到要求的程度
资源利用率	产品或系统在执行其功能时所用的资源和类型满足要求的程度
容量	产品或系统参数的最大限制达到要求的程度
3、兼容性	
共存	产品使用时同时与其他产品共享公共环境和资源的程度
互操作性	两个或多个产品可以交换并使用信息的程度

4、易用性	
可识别性	用户可以设别产品是否满足其需求的程度
可学习性	特定用户能通过学习达到有效、高效、无风险和满意使用产品的程度
可操作性	产品具有易操作和控制的属性的程度
用户错误规避	系统保护用户避免错误的程度
用户界面美感	用户界面带来愉悦交互的程度
可访问性	满足广泛用户群体使用产品的程度
5、可靠性	
成熟度	系统正常使用下满足可靠性的程度
可用性	系统在需要时可操作和可访问的程度
容错性	系统出现软硬件故障时按预期运行的程度
可恢复性	发生中断或事故时，系统恢复受损数据和重建状态的程度

6、安全性	
机密性	产品确保只有授权人才能访问数据的程度
完整性	产品防止未经授权访问、修改程序和数据的程度
不可抵赖性	可以证明发生某种动作或事件的程度
问责性	可以将实体的行为唯一地追踪到实体的程度
真实性	可以证明主题或资源的身份与所称身份一致的程度
7、可维护性	
模块化	系统为减少耦合和功能影响而分散组件的程度
可重用性	一个资产可用于多个系统的程度
可分析性	评估部件变化对系统影响，或诊断缺陷、故障原因，或确认修改部件有效性和效率的程度
可修改性	产品可有效和高效修改的程度
可测试性	可为产品建立测试标准，并进行测试以确定是否满足标准的有效和效率程度

8、便携性	
适应性	产品能优先和高效适应不同硬件、软件或其他操作和使用环境的程度
可安装性	产品在特点环境中能成功安装和卸载的有效和效率程度
可替代性	在同一环境下，产品可替代另一个相同用途产品的程度

软件项目度量：规模、工作量、成本、进度、风险等；

软件产品度量：功能性、可靠性、易用性、效率、维护性、可移植性等；

软件过程度量：成熟度、生命周期、生产率、缺陷植入率、缺陷消除率、过程管理等。

质量指标分解法度量

特性	子特性	推荐的外部度量清单			
功能性	适合性	功能恰当性	功能实现完备性	功能实现覆盖率	功能实现的稳定性
	正确性	业务流程正确性	业务功能正确性	计算及精度正确性	表达表示正确性
	互用性	数据格式符合预期	交互规程符合预期	交互成功率	
	保密性	访问审核性	访问可控性	数据损坏频率	
	符合性	国标符合性	行标符合性	约定符合性	
可靠性	成熟性	测试用例失效密度	测试覆盖率	测试缺陷密度	测试缺陷消除率
		故障发生密度	故障发生排除率	平均失效间隔时间	
	容错	崩溃避免能力	失效避免能力	误操作避免能力	回退能力
	恢复性	恢复的可用性	恢复的有效性	平均恢复时间	平均停机时间
	符合性	国标符合性	行标符合性	约定符合性	

3.1 软件质量保证SQA

—— 软件质量的度量

特性	子特性	推荐的外部度量清单			
易用性	易理解性	直观性	所见即所得	术语描述规范性	业务流程清晰
		功能明显	界面操作切换	界面嵌套深度	
	易学性	帮助文档有效性	在线帮助	平均学会时间	使用帮助的频率
	操作性	手工输入避免能力	必填项的提醒与校验	提示信息容易理解	出错信息提示清楚
		默认值的有效性	输入错误的校验	更新、删除数据的提示	关闭页面的检测与提示
		操作规程清晰便捷	功能规范简约	个性化定制能力	
	吸引性	界面友好、吸眼球	界面可定制	文化、环境适应能力	
	符合性	国标符合性	行标符合性	约定符合性	
效率	I/O设备资源利用	I/O设备利用率	平均I/O访问效率	平均I/O等待时间	
	存储资源利用	存储资源使用量	平均存储资源访问效率	差错率	
	传输资源利用	传输数据量	传输效率	差错率	
	符合性	国标符合性	行标符合性	约定符合性	

3.1 软件质量保证SQA

—— 软件质量的度量

特性	子特性	推荐的外部度量清单			
维护性	易分析性	状态监视能力	故障诊断排除能力	失效分析排除能力	崩溃排除能力
	易变更性	参数化修改能力	修改复杂性	变更周期、效率	变更控制能力
	稳定性	变更成功率	变更的影响面		
	测试性	测试效率	缺陷密度	缺陷排除率	
	符合性	国标符合性	行标符合性	约定符合性	
可移植性	适应性	数据结构适应性	环境适应性	界面一致性	功能一致性
	安装性	安装容易程度	安装所需人力、资源		
	共存性	公共资源使用量	公共资源访问时间		
	置换性	业务和功能的符合程度	数据的持续利用能力		
	符合性	国标符合性	行标符合性	约定符合性	

使用分解方法的软件质量度量需要花费大量的时间和精力来进行数据的定义和收集。而实际上在很多情况下，仅仅需要对软件的总体质量有一个概略性的度量。

一种简易的方法就是基于缺陷的质量度量。

3.1 软件质量保证SQA

—— 软件质量的度量

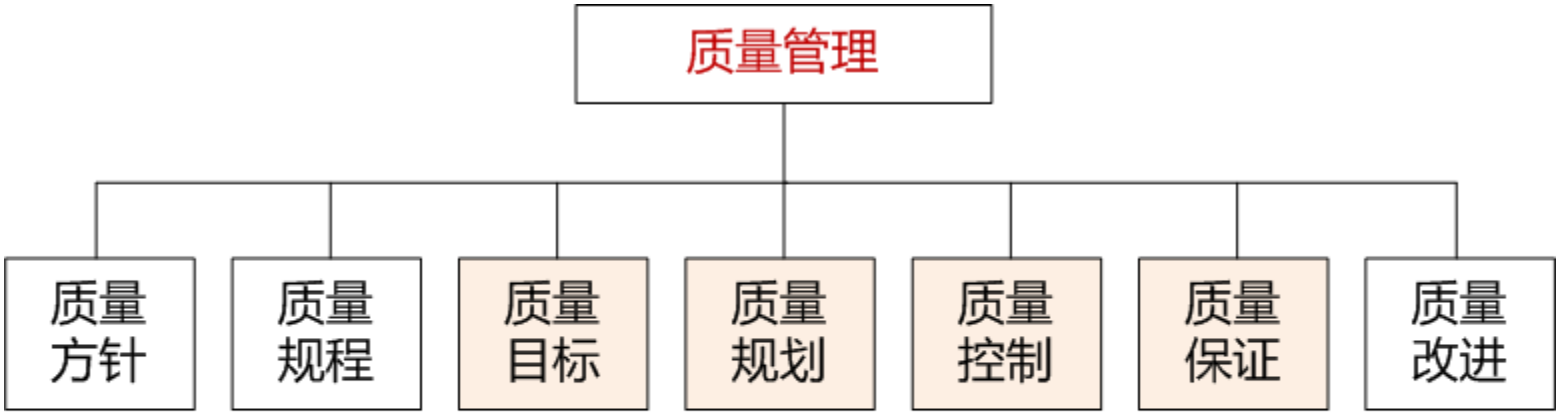
代码质量	(WTP+WF) / KCSI	WTP：运行前，评审测试发现的缺陷数乘权重的累加和 WF：运行后，发现的缺陷数乘权重的累加和 KCSI：新增或修改的代码行数 FLOC：功能点数
产品质量	WF / KCSI	
缺陷植入率	(WTP+WF) / FLOC	
缺陷排除率	WTP / (WTP+WF)	

什么是质量管理

质量管理是指为确保产品或项目质量目标，运用一整套的质量管理体系、手段和方法进行的系统的管理活动，按照组织确定的质量方针、目标和职责，并通过质量体系中的质量策划、质量控制、质量保证和质量改进过程保证最终交付的成果符合质量要求。

通俗的讲：质量管理是为了实现质量目标而进行的所有管理性质的活动。

3.1 软件质量保证SQA



注：

无背景色的，表示企业级工作，通常由EPG组负责

有背景色的，表示产品级/项目级工作，通常由产品经理/项目经理负责

质量管理的内容

质量方针	组织正式发布的质量宗旨、政策、指南和方向
质量规程	组织关于质量管理的标准规程规范、方法技术工具、质量度量要求
质量目标	产品或项目在质量方面的目标要求
质量规划	产品或项目的质量管理过程、质量管理规划、质量度量指标
质量控制	为达到产品或项目质量目标而采取的技术措施和管理措施方面的具体行动
质量保证	为**证实**达到产品或项目质量目标而按计划开展的活动，提供置信度，类似监理
质量改进	致力于收集总结最佳实践、提升组织的质量管理水平、提升产品或项目的质量

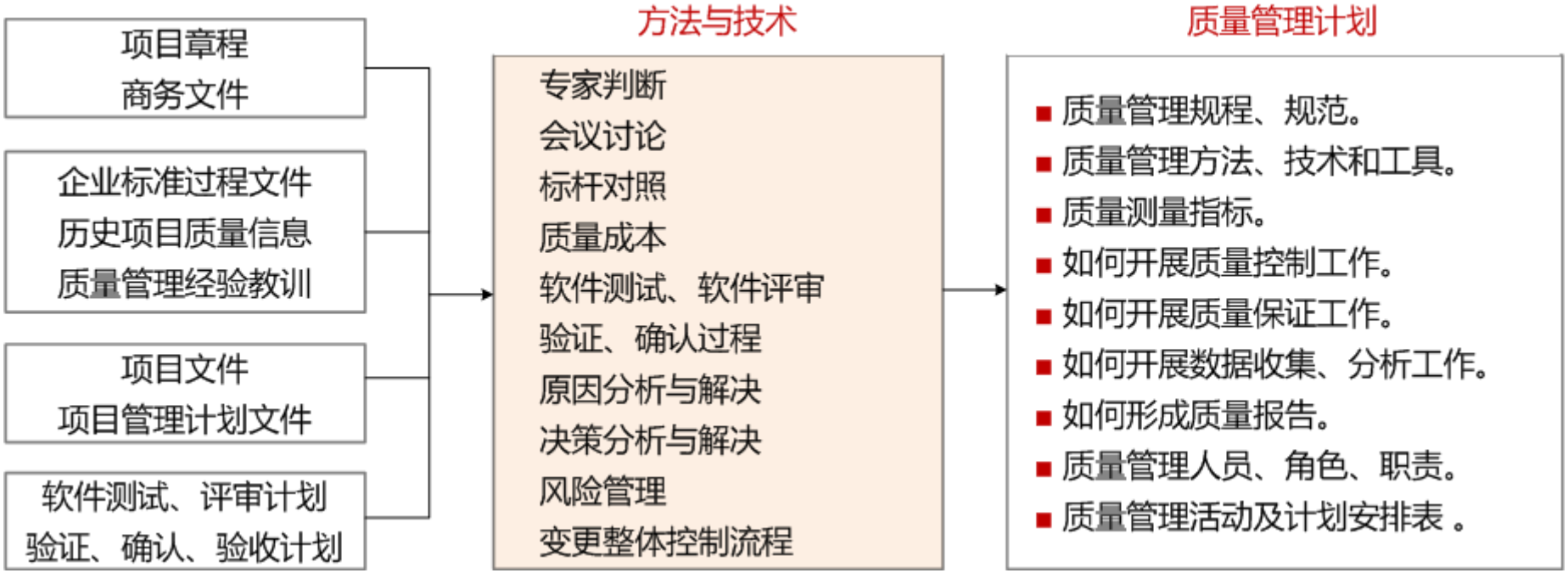
全面质量管理（TQM）

- 质量管理以顾客满意为根本目的，每个产品或项目应有明确的质量目标。
- 质量管理是体系化的、有策划、有计划的，不是想到哪做到哪。
- 质量责任是全员的，不仅仅是QA的，按照全员参与模式开展质量管理控制活动
- 质量活动和质量管理活动必须贯穿项目启动、计划、实施、控制、收尾的全过程。
- 强调对项目的所有活动、所有产出物的全要素质量管理。
- 质量是干出来的，不是检验出来的。
- 质量不是靠技术堆出来的。
- 依据数据和事实，质量应是可置信的，不是随便下结论的。

什么是质量规划

质量规划是确定产品或项目及其交付成果的质量要求、确立项目的质量管理目标、定义质量度量及指标，质量保证、质量控制、质量报告等工作而制定程序规范和计划安排的过程，为如何在整个项目期间管理项目质量提供指南和方向。

质量规划的过程


项目章程
商务文件
企业标准过程文件
历史项目质量信息
质量管理经验教训
项目文件
项目管理计划文件
软件测试、评审计划
验证、确认、验收计划
方法与技术
专家判断
会议讨论
标杆对照
质量成本
软件测试、软件评审
验证、确认过程
原因分析与解决
决策分析与解决
风险管理
变更整体控制流程
质量管理计划
质量管理规程、规范。
质量管理方法、技术和工具。
质量测量指标。
如何开展质量控制工作。
如何开展质量保证工作。
如何开展数据收集、分析工作。
如何形成质量报告。
质量管理人员、角色、职责。
质量管理活动及计划安排表 。

什么是质量控制?

