



# 软件过程与项目管理

**2021年11月30日星期二**

---

## ◆需求管理

# 需求的重要性

---

◆Frederick Brooks在他1987年经典文章“**No Silver Bullet**”中阐述了需求的重要性：

- 开发软件系统最困难的部分就是准确说明开发什么。最困难的概念性工作是编写出详细的需求，包括所有面向用户、面向机器和其它软件系统的接口。此工作一旦做错，将会给系统带来极大的损害，并且以后对它修改也极为困难。

◆需求是产品的根源，需求工作的优劣对产品影响最大。就像一条河流，如果源头被污染了，那么整条河流也就被污染了。

◆国内软件业的痼疾：人们并不清楚究竟该做什么，但却一直忙碌不停地开发。

# 什么是需求-基本概念

---

◆在IEEE软件工程标准词汇表(1997年)中定义软件需求为:

(1) 用户解决问题或达到目标所需的条件或能力。

(2) 系统或系统部件要满足合同、标准、规范或其它正式规定文档所需具有的条件或能力。

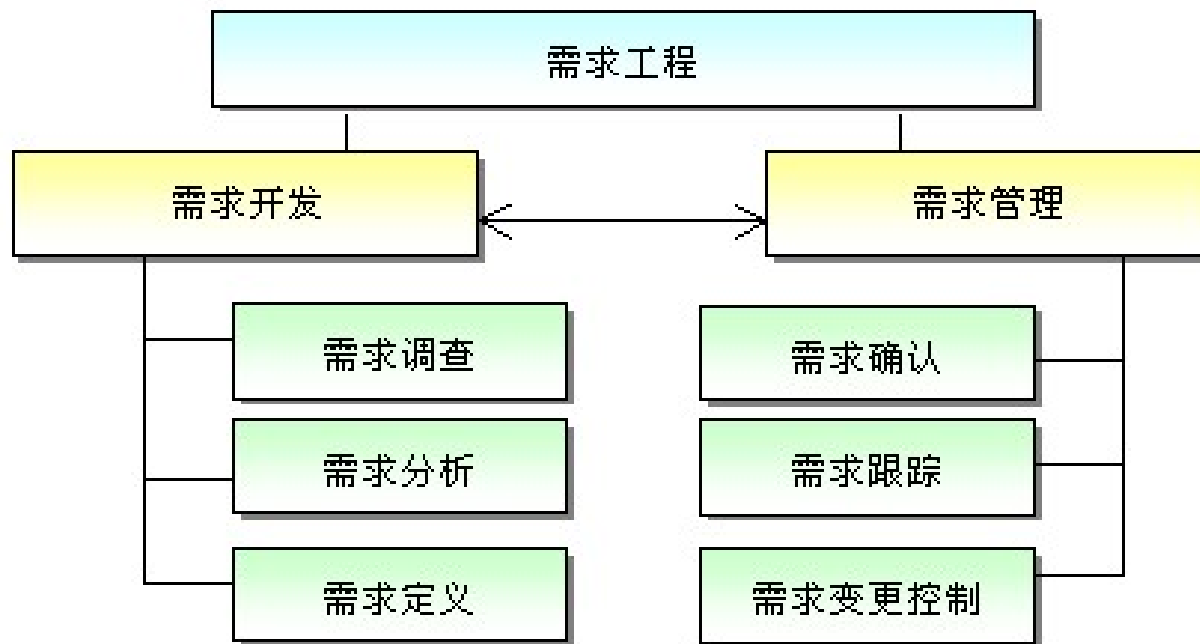
(3) 一种反映上面(1)或(2)所描述的条件或能力的文档说明

◆宽泛地讲, 需求来源于用户的一些“需要”, 这些“需要”被分析、确认后形成完整的文档, 该文档详细地说明了产品“必须或应当”做什么。

# 需求工程

## ◆ 什么是需求工程

- 把所有与需求直接相关的活动通称为需求工程。
- 需求工程中的活动可分为两大类，一类属于需求开发，另一类属于需求管理。
- 需求工程的结构图



# 需求开发

---

◆ **需求开发**的目的是通过调查和分析，获取用户需求并定义产品需求。

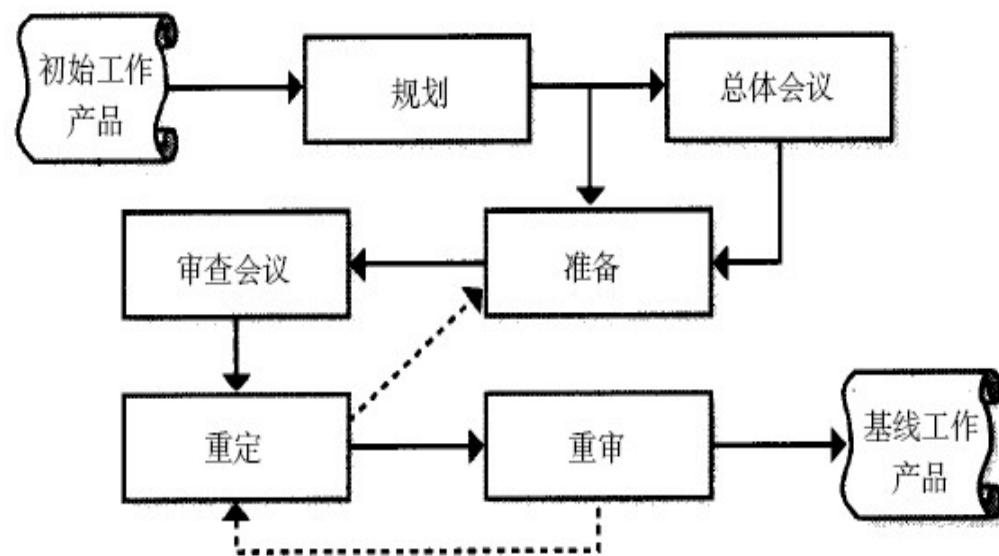
- **需求调查**的目的是通过各种途径获取用户的需求信息(原始材料)，产生<<用户需求说明书>>。
- **需求分析**的目的是对各种需求信息进行分析，消除错误，刻画细节等。
- **需求定义**的目的是根据需求调查和需求分析的结果，进一步定义准确无误的产品需求，产生<<产品需求规格说明书>>，以便开展系统设计工作。

# 需求管理

---

- ◆ **需求管理**的目的是在客户与开发方之间建立对需求的共同理解，维护需求与其它工作成果的一致性，并控制需求的变更。
  - **需求确认**是指开发方和客户共同对需求文档进行评审，双方对需求达成共识后作出书面承诺，使需求文档具有商业合同效果。
  - **需求跟踪**是指通过比较需求文档与后续工作成果之间的对应关系，建立与维护“需求跟踪矩阵”，确保产品依据需求文档进行开发。
  - **需求变更控制**是指依据“变更申请—审批—更改—重新确认”的流程处理需求的变更，防止需求变更失去控制而导致项目发生混乱。

# 实际中的相关活动或者文档



## 需求评审

	时 间	13：30-16：30
	主 题	GAP项目需求评审
	记 录	盛利娜
娜、文意		

包括公司信息、页面组件、模块菜单管理、角色管理、岗位管理、用户管理、组织机构管理、密码策略管理、所有密码失效管理、修改密码、用户登录密码)。

## 会议纪要



# 管理需求变更时面临的现状分析

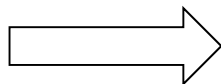
---

◆思考问题：

为什么存在变更？

有哪些主要的变更？

如何对变更进行控制？



需求变更的原因？

导致的问题？

如何实现需求变更管理？

# 需求变更原因分析

---

## ◆需求发生变更的起因主要有：

- 随着项目的进展，人们（包括开发方和客户方）对需求的了解越来越深入。原先的需求文档可能存在这样那样的错误或不足，因此要变更需求。
- 市场发生了变化，原先的需求文档可能跟不上当前的市场需求，因此要变更需求。

◆提出需求变更的动机是好的，目的是希望产品更加符合用户的需求。对项目开发小组而言，变更需求意味着要调整资源、重新分配任务、修改前期工作成果等，开发小组要为此付出较重的代价。如果每次需求变更请求都被采纳的话，这个项目也许永远不能按时完成。

