

# CMMI专题培训

实用软件度量

赛柏科技

① 初始级

② 已管理级

③ 已定义级

④ 定量管理级

⑤ 优化级

# 实用软件度量

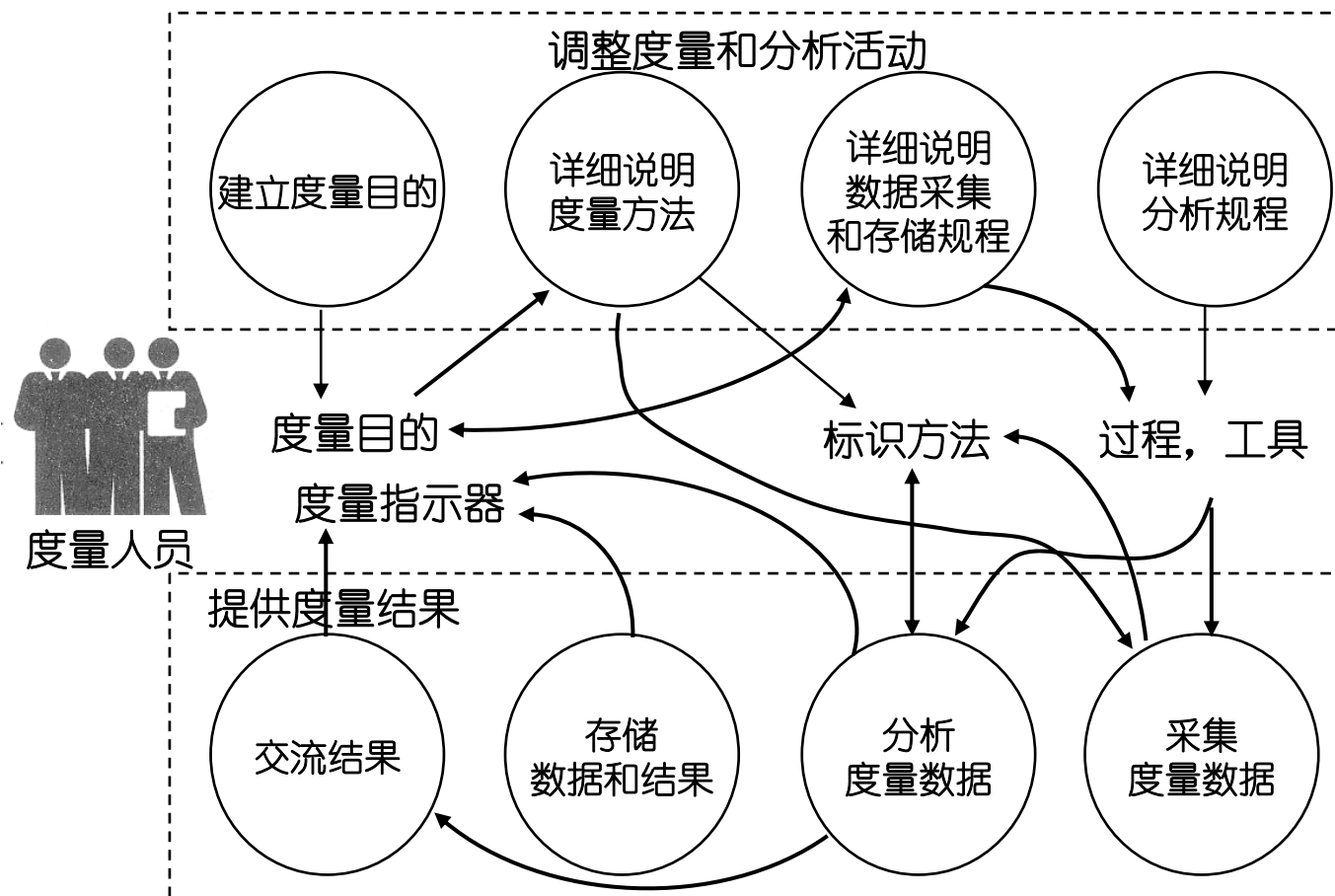
# 纲要

---

- 一. 概述
- 二. 选择度量
- 三. 定义度量
- 四. 建立度量库

# 一. 概述

# 度量和分析PA图示



# 度量的目的

- 理解——获得对过程、产品、资源等的理解；是评估、预测和改进活动的基础。 定量的理解才是对事物本质的了解，真正“心中有数”
- 预测——通过建立预测模型，进行估算和计划。 历史数据能够帮助我们预测和计划
- 评估——产品的质量、过程改进的效果等。对趋势的分析可以使我们找到问题出在哪里
- 改进——根据得到的量化信息，确定潜在的改进机会。度量本身不会改进过程；但它为我们提供了对计划、控制、管理和改进的可视性

软件度量让你知道什么时候该哭，什么时候该笑

# 度量活动中的角色和职责

高层管理者



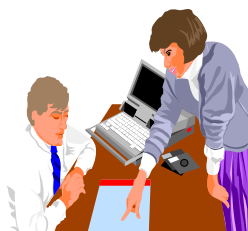
- 确定高层的目标
- 使用度量结果做组织级和企业级的决策

项目经理



- 确定和管理项目的问题（**Issues**）
- 使用度量结果做出整个项目的决策

度量人员



- 定义、裁剪度量，用来满足需要
- 收集和分析度量数据，并报告结果

开发团队



- 提供度量所需要的数据
- 在软件开发实践中利用度量结果

# 度量是有成本的

- 度量是项目管理的重要手段，但度量是需要成本的
- 度量的成本(数据收集和分析)
  - 小于项目总成本的3%
- 管理成本通常估计为项目的总成本的**10-15%**



# 可度量的实体和属性

- 可度量实体是指可度量的事（概念的）物（物理的）
- 可度量属性是指可度量实体某些方面的特征