质量控制是为使产品或项目达到质量要求而采取的技术措施和管理措施方面的活动，目的在于消除质量环上引起不合格或不满意效果的因素、确保产品或项目质量能满足要求，是项目组的全部技术人员和管理人员的职责所在。

质量控制的方法

- 按组织标准规程、规范、模板、方法、技术和工具，开展需求分析、技术解决方案、软件设计、开发集成、测试评审、验证确认等软件工程活动。
- 开展巡查、抽查、里程碑检查，确定是否符合定义标准、质量目标和质量管理计划。
- 通过标杆对照、质量核查引导表等方式，强化阶段产出物、交付成果的符合性审查。
- 正视问题，通过严格的原因分析与解决方案流程，一跟到底。
- 通过项目整体变更控制流程，实现需求、设计的变更管理。
- 对软件评审、软件测试、验证、确认等过程的不符合项，应严肃对待，切忌敷衍了事。
- 收集质量数据，通过偏差分析、绩效审查，评估质量控制绩效，及时采取整改措施。

什么是质量保证?

质量保证是为质量管理和质量控制活动、产品或项目的过程、产出物、交付成果是否符合规定标准、管理计划、质量目标和质量要求，提供恰当和足够的监督，并收集分析相关质量数据、确立产品或项目质量的置信度、定期发布质量报告的过程，也就是类似监理的角色和职责。

质量控制与质量保证的关系

质量控制使每个人都努力满足质量要求、及时纠偏、达到质量目标，而质量保证是监督监理质量管理和质量控制活动，二者是相辅相承的关系，是车之二轮、鸟之二翼。

绝大部分的关键过程域，都需要SQA的监督、审核、评审、客观评价、质量报告等工作，通常企业应有QA组，并下沉到每个产品或项目中。

质量保证方法

- 与项目组一起策划计划，将质量保证计划与项目计划紧密结合。
- 与项目组一起确定项目过程、标准规范、方法技术工具、度量指标。
- 运用文件分析技术，仔细研读相关的项目文件、项目管理文件。
- 充分利用验证与确认、问题分析处理、决策分析处理、风险防控、评审、审核信息。
- 按规程组织评审、审核、审查、测量等质量保证活动，审查软件工程活动、产出物、可交付成果，注重实效。
- 结构化的问题分析与解决技术，即从收集信息、发现并定义问题、批判性思维分析根本原因、确定最佳解决方案、执行解决方案、验证解决方案有效性。

质量保证方法

- 对问题缺陷和不符合项应一跟到底，项目组内不能解决的，应及时逐级上报。
- 质量报告应及时通报给相关干系人。
- 由项目组外部的团队开展审计，如组织内部审计部门、项目管理办公室或组织外部的审计团队负责开展质量审计工作，包括识别差距与不足、优秀实践，帮助组织改进项目过程、提高过程绩效和生产率。
- 识别项目过程的问题和改进机会，总结优秀实践，提升过程能力和过程绩效。
- 收集产品或项目的质量数据，客观地评估，定期发布质量报告。
- 应定期组织对质量保证工作本身的评审，及时吸收改进建议。

质量保证活动

（1）按规程为软件项目制定软件质量保证计划；

（2）按计划开展SQA组的活动；

（3）参与制定项目计划、标准和规程；

（4）评审软件工程活动；

（5）审核指定的软件工作产品；

（6）定期向软件工程组报告其活动结果；

（7）按规程处理评审和审核中发现的不符合项，并文档化；

（8）组织SQA本身的评审，包括活动状态、工作产品和绩效。

（一）制定质量保证计划

质量保证计划的主要内容有：

- QA工作的目的、范围、工作职责和权限
- QA工作在项目中的活动资源（人员、培训、设备和工具等）
- 质量保证工程师在项目组中的各项活动的活动内容和时间表
- 确定项目每周QA例行检查时间；确定质量保证工程师或QA经理独立上报的途径、与项目组的通报方式，确定须进行的检查过程、过程检查的内容及依据标准，确定项目进行质量检查的工作产品
- 确定项目需收集的度量表格；确定检查结果的保存方式

（二）实施SQA活动

质量保证工程师按既定的质量保证计划完成以下工作内容：

- 每周例行审计、检查软件过程活动或工作产品；
- 按计划进行软件过程或工作产品的阶段审计
- 协助项目经理（产品经理）组织并参阅项目（产品）评审会议
- 收集和分析项目（产品）度量数据。

（三）SQA每周例行活动

- 每周根据质量保证计划对项目进行例行检查，形成QA周报，有不符合想时编写不符合项报告提交项目经理、总工、项目组成员等；
- 每周检查完毕填写QA活动独立表，收集QA每周检查总人时、不符合解决总人时、验证总人时等度量数据，及时填写在“度量数据库”中；
- 把QA周报和不符合项报告提交配置管理员纳入配置管理；

（四）QA阶段审计

- 根据项目开发计划在各个生命周期阶段结束前进行QA阶段审计，尤其在基线审计后，里程碑评审前QA阶段审计更加关键，审计不通过则项目无法继续开展。
- 主要审计内容为开发过程中的过程活动及过程产生的相关工作产品，验证其是否符合组织制定的标准、规程和模板要求。
- 及时将QA阶段审计报告和不符合项报告提交项目经理（产品经理）、项目组成员、部门经理等
- 填报“项目度量数据库”并提交配置管理员纳入配置管理

（五）参与项目评审

- 按照质量保证计划参加项目评审会议，客观公正第验证评审会议是否按照组织制定的规程、标准进行，将评审过程及审核结果记录到QA周报，若发现不符合项时，则记录到不符合想报告中。步骤如下：
  - 评审会议前检查评审资料是否符合规程、标准或文档模板要求
  - 参加评审会时检查评审过程是否符合规范，评审是否指定主持人、记录员，
  - 评审会议结束后检查项目经理提交的《评审表》是否填写争取
  - 在制定的复审日期检查《评审表》中的缺陷是否按时解决并验证
  - 填写“项目度量数据库”中评审相关数据
  - 正式评审结束后，每周例行检查时形成关于评审的QA周报。

（六）收集分析度量数据

- 根据质量保证计划定期收集项目度量数据，填写项目度量数据库，保证度量数据的完整性和正确性；
- QA按计划执行每周例行检查、阶段审计、不符合项报告验证等活动后，及时填写项目度量数据库中QA活动度量表。
- QA在评审缺陷验证解决后，填写项目度量数据库中的项目评审度量表
- QA定期或事件驱动分析度量数据，形成文档提交项目经理、部门经理和其他相关人员，以渠道支持决策和采取有效的纠正措施。

（七）不符合项处理

- QA人员在每周例行检查、阶段评审或者参与项目评审过程中发现的不符合项，必须按规定及时处理。
  - 及时记录不符合项，并进行编号
  - 设别不符合项的严重登记
  - 不符合项的分类
  - 不符合项的处理
  - 特殊情况处理

3.1 软件质量保证SQA

——过程与产品质量保证PPQA

质量与过程质量保证PPQA是提供一种有效的组织形式和管理办法，通过客观地检查和监控“过程质量”和“产品质量”，从而实现持续地改进质量。

质量保证是一种有计划、贯彻于整个产品生命周期的质量管理办法。

3.1 软件质量保证SQA

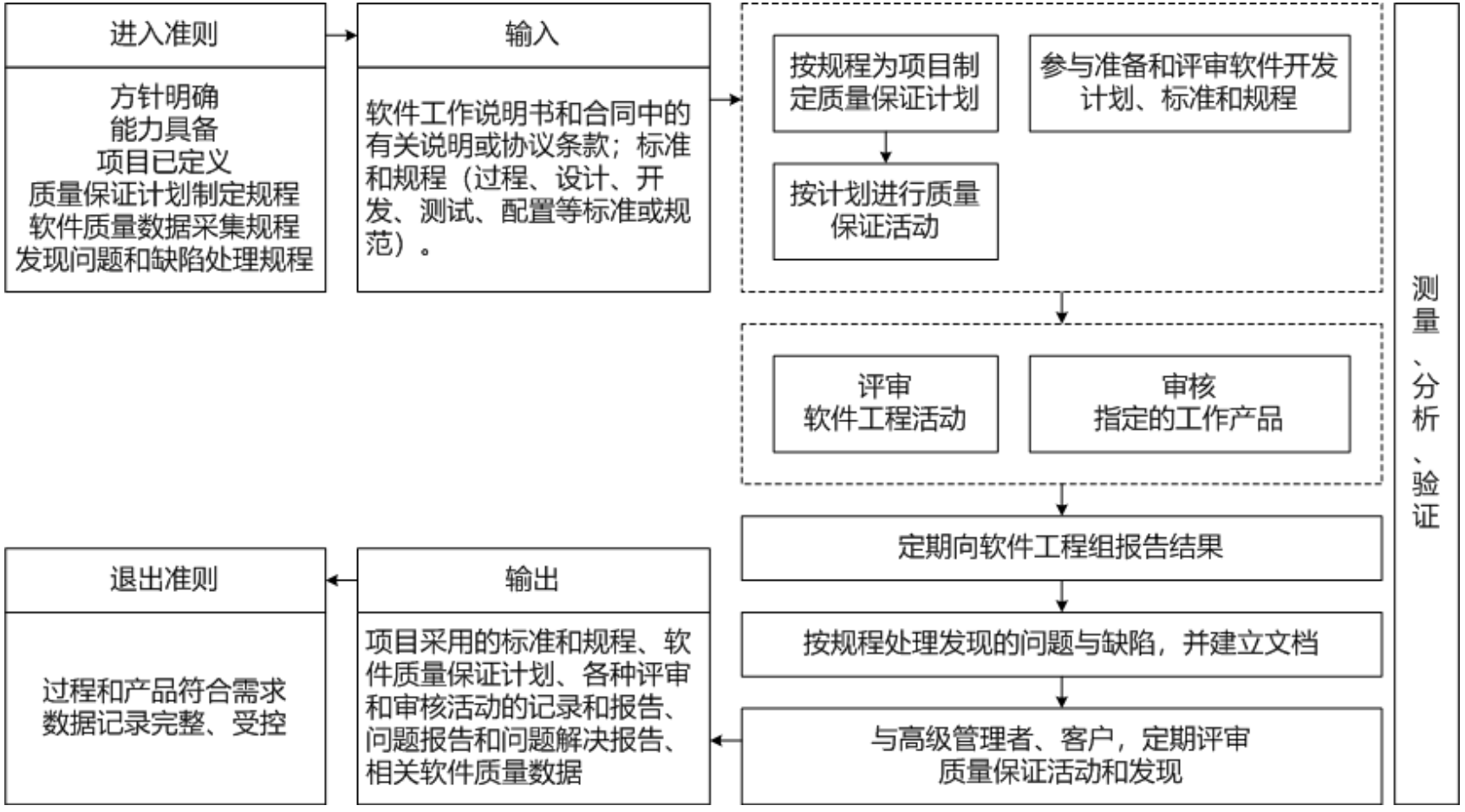
—— 特定目标和特定实践

关键过程域——过程和产品质量保证PPQA

SG1	客观地评价过程和产品	
	SP1.1	客观地评价过程
	SP1.2	客观地评价工作产品和服务
SG2	提供客观深入的了解	
	SP2.1	通报不符合问题并确保解决问题
	SP2.2	建立记录