# 需求变更的控制思路

---

- ◆需求变更控制的目的： 如果需求变更带来的好处大于坏处，那么允许变更，但必须按照已定义的变更规程执行，以免变更失去控制。 如果需求变更带来的坏处大于好处，那么拒绝变更。
- ◆需求变更控制过程中最难办的事情是莫过于“拒绝客户提出的需求变更请求”。通常情况下开发方是不敢得罪客户的，但是无原则地退让将使开发小组陷入困境。 如何解决？
- ◆解决这个问题最好的办法是事先建立“游戏规则”：
  - 开发方与客户方达成“事不过三”的约定（符合中国人的习惯），即允许客户变更三次需求；如果客户第四次变更需求，开发方有权拒绝，除非客户愿意补偿开发方的损失。
  - 如果事先没有“游戏规则”的话，开发方需要一些社交技巧来减缓矛盾。例如建议在开发该产品新版本时修改需求。

# 需求变更管理

---

## ◆需求变更的管理

- 主要涉及分配需求的更改活动、分配需求的更改次数等
- 需求变更从需求说明书或建议开始生效，变更在相关的计划、交付和行为中反映

## ◆项目经理对将变更纳入项目中所需的过程执行负主要责任

## ◆过程输入输出

- 过程的入口准则是接收变更请求，输入则是变更请求和项目已经生产出来的工作产品；
- 主要输出是需求变更的影响分析报告、更改后的计划以及更改后的工作产品，输出准则是变更已经被包含在内

# 需求变更管理相关活动

---

◆需求变更管理主要通过以下的活动来实现：

- 确定需求变更控制过程      **??? 不同阶段的控制策略**
- 建立需求变更控制委员会
- 进行需求变更影响分析      **??? 分析考虑因素**
- 跟踪所有受需求变更影响的工作产品
- 调整需求基线
- 维护需求变更记录和文档

# 变更影响分析

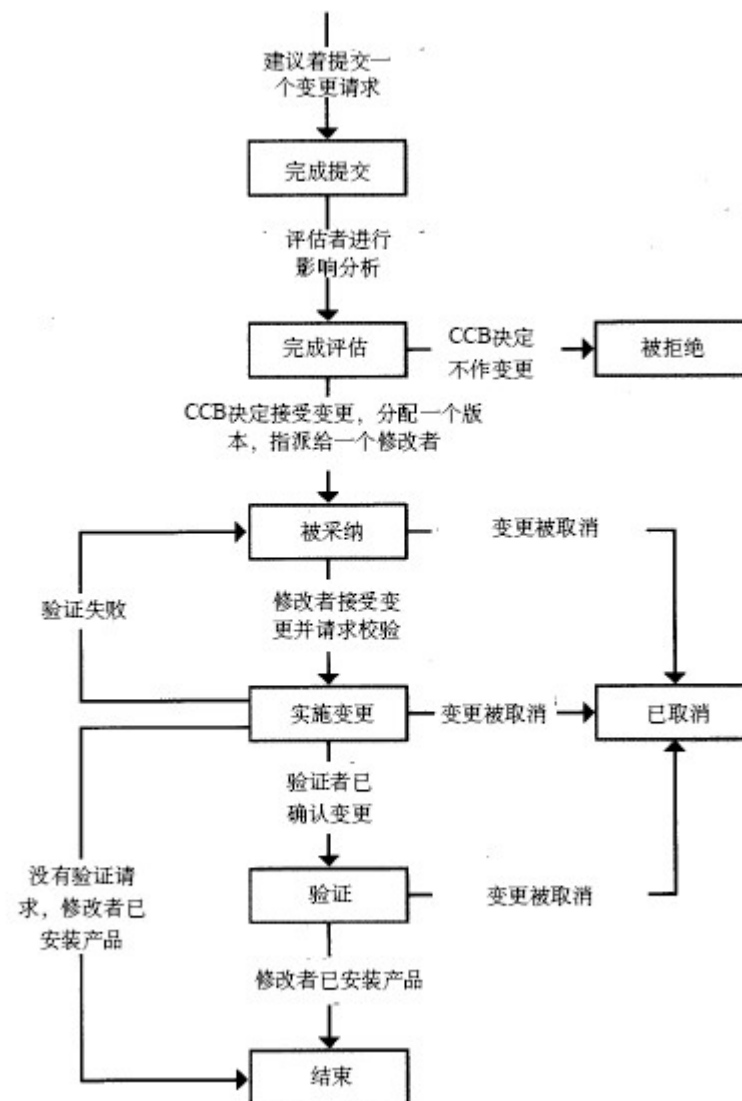
---

◆影响分析可以提供对建议的变更的准确理解，帮助做出信息量充分的变更批准决策。通过对变更内容的检验，确定对现有的系统做出是修改或抛弃的决定，或者创建新系统以及评估每个任务的工作量。进行影响分析的能力依赖于需求跟踪矩阵的质量和完整性。

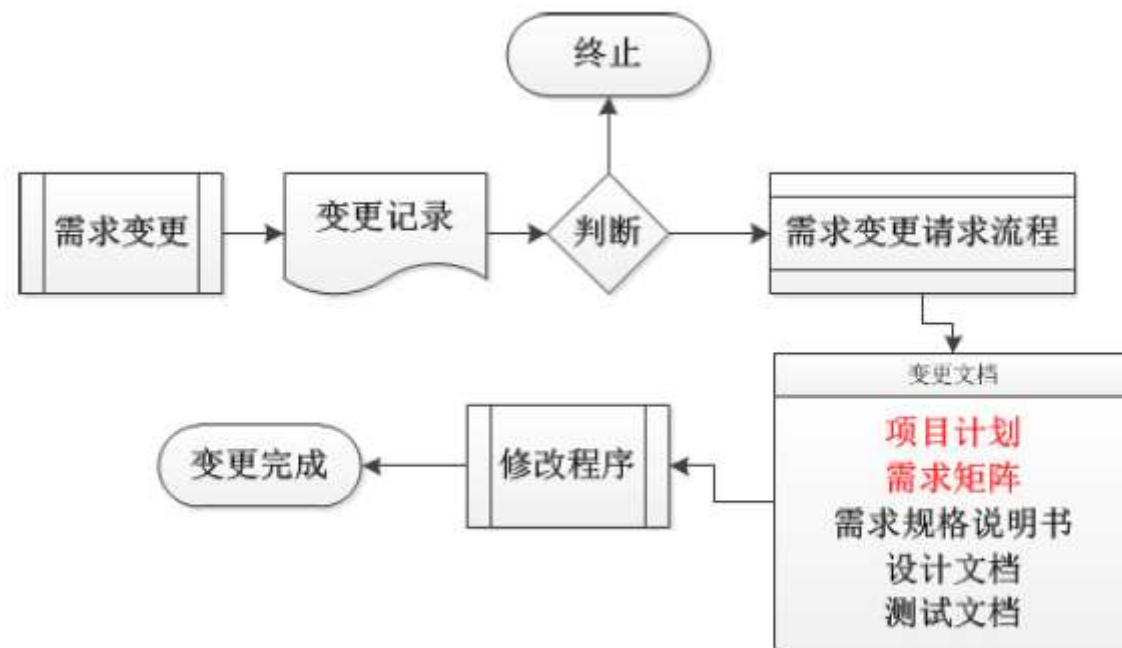
◆CCB (Change Control Board)—变更控制委员会

变更控制委员会可以由一个小组担任，也可由多个不同的组担任，负责做出决定究竟将哪一些已建议需求变更或新产品特性付诸应用。典型的变更控制委员会同样决定在哪一些版本中纠正哪一些错误。许多项目已经有负责变更决策的人员，而正式组建变更控制委员、制定操作步骤会使他们更有效地工作。

# 需求变更流程



# 需求变更管理相关活动文档





# 需求变更表格

***项目		
请求号		日期
变更规格		
影响分析		
进度影响		
工作量影响		
状态		

---

## ◆项目规划

# 估算方法分类

---

## ◆常用的项目估算方法分类

- 数学模型-----适用于非常成熟的软件公司
- 简单直观的“分解-累计”方法
  - 产品规模估计方法
  - 工作量估计方法

# 简单直观的估计方法

## ◆产品规模估计方法

- 项目规划小组先分解产品的功能，制定“产品功能分解和规模估计表”，软件规模的度量单位主要有：代码行、对象个数、页面数等等
- 规划小组各成员独立填写表格
- 汇总每个成员的表格，进行对比分析