## 人

- 身高
- 体重
- 性别
- 国籍

## 软件

- 代码行
- 功能点数
- 复杂度
- 保留缺陷数

## 评审

- 参加人数
- 评审时间
- 被评审工作产品的规模
- 发现缺陷数

## 缺陷

- 数量
- 种类
- 来源

## GQM 方法——一种确定（选择）度量的途径

- GQM (Goal-Question-Metric)
- GQM的主要步骤：
  - 确定度量的目标
  - 提出能够满足目标的问题
  - 确定回答问题所需要的度量

# GQM法举例

---

- 目标：改进项目计划
  - 问题1 项目进度是否发生了偏差？  
度量：里程碑实际到达时间  
任务包完成的数量
  - 问题2 项目工作量是否发生了偏差？  
度量：各种工作量

## GQM方法的优点

---

- 易于操作
- GQM便于识别所需的度量
- 便于确定收集数据的用途
- 使我们明确所收集的数据应该如何分析、解释

# 实施软件度量的前提条件

- 管理者支持
- 全体员工的参与
- 统一的定义
- 有效的度量过程（制定好的度量规程和度量计划）
- 实用的度量工具



## 二. 选择度量

# PA RM中的度量需求

根据度量结果来判断管理分配到的需求活动的状态

- 度量的例子有：
  - 每一个分配到的需求的状态
  - 对分配到的需求的修改活动
  - 累计对分配到的需求的变更的数目，它包括如下几方面修改的总数：提出的修改，未处理的或处理后尚未解决的修改，赞同的修改以及合并到系统基线中的修改

# PA PP中的度量需求

根据历史项目度量结果进行估计，对本项目的度量判断软件计划活动的状态

- 度量的例子有：
  - 估计过程需要历史项目度量数据：软件规模、工作量、进度（里程碑、任务进展）、成本、资源、风险
  - 与其计划相比，在软件项目计划活动中工作的完成情况、花费的工作量以及支出的费用