3.1 软件质量保证SQA

— 特定目标和特定实践



- 3.1 软件质量保证SQA
- 3.2 配置管理CM
- 3.3 度量与分析MA
- 3.4 原因分析与解决CAR
- 3.5 决策分析与解决DAR

3.2 配置管理CM

—— 目的

配置管理CM的目的在于运用配置标识、配置控制、配置状态统计和配置审核，建立和维护项目的整个软件生存周期中工作产品的完整性。

3.2 配置管理CM

—— 特定目标和特定实践

关键过程域——配置管理CM

SG1	建立基线	
	SP1.1	识别配置项
	SP1.2	建立配置管理系统
	SP1.3	建立和发布基线
SG2	跟踪和控制变更	
	SP2.1	跟踪变更需求
	SP2.2	控制配置项
SG3	建立完整性	
	SP3.1	建立配置管理记录
	SP3.2	配置审计与状态报告

3.2 配置管理CM

配置管理的引入

●在开发软件系统的过程中，变化是不可避免的。这些变化使得在同一个项目中工作的软件开发人员之间的彼此不理解难度更加增大。当变化进行前没有经过分析、变化实现前没有被记录、没有向那些需要知道的人报告变化、或变化没有以可以改善质量及减少错误的方式被控制时，大量的不理解问题将会产生；

●协调软件开发以减少由变化带来的不理解性到最小程度的技术称为配置管理。软件配置管理（SCM）是贯穿于整个软件过程中的保护性活动。

3.2 配置管理CM

配置管理的主要内容：

- 制定软件配置管理计划；
- 确定配置项标识规则；
- 实施变更控制；
- 报告配置状态；
- 进行配置审核；
- 进行版本管理和发布管理。

3.2 配置管理CM

IEEE配置管理计划标准

1、引言 配置管理计划的目的、适应范围、使用要求 项目概述 项目中需特别关注的配置管理问题和风险 限制和假设 术语 参考文件 2、软件配置管理 配置管理的组织结构 职责和权限 指令和方针 规程、标准或规范	3、软件配置管理活动 配置管理活动 变更管理和配置控制 配置状态说明 配置审核 接口和子合同方控制 4、软件配置管理进度安排 软件配置管理重要事件的顺序 软件配置管理各项活动之间的依赖关系 进度计划表（甘特图）	5、软件配置管理所需要的资源 采用的工具 使用的设备 所需的知识、技能，以及必要的培训 对其他干系人的要求 6、软件配置管理计划的维护 维护的职责 计划更新的条件和审批 计划变更的交流和通报

3.2 配置管理CM

设别配置项

软件生存周期各个阶段活动的产出物和交付成果，并纳入配置管理的对象称之为软件配置项。

软件配置项的分类、特征和举例

分类	特征	举例
环境类	软件开发环境及维护环境	编译器、操作系统、编辑器、数据库管理系统、设计工具、开发工具、测试工具、项目管理工具
定义类	需求分析及定义阶段得到的工作产品及产出物	需求规格说明书、项目开发计划、设计标准或设计准则、验收测试计划
设计类	设计阶段得到的工作产品及产出物	系统设计规格说明书、数据库设计、用户界面标准、测试标准、系统测试计划、测试用例
编码类	编码及单元测试阶段得到的工作产品及产出物	源代码、目标代码、单元测试数据及单元测试结果
测试类	系统测试完成后的工作产品及产出物	系统测试数据、系统测试结果、操作手册、安装手册
维护类	进入维护阶段以后产生的工作产品及产出物	运行台账、不符合项跟踪表

3.2 配置管理CM

设别配置项

确定配置项就是要确定究竟哪些需要纳入配置管理，成为受控的软件配置项。

软件配置项清单

序号	软件配置项	序号	软件配置项
1	系统规格说明	7	测试规格说明
2	软件项目计划	(1)	测试计划和步骤
3	软件需求规格说明	(2)	测试用例和记录的结果
(1)	图形分析模型	8	操作和安装维护手册
(2)	处理规格说明	9	可执行程序
(3)	原型	(1)	模块可执行代码
(4)	数学规格说明	(2)	链接的模块
4	初步用户手册	10	数据库描述
5	设计规格说明	(1)	模式和数据结构
(1)	数据设计描述	(2)	初始内容（基础参数）
(2)	体系结构设计描述	11	联机用户手册
(3)	模块设计描述	12	维护文档
(4)	接口设计描述	(1)	软件问题报告
(5)	性能设计描述	(2)	维护请求
6	源代码清单	(3)	工程变更指令
		13	软件工程标准和规程

3.2 配置管理CM

为了控制和管理软件配置项，要确立科学、合理、有效的软件配置项区分方法，确保每个配置项必须被独立标识，达到唯一性和可追溯性。

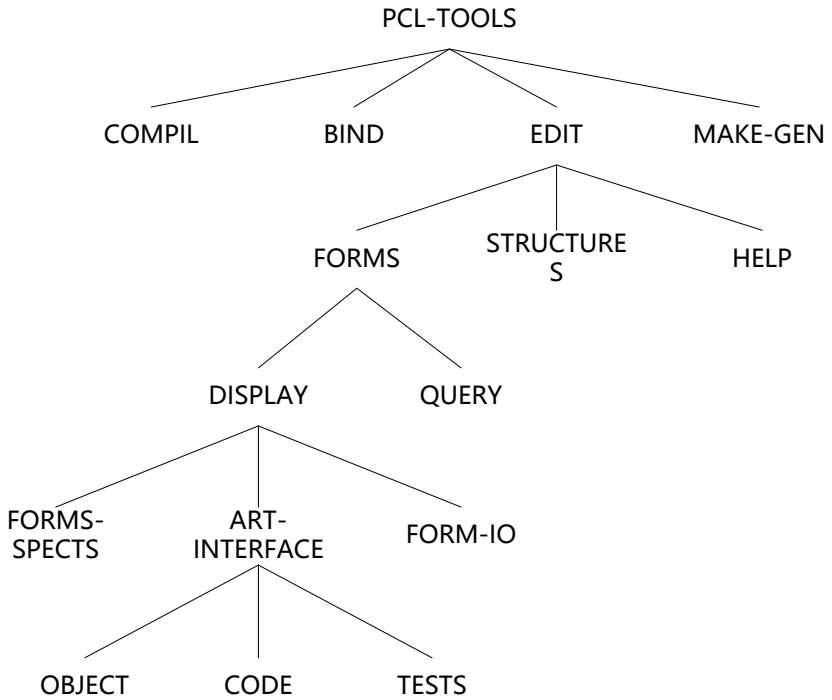
通常采用面向对象的方法组织：

- 名字：是无二义性地标识对象的一个字符串；
- 描述：类型（如文档、程序、数据）、项目标识等；
- 演化：变化和/或版本信息；
- 关系：配置项之间的关系；
- 资源表：处理、更新、引用等。

3.2 配置管理CM

标识配置项

配置项命名通常采用层次式命名规则。



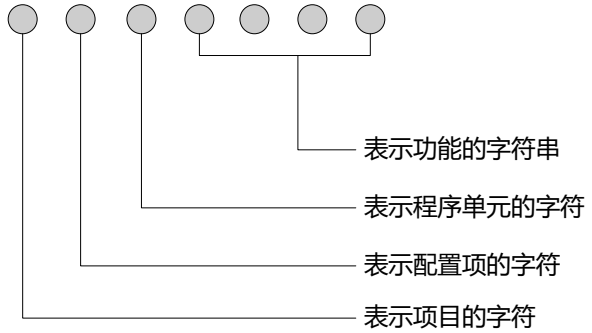
PCL-TOOLS/EDIT/FORMS/DISPLAY/ART-INTERFACE/CODE

3.2 配置管理CM

文档的标识

项目委托组织			文档编号	
项目承担组织			版本号/修订号	
项目名				
文档名				
发布版本			发布日期	
	委托方	日期	承担方	日期
编制				
审核				
批准				
分发信息	日期时间	组织/人员		分数
更新信息	日期时间	更新人	修改页码	版本号

3.2 配置管理CM



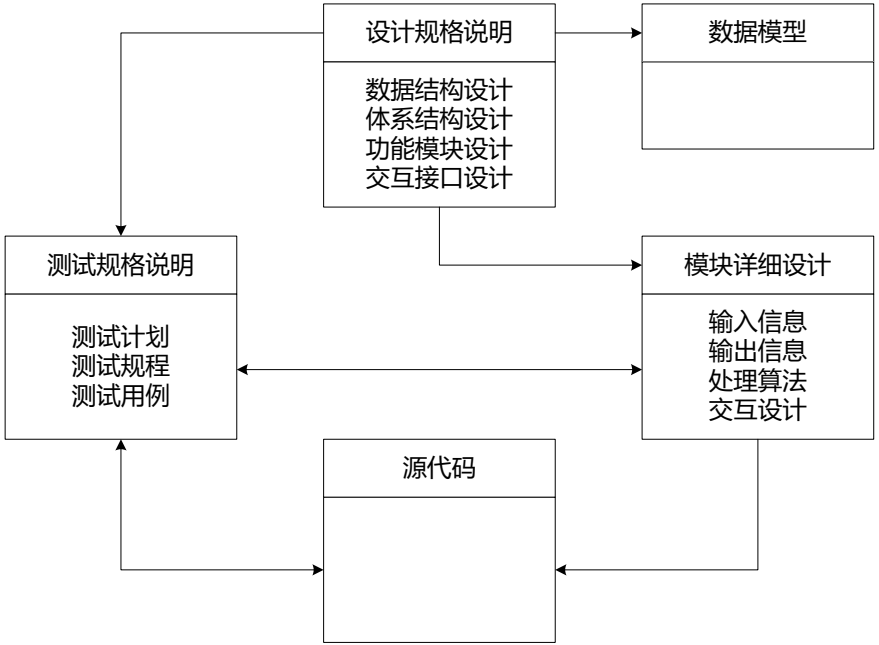
程序单元标识符命名应包括功能模块及层次结构信息。

程序单元的标识

源程序单元首部应包含的标识信息。

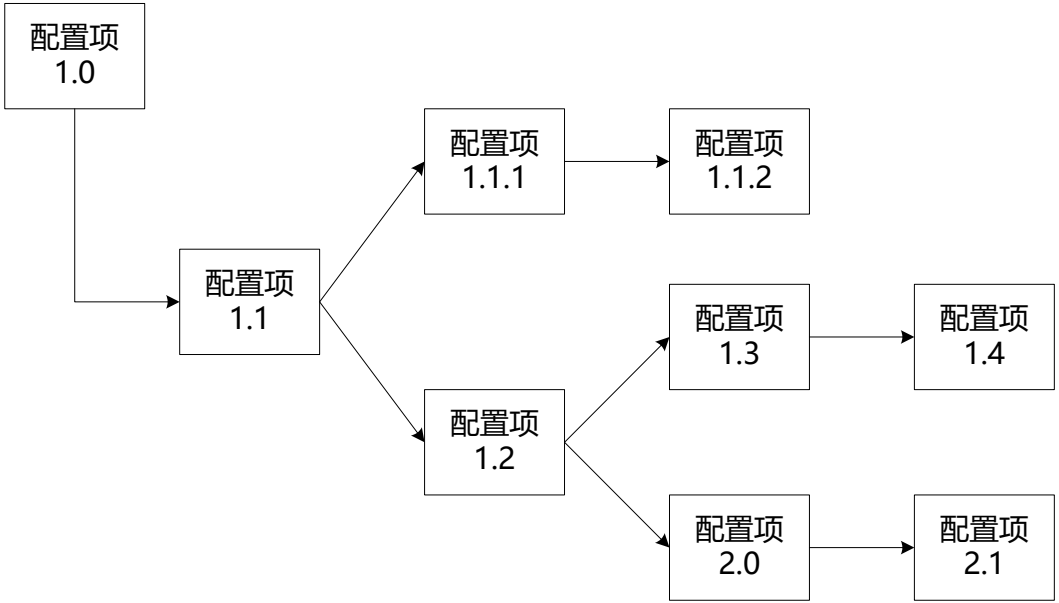
；项目名
；程序单元名及标识符
；功能
；输入信息
；输出信息
；使用语言
；运行环境
；调用者
；被调用者
；版本号
；生成者及日期
；修改者及日期