模块名称	模块的主要功能	新开发的软件规模 (度量单位如对象个数)	复用的软件规模 (度量单位如对象个数)
模块 A	A.1		
	A.2		
	A.3		
模块 B	B.1		
	B.2		
	B.3		
汇总			

# 简单直观的估计方法

## ◆工作量估计方法

- 先估算开发工作量，一般可以把开发过程分为需求开发、系统设计、实现、测试四个阶段，分别估计每个阶段的工作量，然后累计出总的工作量

工作量的度量单位	1 人年 = 12 人月, 1 人月 $\approx$ 22 人天, 1 人天 = 8 人小时		
开发工作量估算公式	项目开发工作量 $\approx$ 新开发的软件规模 / 人均生产率		
开发阶段	人均生产率	软件规模	工作量
需求开发			
系统设计			
实现			
测试			
...			
开发工作量汇总			
管理工作量估算公式	项目管理工作量 $\approx$ 开发工作量 * 比例系数		
项目管理的主要事务	项目规划、项目监控、需求管理、配置管理、质量管理...		
比例系数	例如 20%		
管理工作量汇总			

# WBS

---

- ◆工作量估算的依据是项目的WBS（工作分解结构,work breakdown structure）分解
- ◆WBS将一个项目分解成易于管理的几个部分或者几个子项目，以确保找出完成项目工作项目工作范围所需的所有工作要素，
- ◆WBS是一种在项目全范围内分解和定义各层次工作包的方法，按照项目发展的规律，依据一定的原则和规定，进行系统化的、相互关联和协调的层次分解。结构层次越接近下层，则项目组成部分的定义越详细，最后成为层次清晰、可以具体作为组织项目实施的工作依据

# 估算过程

---

- ◆决定分析类型
- ◆识别范围和应用边界
- ◆计算未经调整的功能点
  - 数据功能
  - 处理功能
- ◆计算调整系数
- ◆计算调整后的功能点

# 实际工作中估算注意事项--1

---

## ◆使用什么单位？

- “一个功能点或一个SLOC的确切定义并不是非常重要的，就好像没有人真正关心一英尺和一米到底有多长一样。重要的是每个人都使用同样的定义” - Walker Royce, IBM Rational

## ◆使用什么样的模型？

- 目前有COCOMO, Price-S等10余种模型，近50个成本估算工具，但精度？

## ◆先估算规模后计算工作量？

- 一个来自传统项目的观点：要计算挖一条沟的人工，先估计一下要挖多少土方



# 实际工作中估算注意事项--2

---

## ◆做好估算的基本条件

- 定义清晰的需求
- 工作拆分充分
- 可用的历史数据
  - 类似的项目
  - Project Profile
- 不断的调整
  - If you can' t plan well, plan often!

# 实际工作中估算注意事项--3

---

## ◆软件项目工作量估算的其他影响因素

- 复杂度包括问题领域、算法复杂性、程序设计语言、软件复用率、可靠性等性能要求、系统平台复杂性、资源的限制等
- 人为因素包括开发人员的能力、经验、稳定性，开发的组织管理能力，用户的配合等
- 工程因素包括开发技术的难度、进度的紧迫性、项目团队的凝聚力、多地点开发等
- 意外事件

## ◆综合修正

- 为了减少误差，需对估算值进行修正和调整
- 通常采用三个时间估计法，乐观时间a、悲观事件b、正常时间m，因此期望时间 $t = (a + 4m + b) / 6$

## 实际工作中估算注意事项--4

---

- ◆在进行工作量估算时，可能会比较集中地考虑需求分析、设计、编码、测试等的工作量，
- ◆往往会忽略以下一些工作量：
  - 用于各模块、子系统、软件系统与硬件/网络系统之间集成的测试、调试等的工作量
  - 用于编写用户文档和设计文档的工作量
  - 用于需求管理、配置管理、质量管理、风险管理等支持过程的工作量
  - 用于项目管理的工作量

# 如何进行进度安排？

---

- ◆确定项目整体持续时间
- ◆决定主要的里程碑安排
  - 理解软件项目中人力分布
- ◆设定详细的进度

任务名称	持续时间 (天)	开始时间	结束时间	资源名	工作量 (小时)

# 编制预算和进度过程中的注意事项

---

## ◆为变更提供预留

- 承认项目中存在变更和问题
- 为这些变更和问题提供管理预留
- 结构不清的项目应该留出更大的预留
- 需要为管理预留的使用建立规范
  - 变更控制流程
  - 管理预留使用的授权 – 客户和高级经理

# 项目风险

---

- ◆何为项目风险?
- ◆项目风险有哪些?
- ◆如何识别项目风险?
- ◆识别后该怎么做?
  
- ◆请列表

# 如何编制项目进度计划？？

---

- ◆项目进度计划：主要包括进度规划（里程碑计划）、阶段计划和个人安排
- ◆进度规划：在项目启动时项目经理制定项目进度规划（一般按软件生命周期的阶段划分，明确项目里程碑）
- ◆阶段计划：在各阶段中项目经理及相关角色制定阶段计划（如调研计划、需求分析计划、系统设计计划、编码计划、测试计划、质量保证计划、安装调试计划、培训计划、验收计划等等）
- ◆个人安排：相当于目前正在实施的周计划，由项目经理在阶段计划基础上制定项目小组成员下周每天的工作计划，并在每周一堆上周的计划进行跟踪
- ◆进度计划制定的原则：充分考虑每个工作人员的实际能力、计划的合理性和实用性、考虑计划任务的关联性、充分利用project工具

# 计划编制方法

---

## ◆计划编制技术和方法

- 关键日期表
- 甘特图
- 里程碑图
- 关键线路法
- 计划评审技术等

## ◆计划编制的统筹协调

- 建立网络图，确定各项任务之间的相互依赖关系，发现可能遗漏的任务，并进行直观的展示
- 为任务分配时间，
- 确定项目组成员可支配的时间
- 为任务分配资源并进行平衡
- 确定管理支持性任务
- 对影响计划的所有内容，包括个人进度计划、项目里程碑、累积的任务汇总、人员阶段汇总，累积的资源汇总、任务分配单等，进行汇总和平衡，解决冲突
- 重复以上过程，并记录下来



# 项目进度计划的主要内容

---


## ◆编制项目计划就是纪录项目进度安排的成果

- 项目进度计划基本内容
- 项目进度的表达形式
  - 带有日历的项目网络图
  - 时间坐标网络图
  - 条形图或者甘特图
  - 里程碑时间表
  - 项目任务书
- 项目进度计划的补充和细节说明
- 进度管理计划
  - 主要说明何种进度变化应该给予处理，可以是正式的或者非正式的、详细的说明或者基本的框架，是总项目计划的辅助说明

# 编制计划的注意事项

---

- ◆计划中除了定义工作之外，还应该为管理者提供了根据计划定期评审和跟踪项目进展的基础
  - 制定进度安排，确定足够多的检查点，以便进行定期跟踪
  - 每个阶段都确定几个检查点，每个检查点必须代表完成了某些准确定义的工作，因此检查点是具体的、可度量的



**如何维护计划实现?**

---

## ◆项目监控

# 项目监控实现时注意事项

---

- ◆项目监控的**目的**是通过周期性地**跟踪项目计划的各种参数**如进度、工作量、费用、资源、工作成果、风险等等，不断地了解项目的进展情况，以便当项目实际进展状况显著偏离计划时能够及时采取纠正措施
- ◆项目监控的基本**原理**是：将项目实际情况与项目计划进行对比，如果发现某些因素的偏差非常大，那么及时分析原因，给出纠正措施
- ◆项目经理**不要企图对所有的项目事务**进行监管，否则要管的事情实在太多了，最终什么都没有管好。一般地，项目监控的**重点**是：任务进度；项目费用；人员业绩；软硬件资源；项目风险

# 监控方法

---

## ◆周报

- 报告本周完成的任务 (WBS)
- 本周遇到的风险和问题
- 下一周的计划
- 本周的度量数据

## ◆分为个人的周报和项目的周报

**注意：“周”不一定是个准确的概念**

# 常用评审方法

---

## ◆周例会/正式项目评审

- 通报状态
- 识别和跟踪问题/风险
- 确定优先级
- 为问题的解决安排人员
- 创建新的承诺

## ◆正式项目评审需要客户的参与

# 项目监控发现问题时处理过程中注意事项

## ◆相关过程域

## ◆变更? -----需求、计划、承诺

计划的变更在项目范围改变或者采取了改正行动的时候发生  
需要正式的变更流程，不要轻易变更计划  
计划变更可能会引起连带的配置变更  
计划变更的时候需要控制计划的版本