# PA PMC中的度量需求

根据度量结果与计划相比较，实施项目监控，并用于判断软件跟踪和监控活动的状态

- 度量的例子有：
  - 软件规模、工作量、进度（里程碑、任务进展）、成本、资源、风险
  - 在进行跟踪和监控的活动中所花费的工作量和其它资源
  - 对软件开发计划的修改活动，它包括对软件工作产品规模的估计、对软件成本的估计、对关键计算机资源的估计以及对进度等的修改

# PA SAM中的度量问题

度量的目的是为了判断供应商合同管理活动进行的状态

- 度量的例子有：
  - 与其计划相比，管理供应商合同活动的成本
  - 与其计划相比，分包产品实际交付的日期
  - 与其计划相比，主承包商交付到分包商的实际日期

# PA PPQA中的度量需求

度量的目的是为了判断SQA活动的成本和进度状态

- 度量的例子有：
  - 与其计划相比，SQA活动完成的里程碑数
  - 在SQA活动中完成的工作，花费的工作量及支出的费用
  - 与其计划相比，产品审查和活动评审的次数

# PA CM中的度量需求

度量的目的是为了判断SCM活动的状态

- 度量的例子有：
  - 处理的修改请求的数目
  - 与其计划相比，SCM活动完成的里程碑数
  - 与其计划相比，在SCM活动中完成的工作，花费的工作量及支出的费用

# 1、按进度跟踪进展

- 定期度量活动和里程碑的实际完成情况
- 按项目计划中文档化的进度比较活动（阶段跨越的时间和里程碑到达时间）的实际完成情况
- 标识进度计划的严重偏离（根据阈值）
- 相对偏离（相对于本阶段）与绝对偏离
- 对关键路径上的活动和非关键路径上的活动，要按不同的阈值进行控制