3.2 配置管理CM



配置项关系的标识可体现配置项之间的相互影响关系

3.2 配置管理CM



配置项演变图可体现配置项的版本演变

3.2 配置管理CM

建立并维护配置管理和变更管理系统：

- 配置管理系统包括存储介质、规程和对配置系统进行存取的工具
- 变更管理系统包括存储介质、规程及对变更请求进行记录与存取的工具
- 确定使用的工具和库结构
- 分配库的存取权限
- 确定管理的流程
- 确定每个配置项的受控级别：不受控、作者控制、CCB控制

3.2 配置管理CM

常用的配置管理工具：

●VSS（Visual Source Safe）：使用简单易学，功能与安全性较弱，且只支持Windows平台；可作为入门工具；

●CVS：开源代码的配置管理工具，安全性和版本管理功能较强；

●SVN：开源代码的版本控制系统，通过采用分支管理系统的高效管理，实现共享资源，实现最终集中式的管理；

●ClearCase：Rational公司的产品，也是目前使用较多的配置管理工具

3.2 配置管理CM

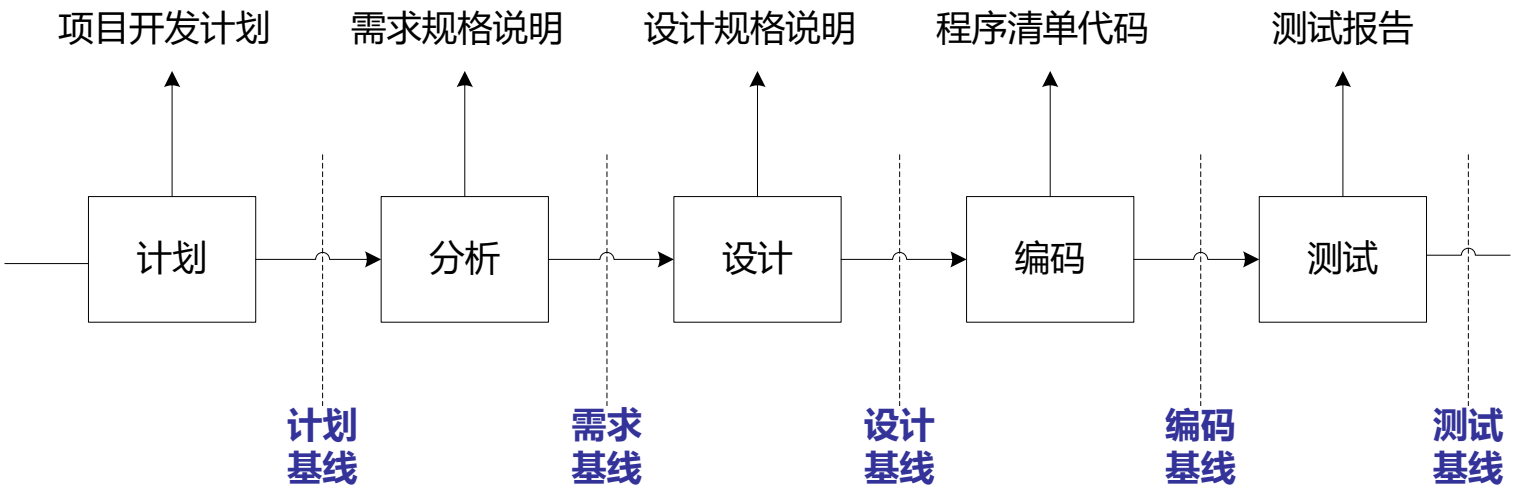
建立和发布基线

基线（baseline）是一个软件配置管理的概念，通常反映一个阶段性工作结束后的、正式发布的产出物的集合。IEEE对基线的定义：已经通过正式复审审核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化/变更控制过程而改变。

建立基线概念是为了把各个开发阶段的工作划分得更加明确，使得本来连续的开发工作在这些点上分割开，从而更加有利于检验和肯定阶段工作成果，有利于变更管理与控制。

3.2 配置管理CM

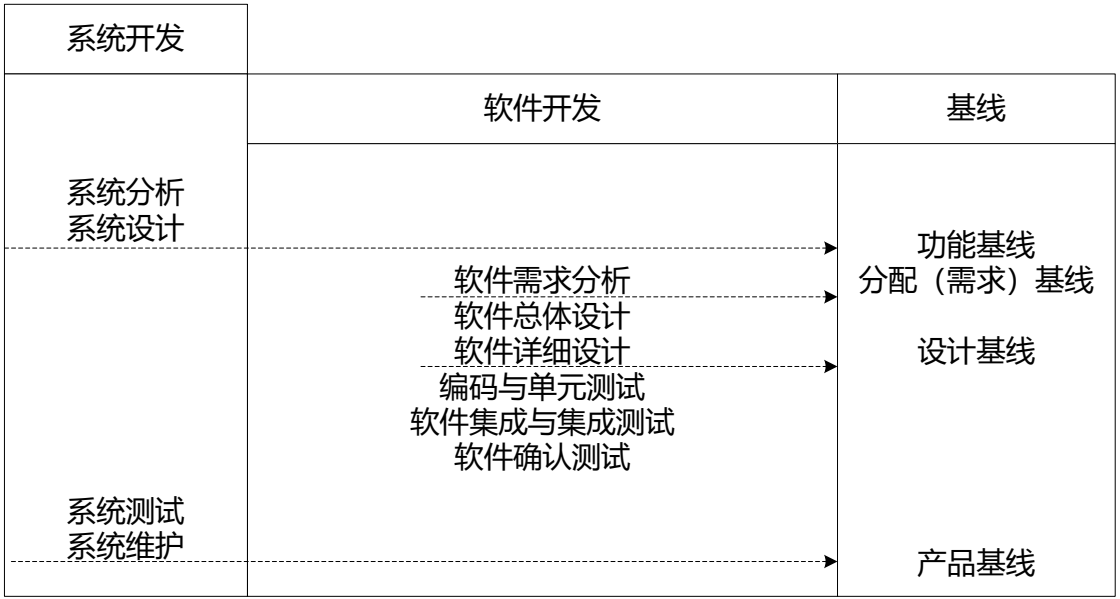
建立和发布基线



基线概念示例

3.2 配置管理CM

建立和发布基线



常用基线

三类配置库：

开发库：存放开发过程中需要保留的各种信息，供开发人员使用。只要开发库的使用者认为有必要，无需对其任何限制；

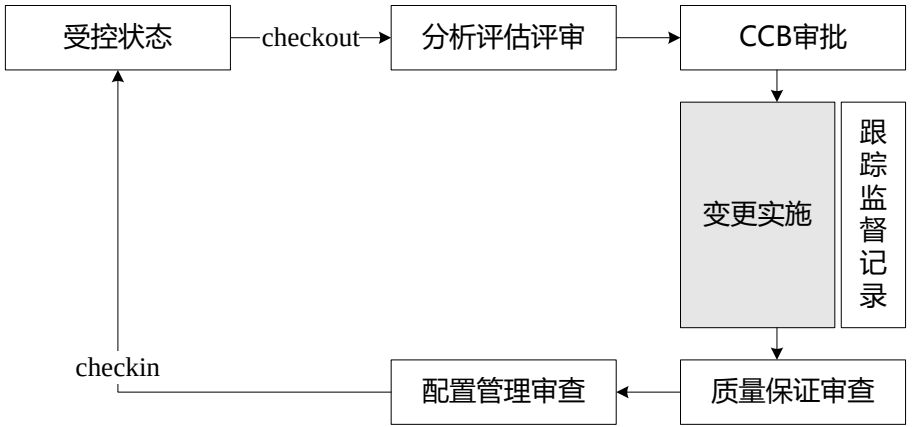
基线库（受控库）：在软件开发的某个阶段工作结束后，将经过审查和批准的阶段工作产品和产出物存放到基线库，需通过正式变更控制才能进行变更；

发布库（产品库）：保存最终的工作产品和产出物，并经正式审查和批准后发布的。

3.2 配置管理CM

对于大型的软件开发项目，无控制的变化将迅速导致混乱，SCM变更控制结合人的规程和自动化工具提供一个变化控制的机制。

变更控制组（change control board，CCB）是配置项变更的监管组织，其任务是对配置项变更申请做出评价、审批以及监督已批准变更的实施。

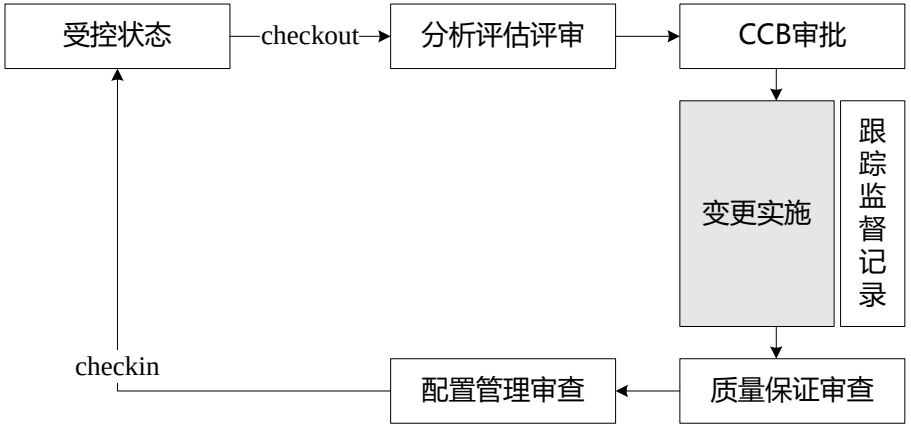


3.2 配置管理CM

配置变更管理

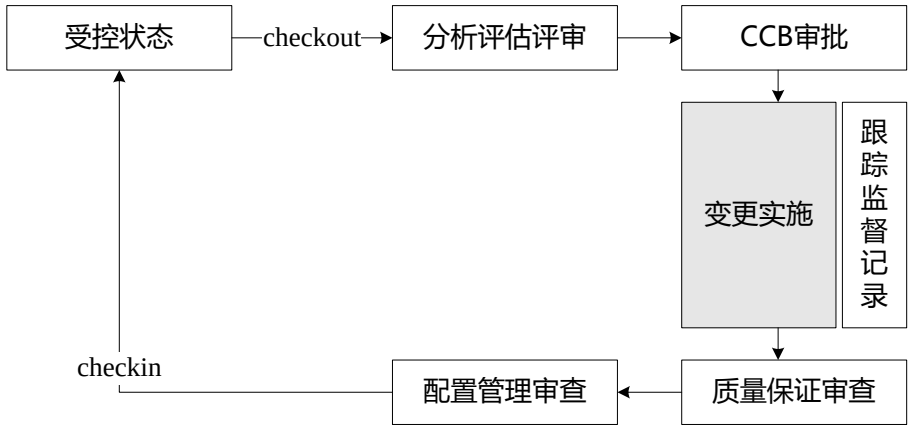
●分析评估评审：一个变化请求被提交和评估，以评价技术指标、潜在副作用、对其他配置对象和系统功能的整体影响、以及对于变化的成本预测；

●CCB审批：评估的结果以变化报告的形式给出，该报告被变更控制组（CCB）使用。对每个被批准的变化生成一个过程变化命令（ECO），ECO描述了将要进行的变化、必须注意的约束、以及复审和审计的标准；



3.2 配置管理CM

●将被修改的对象从项目数据库“提取（check out）”出来，进行修改，并应用于合适的SQA活动，然后，将对象“提交（check in）”进数据库，并使用合适的版本控制机制去建立软件的下一个版本。