需求变更使得项目的日程、成本表现超出警戒值，项目范围变化（Scope Creep），还有其他预示项目会偏离计划的征兆（Expected Turnover）（Poor Quality Indicator）

项目计划的变更可能会带来对内和对外承诺的变更  
对外承诺的变更需要经过高级经理评估和批准  
对外承诺变更需要征得外部风险承担者的同意  
对内承诺的变更也需要所有相关人员的评审和批准

# 项目监控发现问题时处理过程中注意事项

## ◆风险跟踪

- 通过周例会进行跟踪
- 重新评估持续的风险
  - 风险的可能性
  - 风险造成的危害
- 识别出新的风险并进行评估
- 更新风险缓解方案
- 将已经发生的风险改成“问题”



# 项目监控实现过程中注意事项

---

- ◆跟踪与监控的手段不仅仅是周报和例会
- ◆项目中应该建立明确的沟通机制，什么问题应该与谁沟通，沟通形式应该是什么样的
- ◆项目中的问题，承诺和变更在可能的情况下应该随时得到沟通
- ◆相应的沟通结果应该得到恰当的记录
- ◆例会中应该重申日常沟通中牵涉的重要内容
  - 与监控中的沟通要求

# 项目监督与控制小结

---

◆计划了什么事情就跟踪什么事情

◆ 三种主要跟踪方式:

- 日常交流、工作日志或工作周报、项目月报、里程碑状态报告、阶段报告
- 项目组级或公司级的技术例会
- 里程碑/正式项目评审

---

## ◆过程和产品质量保证

# PPQA的特定活动

---

- ◆为裁剪标准过程提供支持和管理;
  - ◆为Delphi估算提供帮助;
  - ◆帮助定义或修改工程过程或过程模型;
  - ◆为工作产品确定验收标准;
  - ◆为项目设置性能目标;
  - ◆参与计划的准备;
  - ◆检查计划（如：项目计划、CMP、Test Plan）是否完整。
- } 客观检查  
的基础

# 实现中常用方法

---

- ◆ PPQA 组织和协调质量活动识别项目中的缺陷，包括：
  - 检查 (Inspection)
  - 走查 (Walkthrough)
  - 同行评审 (Peer Review)

# 产品质量保证活动

---

- ◆在项目策划期间PPQA同项目经理和技术负责人一起**评审**项目质量目标及度量参数
  - 使用G-Q-M方法（Goal-Question-Metrics）来确定有效的度量
  - 初期应尽量保证数据分析的简单性
- ◆**收集和分析**项目的产品质量数据
- ◆**建立**数据存储和采集机制

# 过程评审

---

- ◆在PPQA计划中确定要进行的评审活动,并且得到项目组的承诺
- ◆提前安排评审的时间
  - 应提前通报项目经理, 双方要就日期和时间达成一致
- ◆需要准备评审的检查单
  - 评审的对象是项目已定义的过程
- ◆评审活动要以检视工作产品和访谈作为主要信息来源
  - 如: 评审的核心是项目的估算过程, 则应和参加估算的人员进行访谈, 讨论他们执行这个过程步骤, 并检查文档化的结果
- ◆记录评审结果

## PPQA检查单举例（WBS）

---

检查项	是/否/不适用	注释
是否为项目定义了工程过程模型？		
是否对每个阶段的入口/出口标准、活动进行了文档化？		
是否建立了项目的WBS？		
WBS是否识别了要执行的活动和要开发的产品？		
WBS是否反映了项目所选择的过程模型？		



# 结束过程评审

---

- ◆将发现的问题汇总到报告中，并与项目经理及与问题有关的项目组成员讨论
  - 目的是澄清发现的问题，并就如何解决这个不相符的问题达成一致意见
  - 尽可能在项目级解决不一致的问题
- ◆对于解决一个不符合项，可能的选择办法：
  - 改正现在的做法使之与过程一致
  - 修改过程使它适应项目
  - 基本上接受不一致项，允许特殊情况出现的异常
- ◆指定问题解决人并跟踪到结束

# 工作产品审计

---

- ◆在整个项目周期中，PPQA负责人必须定期审计工作产品，以检验他们是否和可用的标准（如：设计文档模板）相符，是否注意格式和内容，以及必须作的质量检查点是否已经完成
- ◆对于重要的工作产品在置于配置管理以前必须经过PPQA的审计
- ◆其他的工作产品可以由PPQA通过抽查的方式进行审计

---

◆典型的需要审计的产品有：

- 项目计划
- 需求规格说明书
- 设计文档
- 源代码
- 测试计划
- 用户手册

# 完成工作产品审计

---

- ◆ PPQA应该在产品质量检查点结束后进行审计工作
  - PPQA负责人应监控项目开发活动的进展情况，以便预测在指定的时间实施审计工作
  - 如果PPQA负责人要参与产品的质量评审，他在此时就可以做产品审计
- ◆ 项目组成员应向PPQA提交交付产品和产品批准表，以此表示产品的质量评审已完成：
  - 通过检查产品并记录发现的问题
  - PPQA要检验该产品的质量评审是否完成
  - 不论是否通过审计，都应及时将结果反馈给项目组成员
- ◆ 对于产品审计发现的问题处理流程和过程中的NC处理采取相似的处理方式

## 处理不符合项

---

- ◆ 如果不符合项不能在项目级得到满意的解决，PPQA必须要把这些不符合项上报给负责该项目的高层经理
  - PPQA必须具备判断哪些问题要上报以及如何上报的技巧
- ◆ 已经上报给高层经理的不符合项，要记录到问题日志中，要和项目经理定期评审这个问题日志，直到不符合项得到解决。

---

## ◆度量与分析

# 实际中应用度量实例

---

- ◆1. 明确商业目标 (G)
- ◆2. 确定关键成功要素 (Q)
- ◆3. 定义一组度量元 (M)
- ◆4. 收集、验证和保存数据
- ◆5. 数据分析
- ◆6. 纠正和改进

# 一个GQM的例子

---

◆目标（G）：提高劳动生产率

◆问题（Q）：

- 花在返工上的时间是多少？
- 开发人员是否花费了太多时间在支持和管理活动上？
- 平均劳动生产率是多少？
- 劳动生产率是否和开发小组的经验相一致？



---

### ◆度量 (M)

- 每个开发人员花费在返工上的平均人时数
- 每一活动每人时的SLOC
- 对软件开发工具可供使用的预算百分比
- 对支持人员可供使用的预算百分比
- 每一活动管理或支持任务花费的人时比例
- 每个经理对开发人员的比率

# 实际中度量与分析应用的问题

---

- ◆ “如果你不能度量它，就说明你你对它的了解还停留在一个很浅的层次上”
- ◆ 度量，说容易也容易，说难也难
  - 一个致命问题：数据真实可靠吗？
- ◆ 度量的指标分两个层面
  - 对项目的**监控能起作用的**
  - 对**未来的项目能起作用的**
- ◆ 度量的工作也分两级
  - 组织级度量
  - 项目级度量

# 项目中度量全过程

---

- ◆目的：开发和维持度量能力，以便支持对管理需要的信息需要
- ◆度量策划怎么做？
  - 依据公司度量目标，同时考虑项目差异性
  - 依据项目度量目标识别度量项（度量数据）
  - 考虑度量数据采集的来源、频率以及分析方法
  - 考虑度量数据的汇报机制和存储机制
  - 编写度量子计划

---

## ◆建立度量目标

- 建立和维护根据信息需要和目标来确定的度量目标
- 公司有哪些目标？
- 与进度、目标有关的目标。
- 减少工作量投入，减少成本
- 减少需求变更的影响（变更工作量不能超过总工作量的30%，40%）
- 交付功能100%实现
- 尽可能修复发现的所有缺陷

---

## ◆指定度量项

- 根据度量目标指定度量项
- 直接度量项
  - 计划成本和实际成本之间的差距
- 间接度量项（衍生度量项）
  - 成本超支，bug修复率，工作量偏差率
- 度量计划的目的：策划项目数据的依据，并指导收集和分析活动的进行

## ◆数据收集和存储规程

- 怎样获得数据，例如度量数据的采集来源和频率。
  - 采集来源：从里程碑、周报、周会中获得
  - 采集频率：每周、每里程碑
- 怎样存储数据。
  - 储存到云上或服务器上