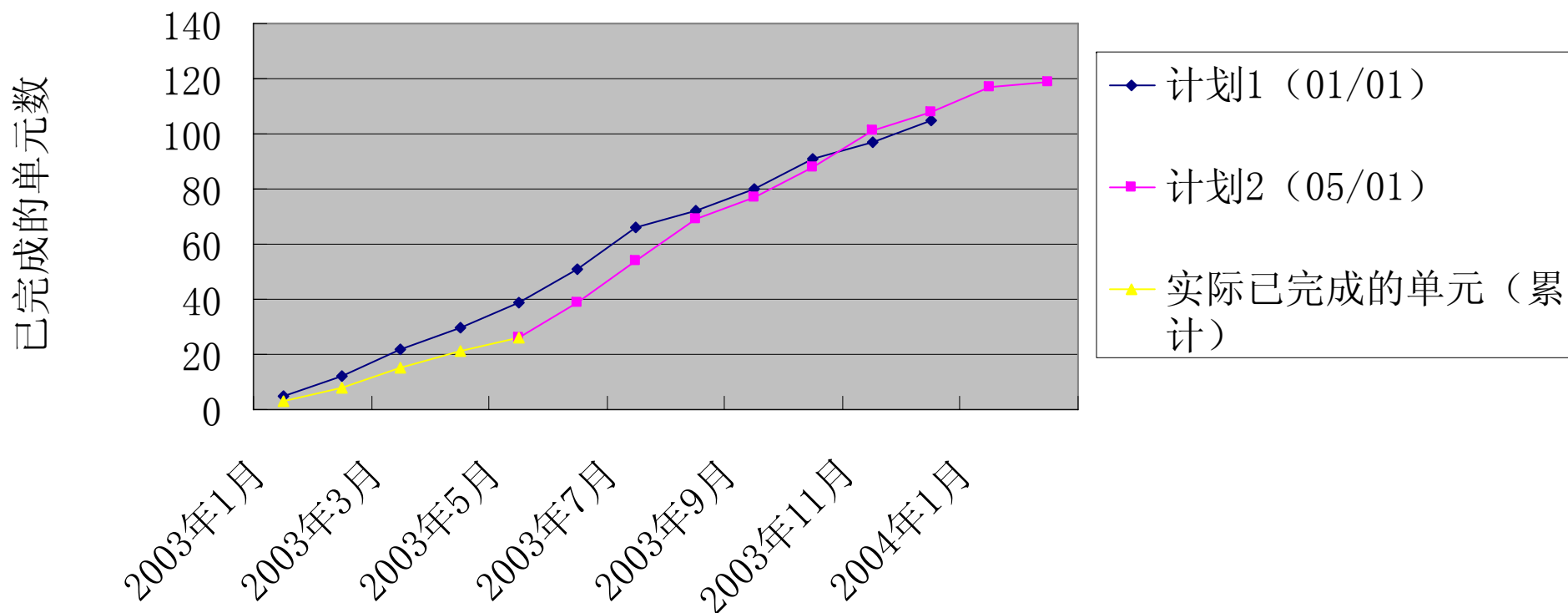
# 里程碑完成情况

- 用于评价关键开发活动和事件的进度、进展和相互依赖关系
- 右边的表对每个里程碑和总进度的延迟天数做了说明，并给出了里程碑进度差异
- 进度有超过12%的延迟（累计25天）
- 当进度差异超过阈值（如10%）时，应该调查延迟的原因。特别是应该调查在实现构造1的设计中的问题。如果必要的话，应该重新计划更现实的进度。

里程碑	计划日期	实际日期	偏差
项目开始	1/1/01	1/1/01	0
需求完成	2/20/01	2/28/01	-8
构造 1			
需求通过	3/10/01	3/12/01	-2
设计完成	4/28/01	Late	-17
编码完成	6/1/01	Started	
测试	6/7/01		
发布	7/2/01		
构造 2			
需求通过	4/2/01	4/2/01	0
设计完成	5/4/01	5/2/01	2
编码完成	6/28/01	Started	
测试	7/9/01		
发布	8/1/01		
总的延迟天数			-25
总的进度长度			212
里程碑进度偏差			-12%

# 工作单元进展 – 软件设计进展

- 按项目计划客观地评估软件设计活动的进展
- 在下图中，每个周期的计划的设计单元和已完成的设计单元使用累计线图描绘。已完成的设计单元的百分比在图的数据表中说明。该图告诉项目经理在前四个月的每个月期间设计进展已经落后原来的计划
- 在5月份时，重新计划了整个设计活动（计划2），并将这个信息添加到图表中。重计划的结果是将设计进度延长二个月。



	1-Jan	1-Feb	1-Mar	1-Apr	1-May	1-Jun	1-Jul	1-Aug	1-Sep	1-Oct	1-Nov	1-Dec
计划1 (01/01)	5	12	22	30	39	51	66	72	80	91	97	105
计划2 (05/01)					26	39	54	69	77	88	101	108
实际已完成的单元 (累计)	3	8	15	21	26							
已完成的百分比	60%	67%	68%	70%	100%							