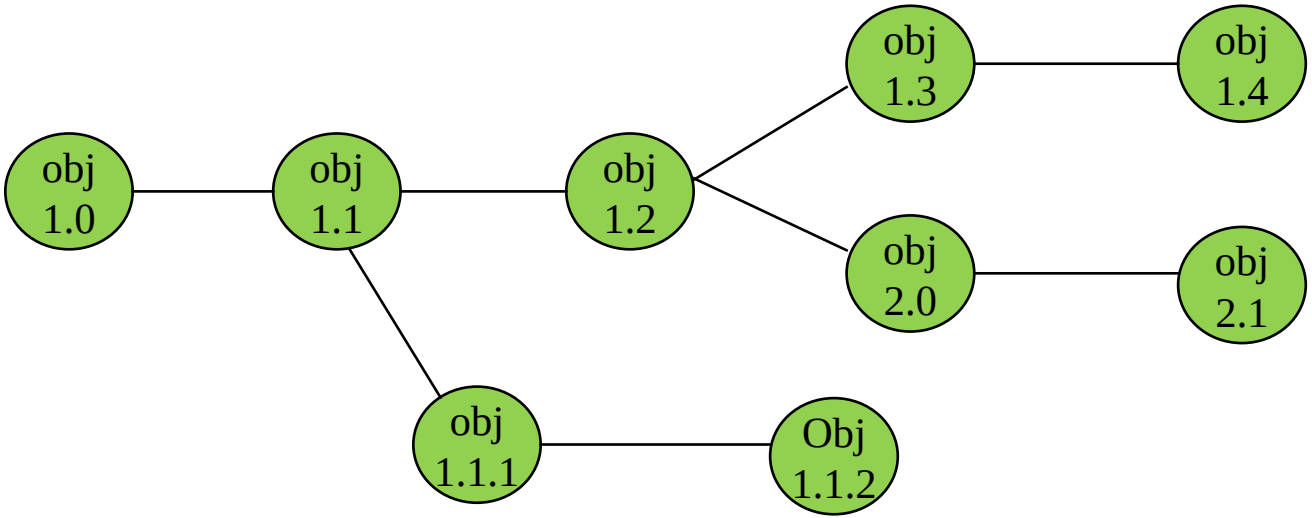
“提取”和“提交”过程实现了两个主要的变化控制要素----访问控制和同步控制：

●访问控制：管理哪个软件工程师有权限去访问和修改某特定的配置对象。

●同步控制：保证由两个不同人员完成的并行修改不会互相覆盖。

3.2 配置管理CM

版本控制是软件配置管理的核心功能。所有置于配置库中的元素都应自动予以版本的标识，并保证版本命名的唯一性。版本在生成过程中，自动依照设定的使用模型自动分支、演进。



3.2 配置管理CM

标识、版本控制和变化控制帮助软件开发者维持秩序，否则情况可能将是混乱和不固定的。然而，即使最成功的控制机制也只能在ECO产生后才可以跟踪变化。

我们如何帮助变化被合适的实现呢?

(1) 正式的技术复审;

(2) 软件配置审计。

正式的技术复审关注已经被修改的配置对象的技术正确性，复审者评估SCI以确定它与其他SCI的一致性、遗漏、及潜在的副作用，正式的复审应该对所有的变化进行。

3.2 配置管理CM

配置审计

配置审计询问并回答如下问题：

- 在ECO中说明的变化已经完成了吗？加入了任意附加的修改吗？
- 是否已经进行了正式的技术复审，以评估技术的正确性？
- 是否适当地遵循了软件工程标准？
- 变化在SCI中被“显著地强调（highlighted）”了吗？是否指出了变化的日期和变化的作者？
- 对象的属性反应了变化吗？
- 是否遵循了标注变化、记录变化并报告变化的SCM规程？
- 所有相关的SCI被适当修改了吗？

配置状态报告（Configuration status reporting, 有时称为status accounting）是一个SCM任务，它回答下列问题：

- 发生了什么事?
- 谁做的此事?
- 此事是什么时候发生的?
- 将影响别的什么吗?

配置状态报告(CSR)的信息流如下：

- 每次当一个SCI被赋上新的和修改后的标识时，则一个CSR条目被创建；
- 每次当一个变化被CCA批准时（即一个ECO产生），一个CSR条目被创建；
- 每次当配置审计进行时，其结果作为CSR任务的一部分被报告。
- CSR的输出可以放置到一个联机数据库中，使得软件开发者或维护者可以通过关键词分类访问变化信息。此外，CSR报告被定期生成，并允许管理者和开发者评估重要的变化。

- 3.1 软件质量保证SQA
- 3.2 配置管理CM
- 3.3 度量与分析MA
- 3.4 原因分析与解决CAR
- 3.5 决策分析与解决DAR

3.3 度量和分析MA

度量和分析的目的在于开发和维持度量能力，提供量化的数据，支持软件过程管理、项目管理的需要。

理解过去：获得对过程、产品、项目等的定量数据，做到“心中有数”

管理现在：基于历史数据建立预测模型，进行估算和计划

预测未来：根据历史的量化信息，找到问题所在，确定潜在改进机会

3.3 度量和分析MA

度量的作用

1	衡量组织的业务目标	度量是基于组织的业务目标或者愿景，并通过量化的信息来表征、衡量业务目标。
2	促进过程能力改进	通过度量、能够清晰的知道问题、不足，并很快掌握问题的根源并找到纠正的方向。
3	提供基准和决策信息	丰富的历史数据库，科学的标准，为项目计划、管理改进、战略决策提供依据。
4	对项目的作用	有助于发现项目的偏差和问题，度量提供一个信息给项目经理，什么情况下需要采取措施，什么情况是属于正常的偏差。
5	对组织的作用	能够为组织来统计类似项目的统计基准提供数据，便于组织能够统计分析中一种平均的能力
6	对客户的作用	提供给客户一种信息，相信我们提供产品或服务的能力是可以衡量和预测的

度量的分类

（1）过程度量：目的是形成组织的各种模型，作为对项目、产品的度量基础；以及对软件开发过程进行持续改进，提高生产力；

（2）项目度量：对软件开发项目的度量，目的是评估项目开发过程的质量、预测项目进度、工作量等，辅助管理者进行质量和项目控制。

（3）产品度量：对项目开发结果即最终的产品度量。通常指的是质量度量。如：软件可靠性度量、软件复杂度度量、软件缺陷度量等。

3.3 度量和分析MA

— 目的

CMMI对度量的要求举例

关键过程域	测量实例
需求管理	每个给定需求的状态； 给定需求的变更活动； 变更给定需求的累计数，包括提出的、未决定的、认可的和你纳入需求基线的总数。
项目策划	与计划相比较，软件项目策划活动的里程碑完成情况； 与计划相比较，在软件项目策划活动种所完成的工作，所用的工作量和所消耗的成本。
项目跟踪与控制	在完成跟踪和监督活动时所花费的工作量、资源量； 对软件开发计划的变更活动，包括对规模估计、成本估计、资源估计、进度安排等。
子合同管理	管理子合同活动的成本与计划相比较； 子合同产品的实际交付日期与计划相比较； 子合同产品的质量与目标要求相比较； 主承包商对分包商的评价。
软件质量保证	软件质量保证活动的里程碑完成情况与计划相比较； 软件质量保证活动所完成的工作、花费的成本和工作量，与计划相比较。
配置管理	配置管理活动的里程碑完成情况与计划相比较； 配置管理活动所完成的工作、花费的成本和工作量，与计划相比较。
组织过程性能	组织的标准过程集中已选定过程的定量了解，过程质量与过程性能与目标相比较，为定量项目管理提供过程性能数据、基线。
定量项目管理	运用组织标准过程的性能数据、基线，对项目进行定量统计分析。

3.3 度量和分析MA

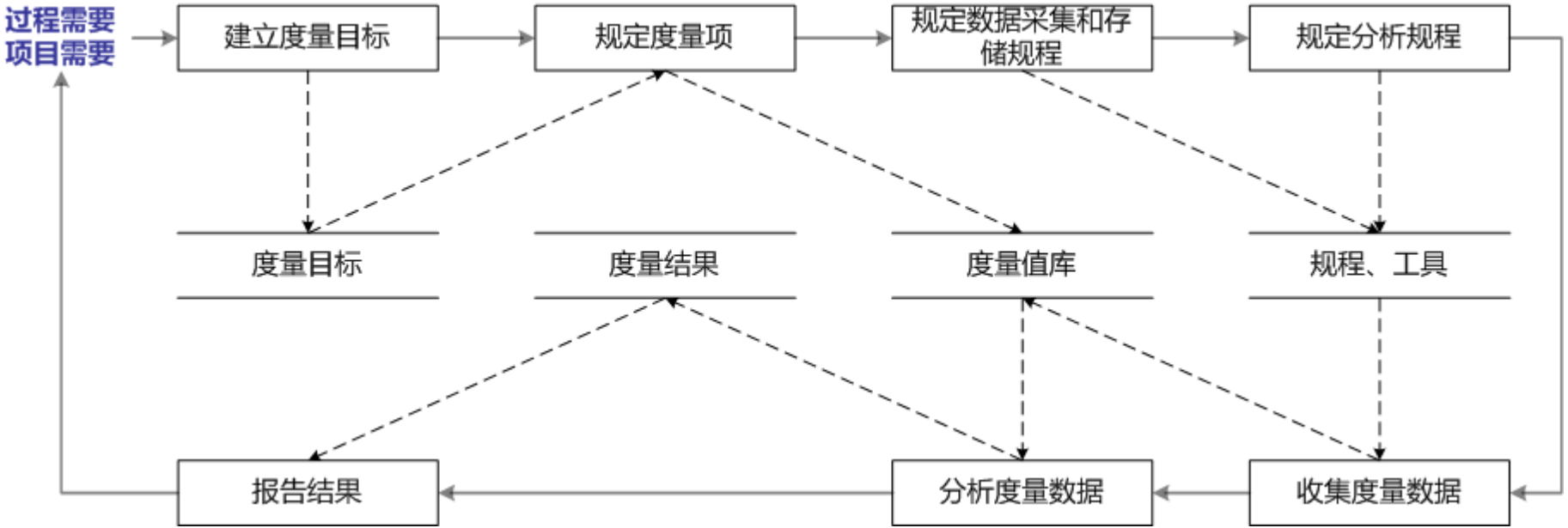
—— 特定目标和特定实践

关键过程域——测量和分析MA

SG1	协调测量和分析活动	
	SP1.1	建立度量目标
	SP1.2	规定度量值
	SP1.3	规定数据采集和存储规程
	SP1.4	规定分析规程
SG2	提供度量结果	
	SP2.1	收集度量数据
	SP2.2	分析度量数据
	SP2.3	存储数据和结果
	SP2.4	通报结果

3.3 度量和分析MA

— 特定目标和特定实践



3.3 度量和分析MA

从组织（项目）管理的需要导出度量目标。基于组织的业务目标或愿景，通过量化的数据来表征、衡量业务目标。

管理需要与度量目标的来源可能有：项目计划、项目绩效、管理目标、战略计划、业务计划、合同要求、重复问题、过程改进计划等。

可能的度量目标：改进开发过程、改进软件估算、改进项目控制、缩短项目周期、减少开发成本、提升软件质量、提升软件性能、减少项目风险发生等。

3.3 度量和分析MA

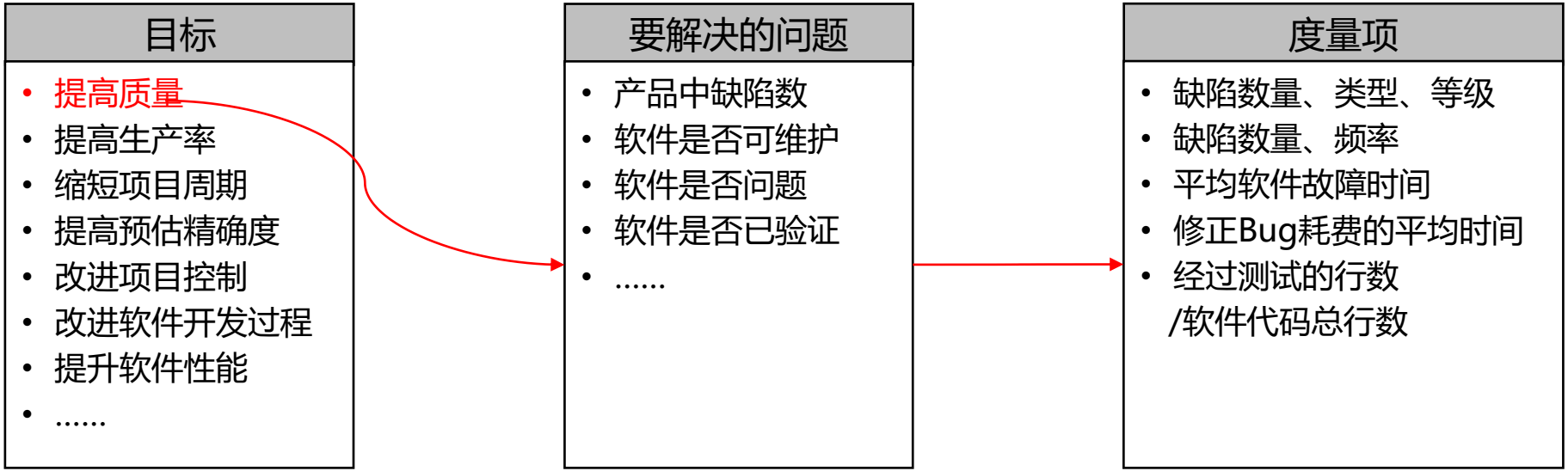
从组织（项目）管理的需要导出度量目标。基于组织的业务目标或愿景，通过量化的数据来表征、衡量业务目标。