---

## ◆度量数据分析规程

- 怎样沟通和分析度量数据
- 采用什么方法和工具进行分析。
  - 使用Excel饼状图，趋势图，直方图，鱼骨图等

## ◆数据采集和分析

## ◆通报结果

- 每周给所有相关干系人通报度量和分析活动结果

---

## ◆配置管理

# 配置管理概念

---

- ◆软件配置管理的目的是在项目的整个软件生命周期内建立并维护软件项目产品的完整性，记录并报告配置的状态，验证配置项的完整性和正确性。
- ◆配置管理的目的是建立和维护工作产品的完整性，使用配置项，配置控制，培植状态统计和配置审计
- ◆通过配置标识、配置控制、配置状态报告和配置审计等手段，建立和维护工作产品的完整性



# 配置管理主要任务

---

- ◆识别在指定时间形成基线的产品配置
- ◆控制配置项变更
- ◆由配置库构建和发布产品
- ◆提供精确的配置状态
- ◆维护在整个软件生命周期中配置的完整性和可跟踪性

# 基本概念

---

## ◆配置：

- 是在技术文档中明确说明最终组成软件产品的功能或者物理属性，它包括了即将受控的所有产品特性、内容及其相关文档，而且包括软件版本、变更文档和软件运行的支持数据，以及其他一切保证软件一致性的组成要素

## ◆配置项--凡是纳入配置管理范畴的工作成果统称为配置项(CI)。配置项主要有两大类：

- 属于产品组成部分的工作成果，例如源代码、需求文档、设计文档和用户说明书等等即目前公司采用的配置项。
- 在管理过程中产生的文档例如各种周报、监控报告等等，这些文档虽然不是产品的组成部分，但是值得保存，也就是我们的数据项。

每个配置项的主要属性有：名称、标识符、版本、作者、日期等。所有配置项都被保存在配置库里，确保不会混淆、丢失。配置项及其历史记录反映了软件的演化过程

---

## 配置项特点

---

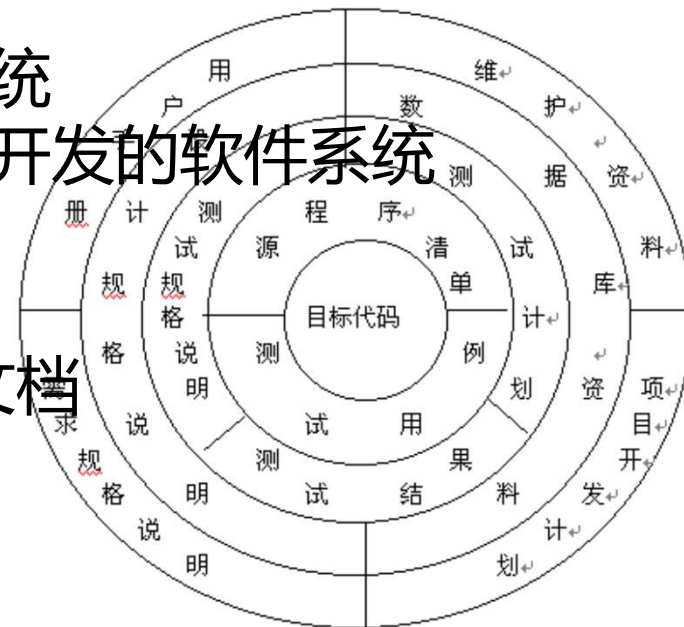
- ◆它会被两个或两个以上的项目 成员共同使用。
- ◆它会随着项目的开展而发生变化。
- ◆对项目重要的工作产品。
- ◆一些工作产品之间的关系非常紧密，一个变化其他的就会受到影响。
- ◆配置项本身的变化可以使用“版本管理”对其进行控制

# 软件工作产品实例

可作为SCM控制对象——配置项/单元的软件工作产品

- 过程文档（如计划、标准或规程）
- 软件需求
- 软件设计
- 软件代码单元
- 软件测试规程
- 软件测试活动所需的软件系统
- 为交付给顾客或最终用户所开发的软件系统
- 编译程序
- 其它支持工具

因此主要可以分为：软件产品和文档



# 配置管理任务

---

◆基本目标之一---为客户提交高质量的软件产品，包括：各种资源以及构成资源或者目标代码的目标文件，还包括以这些文件来构建工作系统的脚本以及相关的文档。

- 在项目中，资源和文档通常以**很多独立文件**的方式来维护，当项目进展时，文件发生了改变，产生了**不同的版本**
- 即使将项目的各个部分组合起来，构建成系统，也是很困难的任务，怎样保证合并的事源程序的正确版本以及没有遗漏任何程序？怎样保证传送的文档的版本是正确的，该版本和最终交付的软件是一致的？

因此必须正确跟踪软件开发过程中的各种中间产品、版本以及软件产品的版本，这个活动不是由常规的软件管理或者开发过程来处理的，而是被看成CM (Configuration management)的一部分

# 配置标识—例子1

---

例1: 文档的标识

标识规则:

VVV-WWW-BBB-XXX.NN.MM

含义:

VVV - 系统识别符

WWW - 分系统识别符

BBB - 文档类别

XXX - 版本标识

NN - 发布号

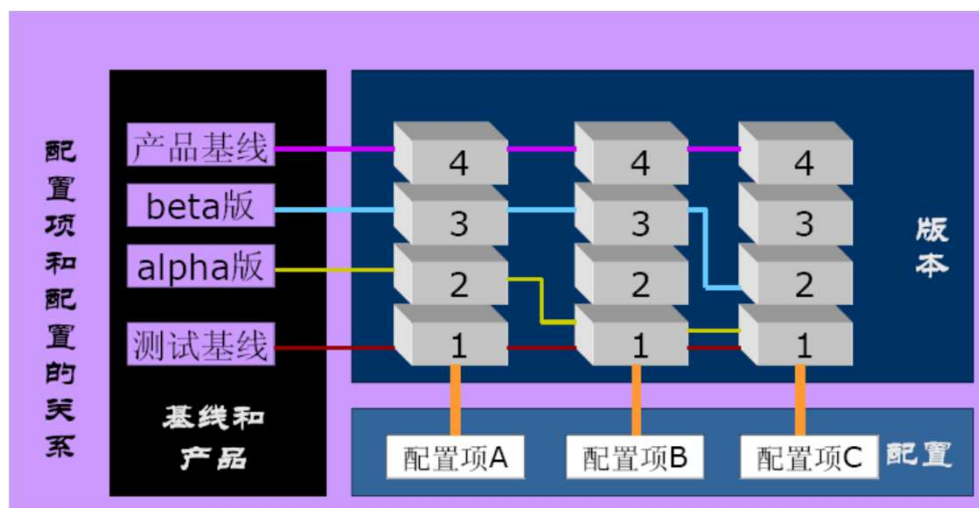
MM - 修订标识

# 配置项实例

阶段名称	配置项
需求 (RD)	原始需求记录表
	项目软件需求说明书
策划 (PP)	PDP 说明
	项目进度计划
	项目总体计划
	配置管理计划
	项目级质量保证计划
实施 (PI)	项目阶段工作报告

# 软件配置管理

- ◆软件配置管理是指通过执行版本控制、变更控制等规程，以及使用合适的配置管理软件，来保证所有配置项的完整性和可跟踪性
- ◆配置管理是对工作成果的一种有效保护
- ◆配置管理是否有效取决于三个要素：人、规范、工具