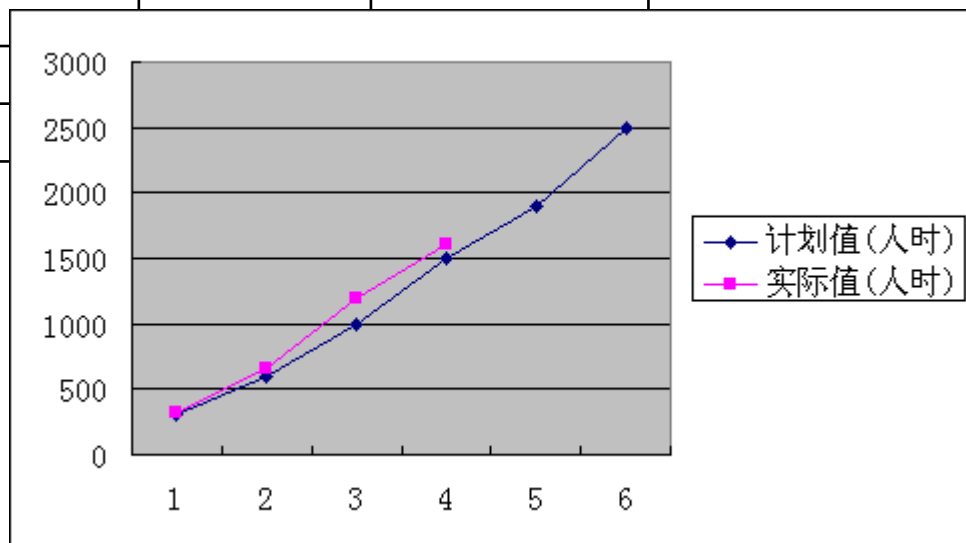


## 2、跟踪成本和工作量、人数

- 定期度量实际工作量和所花的成本及分配的人员
- 与项目计划中记录的估计和预算比较实际的工作量、成本、人员配置
- 标识与项目计划中的预算的严重偏离

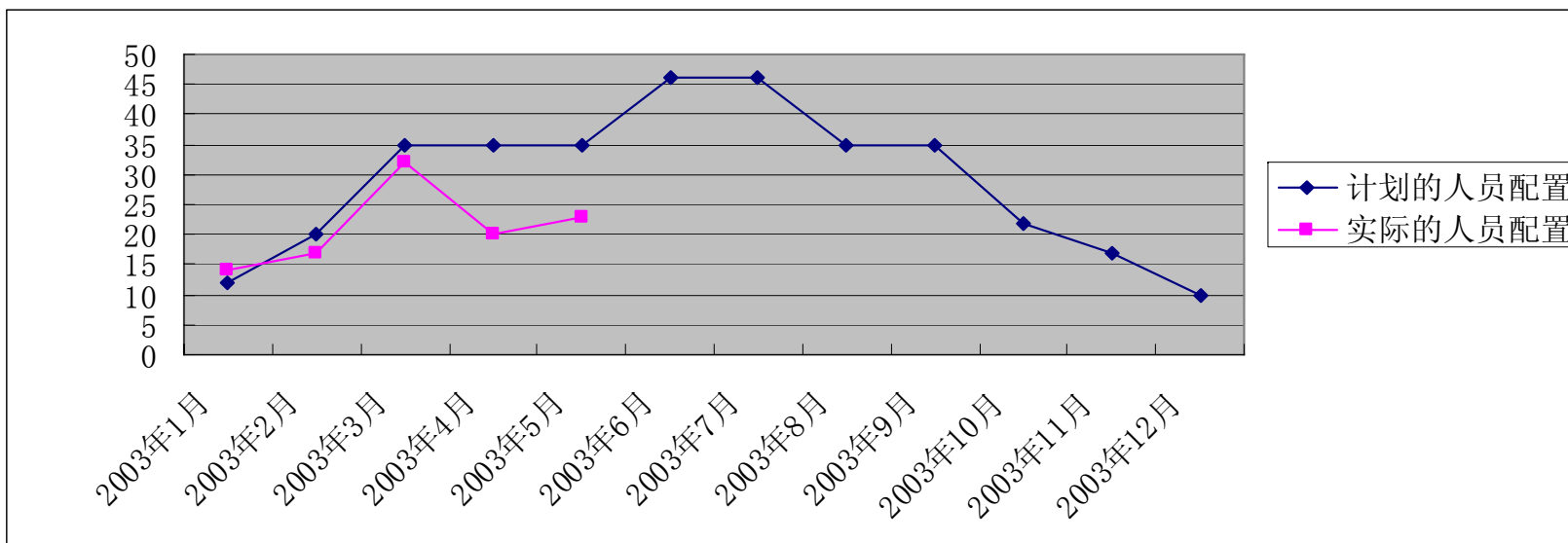
# 工作量示例

周或月或阶段	计划值(人时)	实际值(人时)	偏差%
1	300	320	7%
2	600	660	10%
3	1000	1200	20%
4	1500	1600	7%
6	1900		
7	2500		



# 人员配置情况

- 要评估是否有足够的人力资源来按进度完成项目。
- 在下图中比较了计划的人员数与实际指派的人员数，并在数据表中给出了人员配置差异（Staffing Variance）。
- 该图告诉项目经理最近一段时期的人员配置低于计划34%。
- 在12个月的项目进度表中人员配置计划最高达46人。在4月份时产生了43%的人员配置差异。在5月份时，人员配置仍然低于15%的阈值，因此需要额外的调查。