管理需要与度量目标的来源可能有：项目计划、项目绩效、管理目标、战略计划、业务计划、合同要求、重复问题、过程改进计划等。

可能的度量目标：改进开发过程、改进软件估算、改进项目控制、缩短项目周期、减少开发成本、提升软件质量、提升软件性能、减少项目风险发生等。

3.3 度量和分析MA

明确说明应对度量目标的度量项，将度量目标细化为精确的、可度量的度量项。



3.3 度量和分析MA

定义度量项

度量目标与度量项举例

度量目标	关联活动	能力表示方式	基本度量项
增强项目估算的准确度?	项目估算	规模估算与实际对比	功能模块数
			实际项目的功能模块数
			文档规模
			实际项目的文档页数
			源代码行(LOC)
			实际项目的功源代码行数
		预计与实际工作量对比	项目估计工作量
			项目实际工作量
	成本估算与跟踪	成本对比	项目花费资金
			计划人力成本(BCWS)
			实际人力成本(ACWP)
			成本性能指数(CPI)
			成本差异(CV)
	进度预计与跟踪	项目进度	进度人力预算(BCWP)
			进度差异(SV)
			日程性能指数(SPI)
			项目工期

度量目标	关联活动	能力表示方式	基本度量项
提升产品质量	测试活动	case覆盖性	测试Case数
			Bug查出率
			测试花费工时
		严重bug数	Bug数
		测试密度	测试密度
		Bug率	Bug率
	评审活动	缺陷率	需求缺陷数
			概要设计缺陷个数
			详细设计缺陷个数
			文档页数
	走查	问题数/Kloc	代码走查问题数
	验收活动	未修正BUG数	基本检查实施确认
			交付前BUG分布分类统计
			交付后BUG分布分类统计
			交付前BUG发现／修改对比
			交付前BUG修正率统计
			交付前平均BUG修正周期
		需求不一致数	需求不一致数
		未处理缺陷数	未处理缺陷数
		QA的不符合数	QA的不符合数

3.3 度量和分析MA

明确说明如何获得并存储度量数据。

收集方法：有助于确保适当地收集正确的数据。可依据项目的实际情况、参考组织提供的《度量方法指南》，确定本项目的度量方法。

存储与检索规程：有助于确保数据的可用性和可访问性。

数据收集与存储规程举例

序号	管理目标	度量目标	度量指标	度量方法	度量数据存储
1	项目工作量。功能模块数体现了工作量方面的衡量属性	功能模块	功能模块数（个）	1. 依据《详细设计规格说明书》统计出功能模块总数。 2. 得到的资料填写到《度量数据表》中项目规模子表。	将《度量数据表》存放在配置管理系统中
2	项目规模如何衡量的一个标准	文档页数	文档的页数（页）	1. 分别统计项目开发各阶段文档资料的总页数。 2. 得到的资料填写到《度量数据表》中项目规模子表	将《度量数据表》存放在配置管理系统中
3	度量软件项目总规模	源代码行	源代码行数（LOC）	1. 使用工具软件LineCount进行统计 2. 将资料填写入《度量数据表》中项目规模子表。	将《度量数据表》存放在配置管理系统中
4	依据项目规模测算出平均生产率，得出项目产出能力	生产率	间接度量指标（LOC/人天）	1. 通过运算式：生产率 = 源代码行 / 项目工作量 (LOC/人天) 2. 评估结果填写入《度量数据表》中项目规模子表	将《度量数据表》存放在配置管理系统中

3.3 度量和分析MA

明确说明如何分析并沟通度量数据，确保能进行适当的分析与报告以应对文档化的度量目标。

分析规程应该说明所有进入分析的数据的质量。

《分析方法指南》

3.3 度量和分析MA

分析规程举例

序号	分析项目	度量目标	分析方法
1	工作量分析	项目总工时统计	方法1： 1、取得依度量指南产生的《度量数据表》中的项目工作量子表。 2、利用Excel的制图功能，产生工作量分布柱状分析图。 方法2： 1、依据度量指南中的工作量分布度量方法,取得工作量分布数据。 2、将度量数据填写入《项目进展报告》中的工作量分布表。 3、利用Excel的制图功能,产生工作量分布柱状分析图。
2	项目进度差异表	进度差异(SV)	1. 取得依度量指南产生的BCWS、BCWP、ACWP值。 2. 将度量数据填写入《项目进展报告》中的盈余分析表。 3. 利用Excel的制图功能，产生项目进度差异趋势分析图
3	项目成本差异	成本差异(CV)	1. 取得依度量指南产生的BCWS、BCWP、ACWP值。 2. 将度量数据填写入《项目进展跟踪报告》中的盈余分析表。 3. 利用Excel的制图功能，产生项目成本差异趋势分析图。

3.3 度量和分析MA

度量整体方案

序号	分析度量项目	举例：工作量度量
1	度量目标	为下个节点工作计划提供依据和参考
2	项目管理目标	管理整个项目花费的计划和实际总工作量
3	度量指标	活动工作量分布
4	指标表示	（1）用表格表示原始计划工作量（人天）、实际工作量（人天）、原始计划工作量偏离、实际工期、标准人数、各工作类型（开发、PPQA、CM、项目管理）工作量；（2）用图表表示生命周期各阶段工作量、工作量偏差
5	分析方法	（1）查看阶段工作量分布情况；（2）查看各类活动的工作量分布和占比
6	加工数据	（1）阶段工作量分布；（2）各阶段各类活动的工作量分布；（3）各活动总的工作量分布
7	加工方法	（1）阶段工作量/总工作量；（2）阶段活动工作量/阶段工作量；（3）活动总工作量/总工作量
8	统计数据	（1）阶段工作量；（2）总工作量；（3）阶段活动工作量；（4）活动总工作量
9	统计方法	（1）阶段内各活动工作量求和；（2）各阶段活动工作量求和；（3）各阶段活动工作量求和
10	统计来源	日志
11	统计单位	人时
12	采集、验证和存储规程	按照项目经理要求的频率（通常为每周）从日志表采集数据，填入“项目工作量分布”度量指标表，存入配置管理库。验证人根据度量数据检查单提出的问题来验证度量数据的完整性、有效性
13	分析、通报规程	按照项目经理要求的频率（通常为每周或每月）对采集和计划的数据进行分析。根据统计数据和加工数据用图表方式表示度量指标。分析结果汇报给项目经理或高层经理

3.3 度量和分析MA

与所有相关干系人沟通与分析活动的结果。《项目进展报告》

1、项目概况信息

（1）生产率： 实际工作量（人月）、项目规模（千行代码数KLOC、功能点数FP）、生产率（KLOC/人月、FP/人月）

（2）质量：验收缺陷数、产品质量（缺陷数/FP）

（3）客户投诉情况：

2、项目工作期

分阶段、分任务、分活动，描述计划进度、实际进度、项目开始日期和工期得延迟情况。

	计划进度		实际进度		延迟	
活动任务	开始日期	工期	开始日期	工期	开始日期	工期
需求分析						
概要设计						
详细设计						
…						

3、过程实施情况：组织标准过程裁剪情况，形成的项目过程；项目过程的执行情况。

4、采用工具：开发工具、管理工具

5、风险管理：识别的风险、发生的风险、风险缓解措施及成效。

6、项目规模：（1）使用语言；（2）KLOC、FP；（3）（简单、中等、复杂）单元规模统计；

7、工作量：（1）项目组人数；（2）估计工作量；（3）实际工作量；（4）质量成本=（评审工作量+返工工作量+测试工作量+项目能力培训工作量）/ 总工作量 × 100%；（5）工作量分布；

3.3 度量和分析MA

8、缺陷情况

（1）缺陷分布

阶段	估计		实际		偏差（%）
	缺陷数	比例	缺陷数	比例	
需求评审					
设计评审					
开发结束评审					
系统测试					
验收测试					
总计					

（2）缺陷消除率

阶段	估计		实际		偏差（%）
	缺陷数	比例	缺陷数	比例	
需求评审					
设计评审					
开发结束评审					
系统测试					
验收测试					
总计					

（3）缺陷严重性分布

序号	严重性	缺陷数	占比（%）
1	轻微		
2	次要		
3	严重		
4	重大		
5	其他		
	总计		

（4）缺陷类型分布

序号	缺陷类型	缺陷数	占比（%）
1	设计问题		
2	不良代码		
3	初始化		
4	逻辑处理		
5	输入输出		
6	系统接口		
7	其他		
	总计		

3.3 度量和分析MA

9、偏差分析：偏差现象、偏差原因、结论意见

10、过程资产：

项目过程定义；

项目估算、项目计划、配置管理计划、质量保证计划；

项目过程数据库、项目文档库、度量值库

- 3.1 软件质量保证SQA
- 3.2 配置管理CM
- 3.3 度量与分析MA
- 3.4 原因分析与解决CAR
- 3.5 决策分析与解决DAR

3.4原因分析与解决CAR

—— 目的

原因分析与解决的目的在于识别导致缺陷和其他问题的根本原因，在理解已定义过程和实施已定义过程的基础上，判断这些缺陷产生的根源和这些根源存在的程度，从而找出对策、采取措施消除问题的根源，防止将来再次发生同类问题。

3.4原因分析与解决CAR

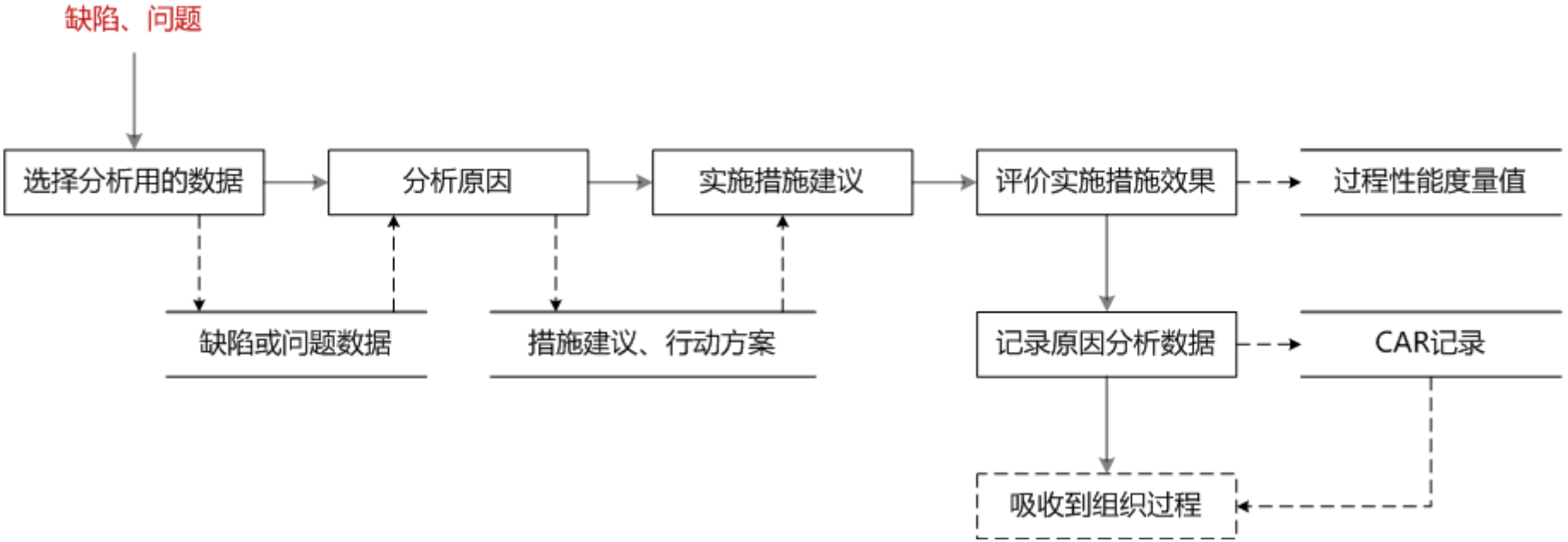
—— 特定目标和特定实践

关键过程域——原因分析与解决方案CAR

SG1	确定缺陷原因	
	SP1.1	选择分析用的数据
	SP1.2	分析原因
SG2	处理缺陷原因	
	SP2.1	实施措施建议
	SP2.2	评价变更的效果
	SP2.3	记录数据