# 软件配置管理活动

---

## ◆可归结为4个主要功能

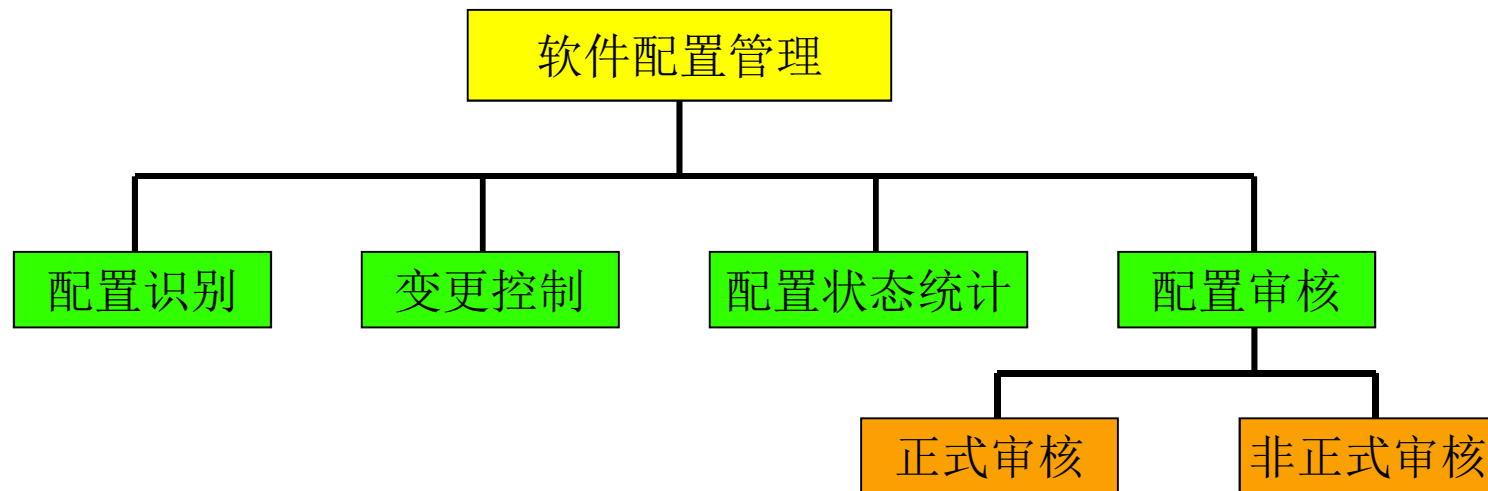


图 软件配置管理的功能

# 执行配置控制

---

## ◆执行配置控制活动主要有两个：

### ➤ 涉及管理程序（和文档）的状态转移

- 状态转移包括在状态变化时从一个目录转移到另外一个目录下，当进入了修改时产生新版本，并确保每个程序（文档）经过其生命周期并达到其基线，一旦所有的配置项都进入了基线，软件就可以作为产品使用了

### ➤ 涉及管理必须实现的需求变更

- 实现需求变更请求的主要活动包括：接受变更请求（和影响分析）、建立跟踪机制、检出需要变更的配置项并执行变更、检入配置项、使配置项走过其生命周期

# 基线 Baseline

---

- ◆IEEE对基线的定义:已经正式通过复审核批准的某规约或者产品,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控制过程改变
- ◆基线: 软件文档或源码(或其它产出物)的一个稳定版本,它是进一步开发的基础。
- ◆基线 (Baseline) 由一组配置项组成, 这些配置项构成了一个相对稳定的整体。基线中的配置项被“冻结”了, 不能再被任何人随意修改。
- ◆经过正式评审和认可的一组软件配置项, 此后, 它们将作为下一步开发工作的基础, 而且只有通过正式的变更控制流程才能被更改, 例如设计说明书是编码的工作的基础, 它可以成为软件基线

# 基线

---

- ◆基线通常对应于项目/开发过程中的里程碑，一个产品可以有多个基线，也可以只有一个基线。
- ◆基线的主要属性有：名称、标识符、版本、日期等
- ◆基线是项目储存库中每个工件版本在特定时期的一个“快照”。它提供一个正式标准，随后的工作基于此标准，并且只有经过授权后才能变更这个标准。
- ◆对基线的更改必须遵循变更控制规程。
- ◆交付给外部顾客的基线一般称为“放行”基线，内部使用的基线一般称为“构造”基线。

# 基线作用

---

- ◆基线作用：重现性、可追踪性和报告。
- ◆重现性是指及时返回并重新生成软件系统给定发布版的能力，或者是在项目中的早些时候重新生成开发环境的能力。
- ◆可追踪性建立项目工件之间的前后继承关系。其目的在于确保设计满足要求、代码实施设计以及用正确代码编译可执行文件。
- ◆报告来源于一个基线内容同另一个基线内容的比较。基线比较有助于调试并生成发布说明。

# 基线建立时机

◆基线建立时机：一般可以在里程碑完成前（或每次迭代结束时）

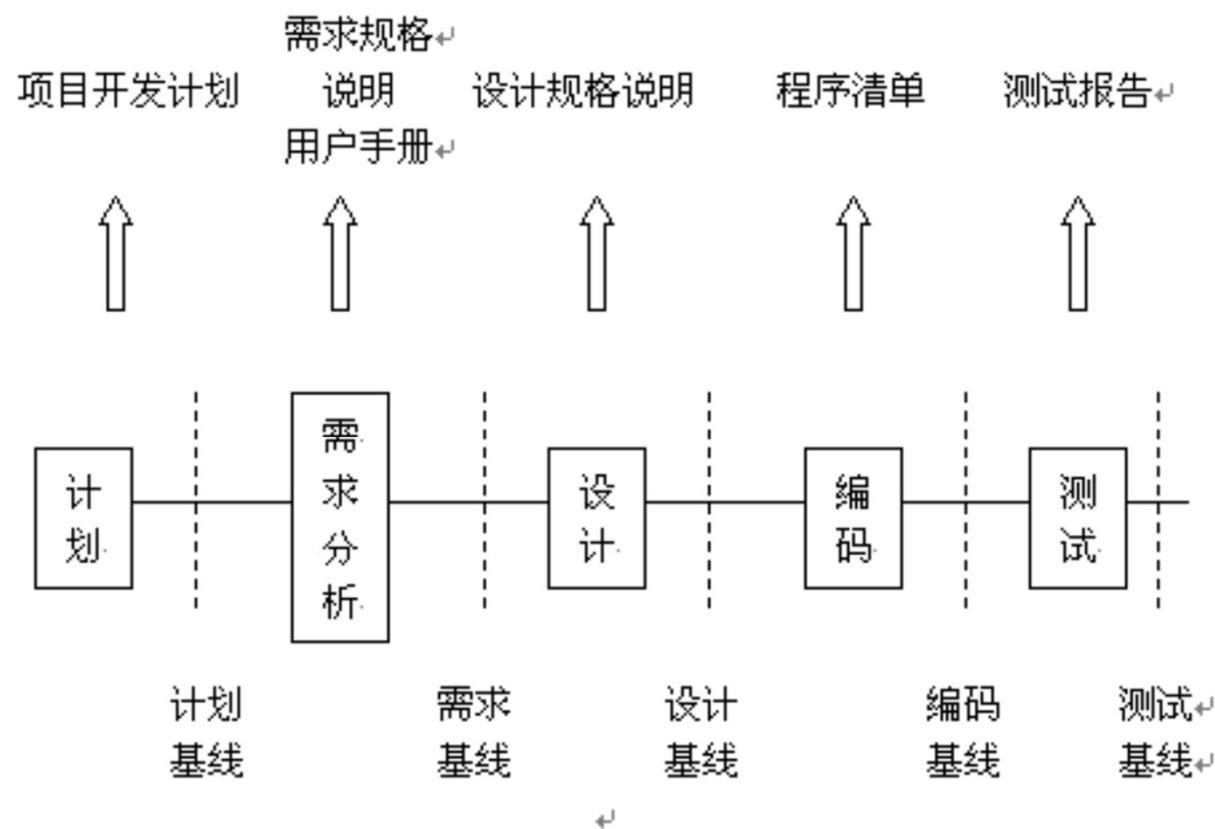


图 9 软件配置基线

## 基线建立流程（1）

---

- ◆基线是开发活动进行到一定阶段，由 P M来确认，通知配置管理员对相关的配置项的某个版本做的一个标记。对于整个项目来说，就是一个build版本，在开发活动中起到承前启后的作用，是对前一阶段开发成果的体现，后一阶段开发的起点。
- ◆文档库基线建立：在需求、计划、设计、编码、测试、实施结束后建立基线，各阶段有变更则相应建立变更后的基线。

## 基线建立流程（2）

---

### ◆开发库、测试库基线建立：

cm建库并分配权限给相关人员，开发完成后，开发人员在Release目录下提交测试安装包，CM建立基线即对源代码、安装包打tag标识。CM检出该阶段的安装包，提供给测试人员。