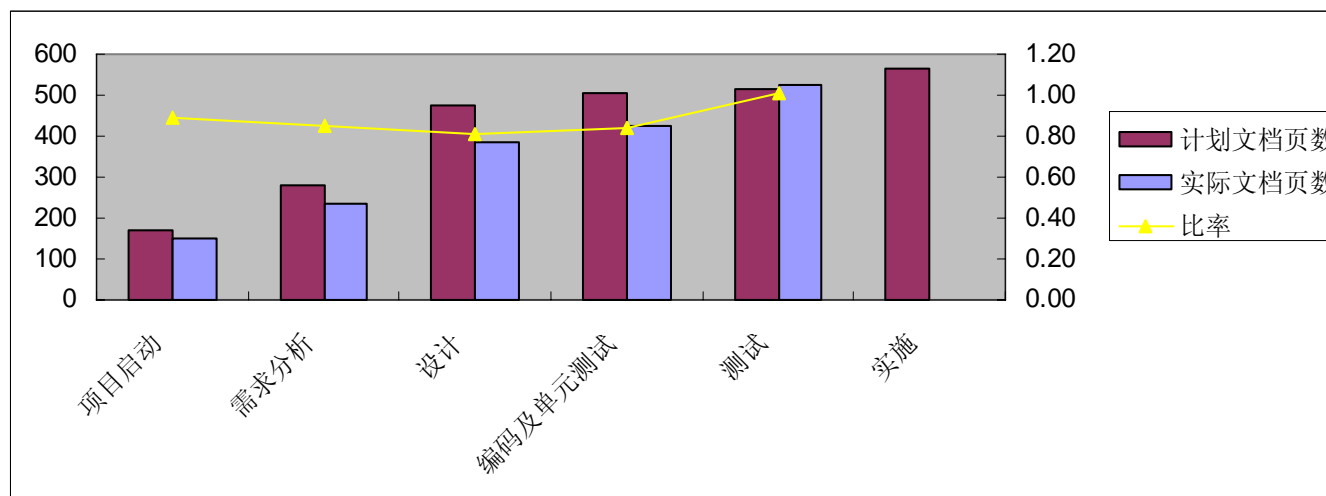
	1-Jan	1-Feb	1-Mar	1-Apr	1-May	1-Jun	1-Jul	1-Aug	1-Sep	1-Oct	1-Nov	1-Dec
计划的人员配置	12	20	35	35	35	46	46	35	35	22	17	10
实际的人员配置	14	17	32	20	23							
人员配置差异	17%	-15%	-9%	-43%	-34%							