3.4原因分析与解决CAR

—— 特定目标和特定实践



3.4原因分析与解决CAR

分析时机

原因分析与解决通过避免将缺陷引入产品，以改进质量和生产力。在缺陷发生后再进行侦测，是不符合成本效益的。更有效的方式是，将原因分析与解决的活动，集成到项目的每个阶段，提取预防缺陷和问题发生，这样更为有效。

所有的缺陷进行原因分析并不实际。在这种情况下，必须在估计的投资与预期的质量、生产力、周期时间、所选择的缺陷目标的回报之间做权衡。

3.4原因分析与解决CAR

分析时机

作为寻找问题根本原因的过程和方法，原因分析既可以用于组织的过程改进也可以用于项目层面。

组织层面多采用定期分析的方法，如每年度针对组织商业目标的实际情况或某个过程的特殊要求进行改进性分析。

项目层面多采用实践驱动的方法，如项目出现了与预期目标严重偏差或某个过程出现异常情况。通常在项目某个阶段结束或稍前时间。

3.4原因分析与解决CAR

选择问题

要进行“刨根究底式”的原因分析成本是比较高的，企业没有必要全部问题都采用原因分析，次要问题或显而易见的就不需要采用原因分析过程，要选择有价值的问题进行这种分析，这些问题的改善对提高企业过程能力、实现企业商业目标是非常有用的。组织级选择原因分析的原则：

- 组织级的过程性能基线或过程性目标没有达到预期目标的
- 公司的KPI没有达到预期目标
- 公司多个项目均存在共性问题，需上升到组织级

3.4原因分析与解决CAR

选择问题

原因分析与解决CAR场景举例：

1、问题或失败反复多次出现时，同样的问题多次在项目或者组织执行中出现。

2、项目实施、组织改进执行过程中明显与计划目标或需求发生偏移时。

3、测试阶段BUG的数量及比例远远高于预期时。

4、过程性能目标超出或低于预期时。

5、某一改进流程执行结果不能够满足预期质量与过程性能目标时。

3.4原因分析与解决CAR

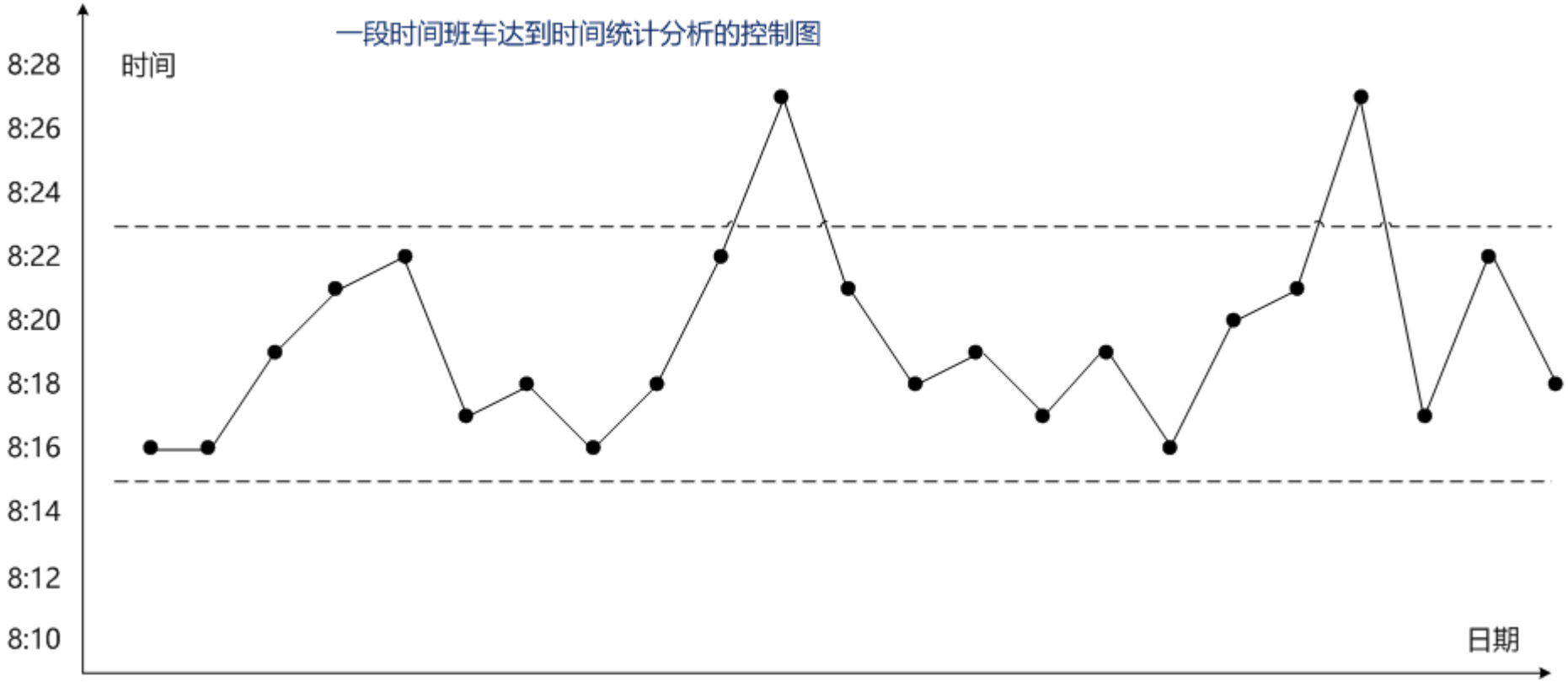
首先制定原因分析计划，明确问题来源，包括实际过程与目标偏差、内部或外部客户调查、商业目标改变等。

然后开展原因分析，采用的分析工具包括统计分析控制图、鱼骨图、PARRTO图、散点图、假设检验、方差分析等。

最后文档化行动方案：行动建议的提出者、原因的描述、原因的分类、行动建议描述、实施行动建议所需要的时间、成本及其他资源，实施行动的预期收益、行动建议分类等。

3.4原因分析与解决CAR

统计分析控制图



3.4原因分析与解决CAR

统计分析控制图

控制范围、预期范围；

预期范围内，由于交通流量、天气等共同原因引起的变化；

预期范围外，特殊原因引起的变化，一次是车门出了问题、一次是换了司机；

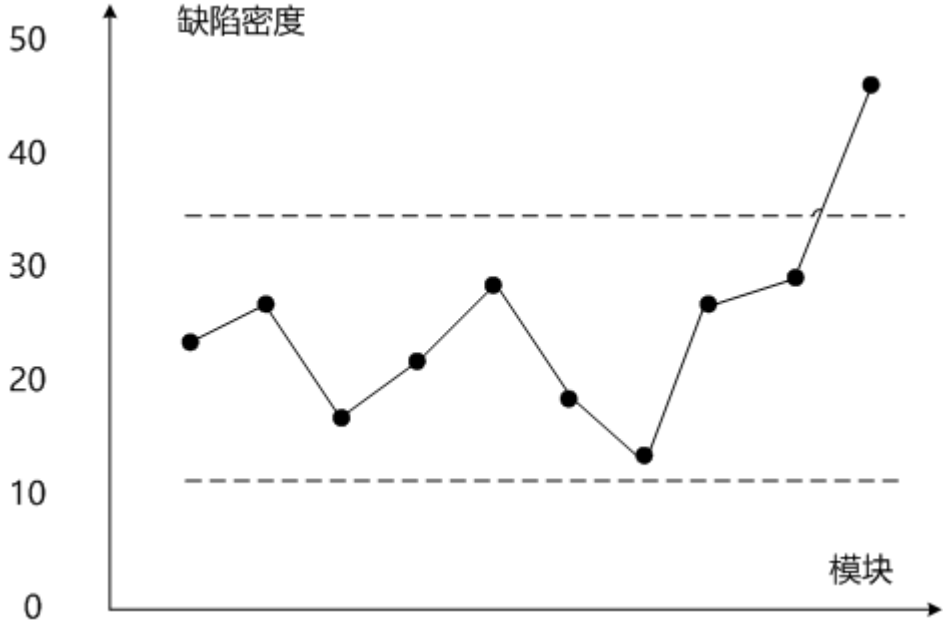
统计分析的控制图，目的是保证处于预期的控制范围内。对于控制范围之外的，要检测分析出原因，并加以消除。