测试人员测试无BUG后，提交到测试库，CM对测试库的安装包及源代码打tag标识。

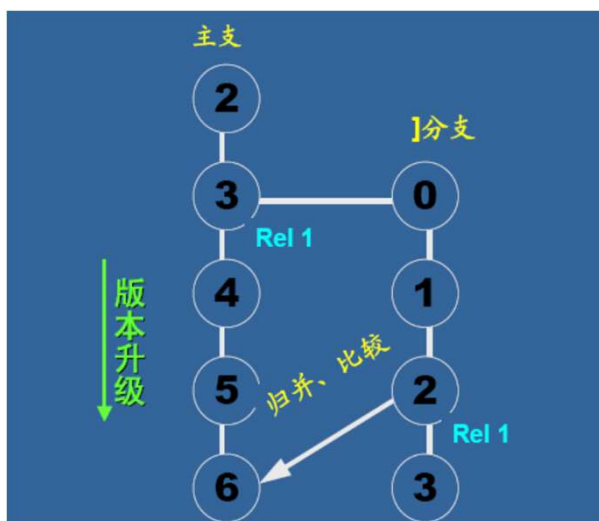
CM检出测试库的安装包、说明文档、使用手册提交到受控库。发放产品由该库检出。



# 版本控制

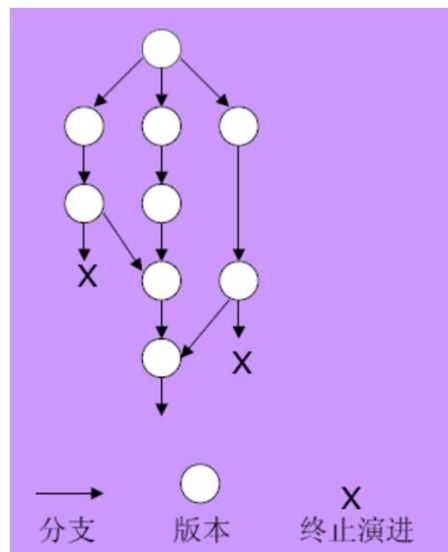
## ◆分支、合并、比较

(文件比较 - 用来比较两个或多个分支或基线中具有相同名字的文件，并识别这些不同的文件。)

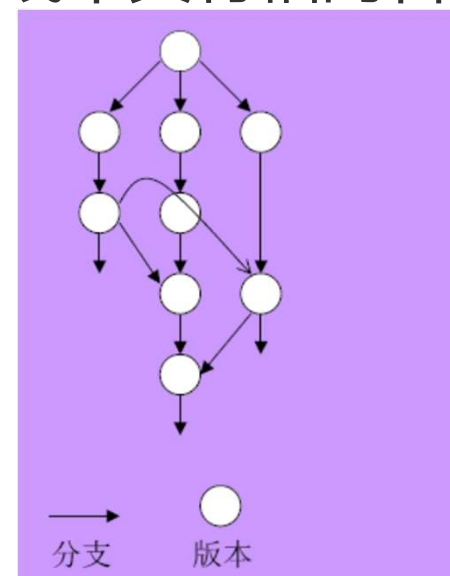


分支作用

- 缺陷修正
- 多版本
- 并行开发



- 所有合并必须合并到主分支，子分支完成合并后，不再演进
- 适用于对软件缺陷的修正工作管理



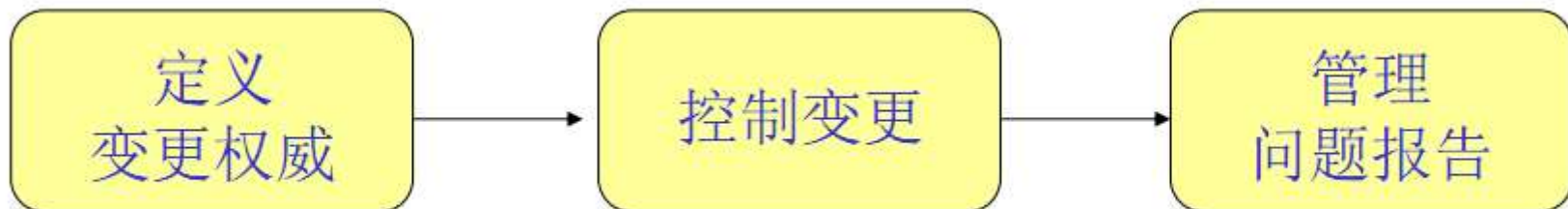
- 但是所有的变化均要反映到主分支，各个分支均可进行合并
- 适用于具有核心版本的产品
- 围绕核心版本推进各个分支的演进
- 一个子分支的演进可以通过主分支传递给其他子分支

# 控制基线的变更

---

## ◆目的

- 识别变更授权人
- 维护配置的稳定性和完整性
- 确保变更控制的有效性



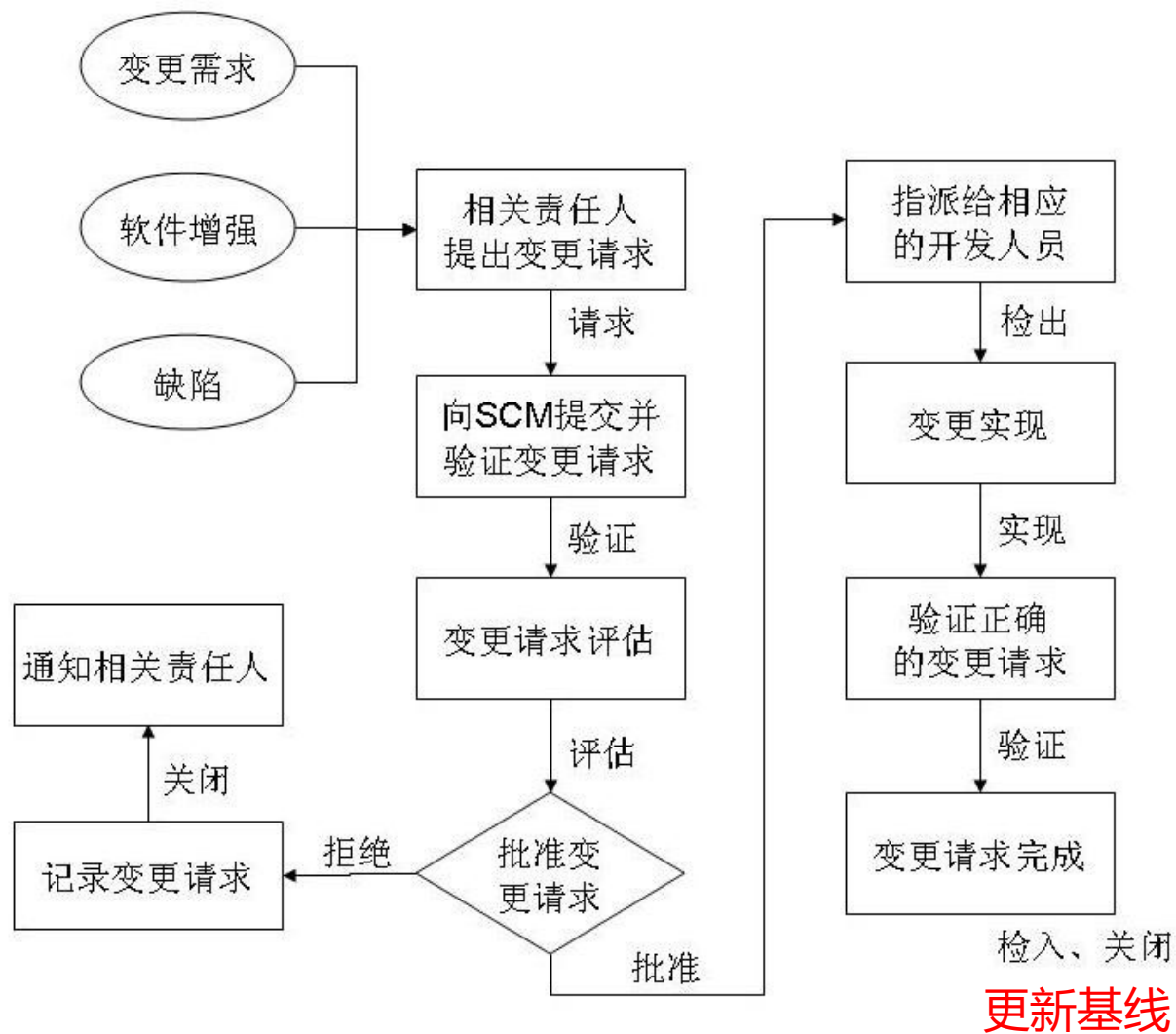
# 变更管理

---

## ◆变更管理

- 变更管理是配置管理的一个重点和难点。变更控制的目的是防止配置项被随意修改而导致混乱。
- ◆当基线库配置项需要变更时，一定要实施变更流程：
  - 变更实施前必须填写《变更申请表》，
  - 经CCB评审通过后，
  - 才能从基线库中提出需变更的配置项并实施变更。
  - 变更实施完成后，必须通过评审验证才能重新进入基线库。