3.4原因分析与解决CAR

统计分析控制图

代码评审的缺陷密度数据

模块	1	2	3	4	5	6	7	8	9	10
缺陷密度	21.1	22.4	16.3	20.6	24.2	17.9	12.2	24	24.7	44.2



分析第10个模块的特殊原因：

是否按规程进行评审？

评审组同行的素质？

评审工作细致了？

未做设计评审？

未做代码走查？

3.4原因分析与解决CAR

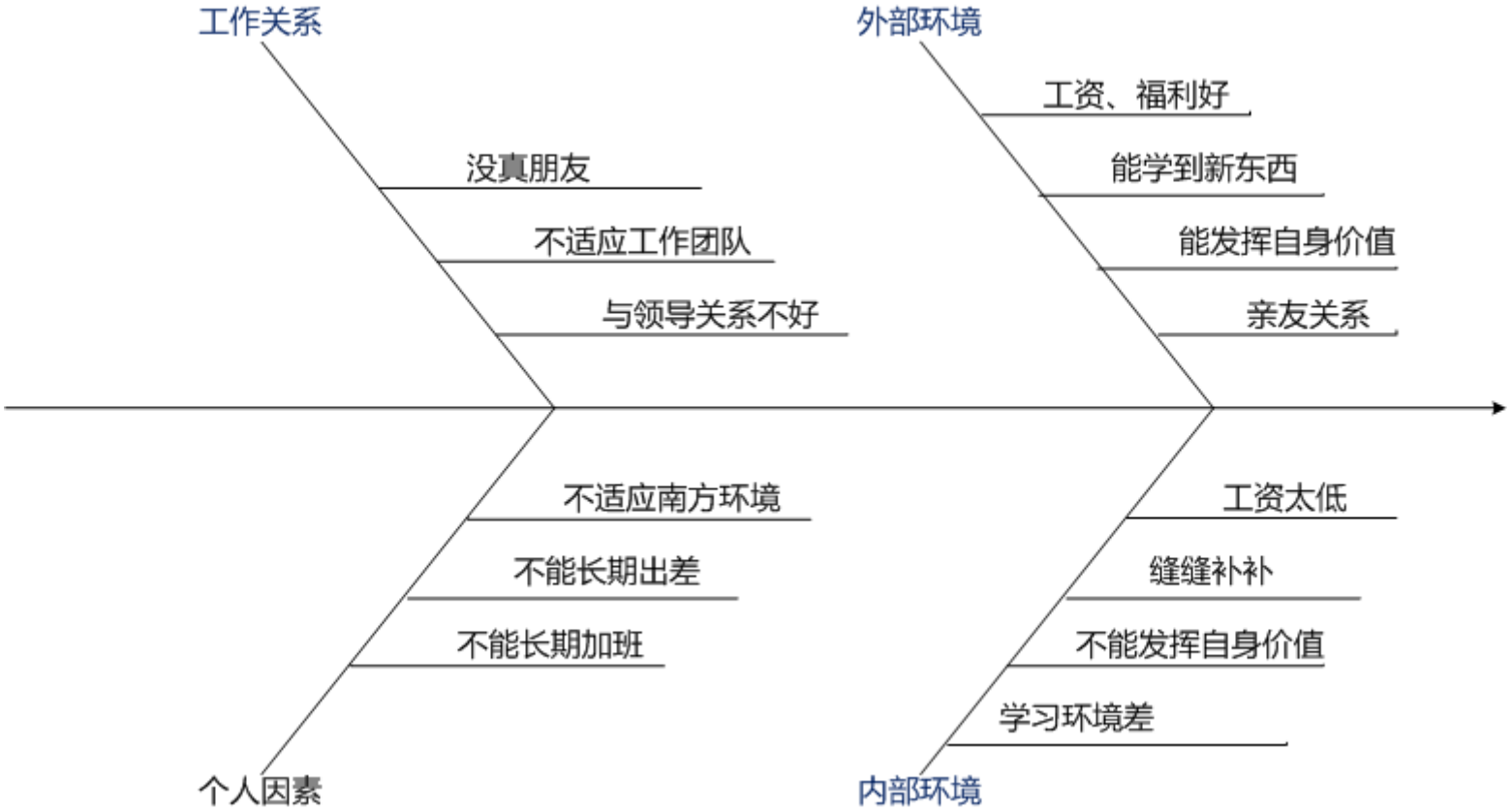
统计分析控制图

任何一个经典控制图都有一条中心线，在其两边各有一条控制边界。中心线和上下控制边界是从采集的数据值计算出来的，是一种估计值。

中心线一般取数据值的平均值 $\overline{x}$。设σ是标准方差，上下控制边界一般取 $\overline{x} \pm 3\sigma$。

3.4原因分析与解决CAR

鱼骨图分析法



3.4原因分析与解决CAR

鱼骨图分析法

①分析结构：针对问题点，甄别不同层别的方法，例如：工作关系、个人因素、外部环境、内部环境等，按头脑风暴分别对各层别、类别找出所有可能原因。

②分析要点：需要尽可能多而全地找出所有可能原因，而不仅限于自己能完全掌控或正在执行的内容。场景化还原现场，通过相对条件的比较，找出相关性最强的要因归类。最后在选取重要原因时，应标识在鱼骨刺的最末端。

③使用步骤：首先查找要解决的问题，把问题写在鱼骨的头上，共同讨论问题出现的可能原因，研究为什么会产生这样的问题，针对问题的答案再问为什么？

3.4原因分析与解决CAR

制定并实施行动计划

制定行动计划。包括：负责实施行动计划的小组成员、对行动计划的详细描述及对应的进度计划、实施行动的预期目标、用于评估实施效果的度量项及评估工具、采集度量数据的时机、评估实施效果的时机、实施中定期沟通的干系人、实施行动的成本等；

实施行动计划。将相关任务分配给执行小组的各成员，执行小组成员按照行动计划的要求实施行动，CAR负责人通过PMC进行过程跟踪，度量员按照计划要求定期收集度量数据，CAR负责人记录实施中出现的风险和问题。

3.4原因分析与解决CAR

CAR负责人根据收集到的度量数据，利用评估工具对度量数据进行分析，观察行动后组织/项目中的问题或缺陷是否达到预期目标。

如果改进效果未达到预期的目标，CAR负责人需进行分析并判断是否需要重新进行新一轮的CAR过程。若果达到预期目标，CAR负责人可以建立相应的模型以制定以后的组织或项目活动，并通过EPG对组织的标准过程进行更新。

3.4原因分析与解决CAR

做好原因分析的建议

（1）清楚地描述问题，包括缺陷的严重性、缺陷的关联性和分类、问题被引入和发现的时间等；

（2）分析措施建议和确定他们的优先顺序，主要根据缺陷可能对质量和效率造成的影响，为防止这类缺陷而实施过程改进的成本等因素决定；

（3）要消除其中有差错倾向的过程环节，就要改进相应的过程；

（4）使整个过程或子过程实施自动化，可有效提高过程管理的质量和效率；

（5）补充防止缺陷的步骤，如记录过程活动、增加开发和维护期间的例行会议；

（6）就普遍性问题和防止他们再次发生的措施，需要对所有员工进行培训。

- 3.1 软件质量保证SQA
- 3.2 配置管理CM
- 3.3 度量与分析MA
- 3.4 原因分析与解决CAR
- 3.5 决策分析与解决DAR

3.5 决策分析与解决DAR

—— 目的

决策分析与解决的目的是使用正式的评价过程，遵循已建立的准则，对已识别的多个备选方案进行评价，选择可行的解决方案。

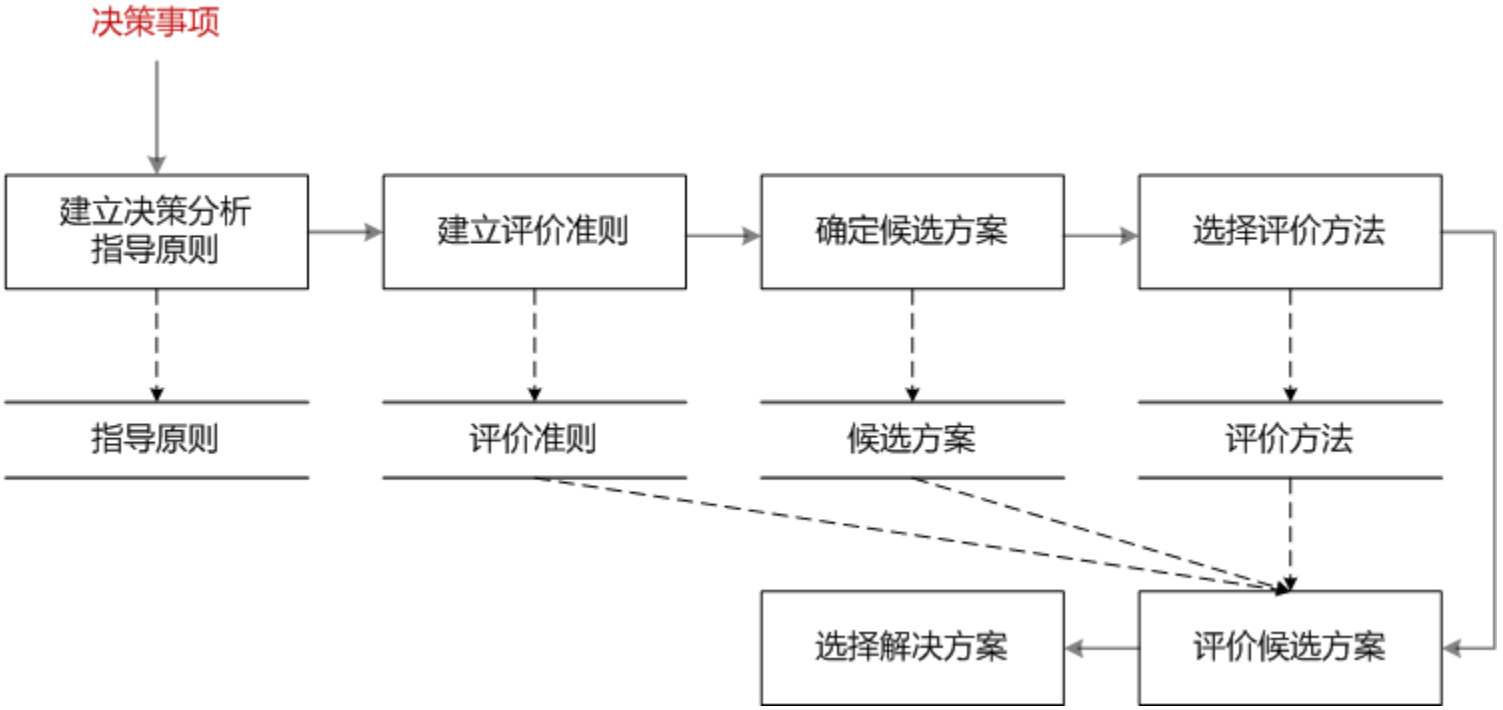
3.5 决策分析与解决DAR

—— 特定目标和特定实践

关键过程域--决策分析与解决DAR		
SG1	根据一定的标准对可选方案进行评估的基础上进行决策	
	SP1.1	建立决策分析指导原则
	SP1.2	建立评价准则
	SP1.3	设别可选的解决方案
	SP1.4	选择评估的办法
	SP1.5	评价备选的解决方案
	SP1.6	选择解决方案

3.5 决策分析与解决DAR

—— 特定目标和特定实践



3.5 决策分析与解决DAR

决策分析指导原则

（1）决策直接关系到的论题可能是中高风险的；

（2）决策关系到对软件配置管理之下的工作产品的变更；

（3）如果决策不当，可能造成进度拖延、成本大幅提升；

（4）决策影响到项目目标实现的能力；

（5）决策过程的成本可以接受；

（6）当获得20%的材料而达到80%的总体材料成本；

（7）技术上的失败将造成严重后果；

（8）可能大大减小风险、工程变更和生产成本的问题。

3.5 决策分析与解决DAR

项目管理中的决策议题举例：

设备选择、软件技术方案研究、架构或设计方案选择、使用复用或现成品组件、选择供应商、工程支持环境或相关工具、测试环境、交付的备选方案等；

技术方案选型是比较揪心的一个过程，需要在满足需求、技术先进性、成熟度、成本、人力、工期等方面进行平衡，而且这些方案是项目干系人和项目组成员在头脑风暴会议上，通过快速互动、刺激提出有创意的备选技术解决方案。

3.5 决策分析与解决DAR

作为决策基础的评价准则，就是判断候选方案优劣、选择最佳方案的依据，这些准则必须反映相关利益者的需求和目标。

评价准则的建立，应该追溯到客户需求、应用场景、用例和质量标准等，并考虑技术限制、环境影响、风险、所有权和生命周期成本等因素。

通常将评价准则定为不同的优先级或权重因子，优先级高的准则对决策的影响最大。

举例说明：确定采购方案时，应考虑采购对象的如下特征：可操作性、复杂性、是否有中文版、价格、性能、市场占有率、厂家的大小、厂家的名声或声誉等。以及这个特性在整个评价中所占的比例，也就是说某个特征对评价结果的影响程度。

3.5 决策分析与解决DAR

Delphi方法，对所要预测的问题，按照规定的程序，背靠背地征询专家的意见，经过几轮征询，使专家小组的意见趋于集中的方法;

加权打分法，将评分的准则设置不同的权重或系数，评分人针对评分细则分别打分，然后加权汇总得到结果。

头脑风暴会议，经过相互启发和反馈，可以激发出变革型的解决方案；

专家组的咨询，基于领域知识和经验，可以得到较全面的解决方案；

调查分析法，全面调查分析、并参照组织内外的类似案例。

3.5 决策分析与解决DAR

确定可选方案：决策人根据要决策对象的实际情况，列出尽可能多的解决方案，一般三个以上，每个方案都要有自己的原因说明、解决办法等。

评估方案：根据已指定的评估方法和评估准则对可选方案进行评估，从候选方案中选出解决方案。

重要的是既要记录下为什么选择某一解决方案，也要记录为什么弃用另一项解决方案。

课后作业1

1、软件过程的定义

软件过程（Software Process，SP）是人们建立、维护和演化软件产品整个过程中所有技术活动和管理活动的集合，是人们用来开发和维护软件及其相关产品的活动、方法、实践和改进的集合。其中相关产品是指项目计划、设计文档、代码、测试文档和用户手册等。

课后作业1

2、CMM/CMMI 模型使用什么表达方式来描述软件过程改进?

CMM/CMMI 使用级别为一个要改进其软件开发过程的组织描述了一种推荐的演变路径。

CMM/CMMI支持两种使用级别的改进路径。即能力级别和成熟度级别。能力级别路径使组织能够渐增地改进由组织选定的、一个（或几个）单独过程域的过程。成熟度级别路径使组织能够通过渐增地解决连续的过程域集来改进相关的过程集合。

课后作业1

3、**组织过程焦点过程域主要解决什么问题?**

建立并维护本组织的过程和过程财富，以及识别、策划和实施本组织的过程改进活动。

组织过程焦点，要求高级管理者、部门经理、全体成员都能聚焦到软件过程，确保组织全员对软件过程活动的决心、分工和职责，成立软件工程过程组EPG，促进并保持对软件过程的制订、维护、执行、评估、改进，也就是统一认识、统一行动。

课后作业2

1、什么是质量保证?

2、什么是基线，常见的软件配置基线有哪些?

3、度量的目标是什么?

邮箱：jgzhen4406@sina.com

文件命名：课后作业2（姓名，学号）