# 变更管理流程



# 定义变更权威

---

- ◆正式基线（如客户需求）通常在软件配置控制委员会CCB的控制之下 (Formal)
- ◆工程过程中的基线(如设计、源代码基线)一般由项目经理或者项目技术主管进行控制(less formally)
- ◆细化管理级别，避免官僚主义

# 配置变更委员会CCB

---

## ◆CCB是授权进行正式基线变更的机构

- 例如客户需求、运行基线

## ◆职能：

- 确保变更被分类以及被评估
- 决定需要实施的变更的优先级
- 评审和批准变更
- 确保只有被批准的变更得到实施

## ◆CCB成员最好各司其职,成员可能包括：

- 项目经理，配置管理员，质量保证人员，开发人员 代表，客户代表

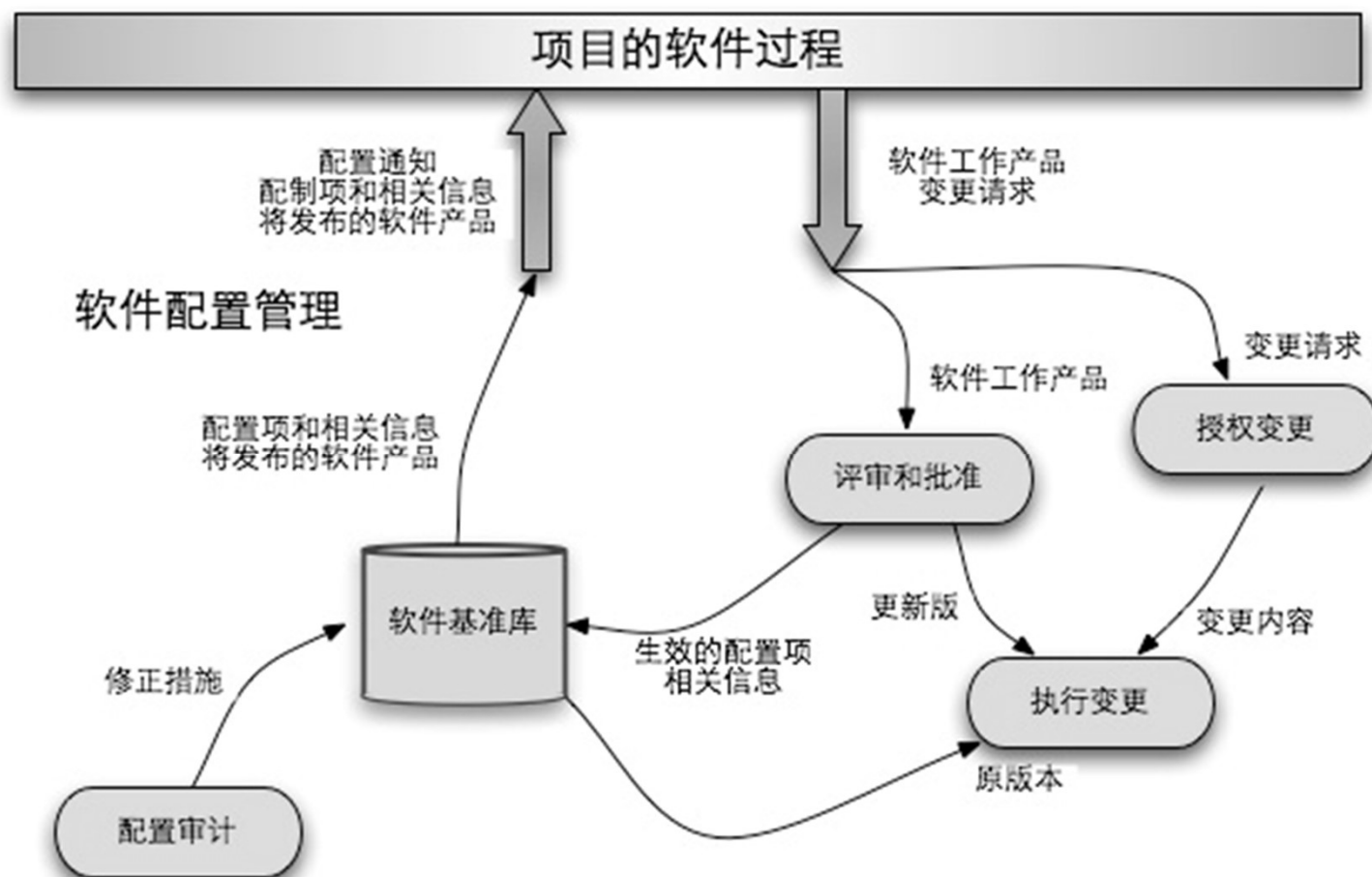
# 相关变更文档

项目名↵		变更请求标识↵	
变更请求:		变更请求人	日期↵
变更理由↵			
变更描述↵			
影响范围↵			
变更优先性考虑↵			
评估变更工作量↵			
分析与评估		分析人	日期↵
分析与评估意见↵			
审批		CCB 负责人	日期↵
CCB 审查意见↵			
变更实施		实施负责人	日期↵
变更实施情况↵			
质量保证审查		QA 负责人	日期↵
审查意见↵			
配置管理审查		CM 负责人	日期↵
审查意见↵			

变更申请考虑因素:

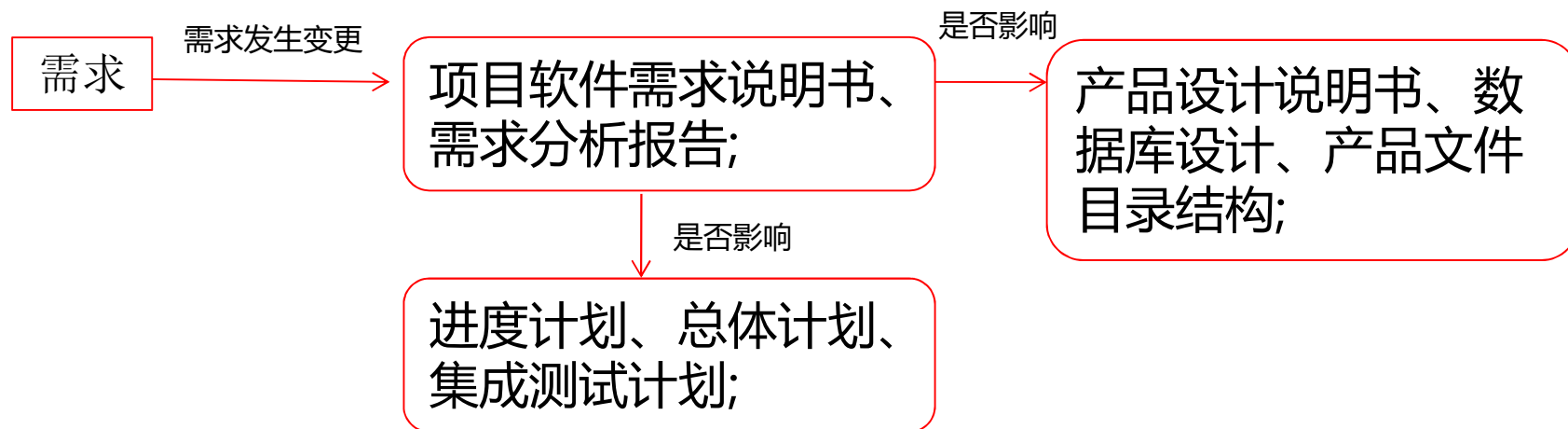
- 规模大小
- 是否有备选方案
- 变更复杂度
- 严重性
- 进度
- 成本
- 测试

# 实现中的配置管理基本流程





# 变更会涉及到的配置项分析实例



---

## ◆ 供应商协议管理

# 采购流程

---

- ◆确定采购的类型
- ◆选择供应商
- ◆签署协议
- ◆执行协议
- ◆验收
- ◆接收