### 3、跟踪工作产品和任务的属性

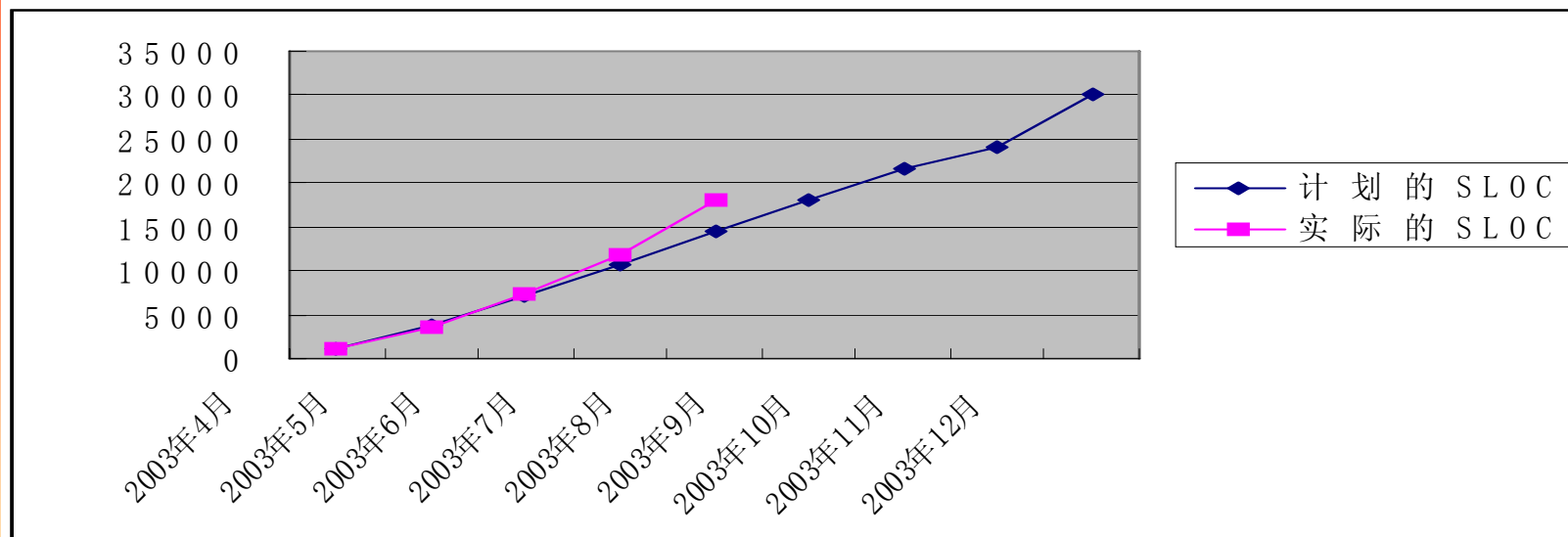
- 工作产品和任务的属性包括规模、复杂度、权重等
- 跟踪内容包括：
  - 定期度量工作产品和任务的实际属性，如规模或复杂度（和对属性的变更）
  - 将工作产品和任务的属性的实际值（和对属性的变更）与项目计划中记录的估计值进行比较
  - 标识与项目计划中的估计的严重偏离

# 文档规模

各阶段文档规模分析图							
	项目启动	需求分析	设计	编码及单元测试	测试	实施	验收
计划文档页数	168	278	476	506	516	566	576
实际文档页数	150	237	387	427	523		
比率	0.89	0.85	0.81	0.84	1.01		
数据采集时间	2003-9-16	2003-10-9	2003-10-28				
分析	NA	NA	NA	NA	NA		
对策	NA	NA	NA	NA	NA		



# 代码规模



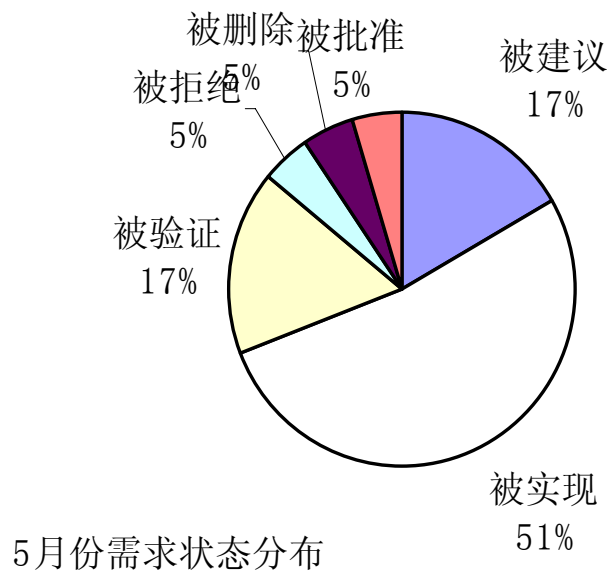
	Apr 01	May 01	Jun 01	Jul 01	Aug 01	Sep 01	Oct 01	Nov 01	Dec 01
计划的SLOC	1200	3800	7200	10800	14400	18000	21600	24000	30000
实际的SLOC	1190	3500	7250	11900	18000				
比率	0.99	0.92	1.01	1.10	1.25				

## 4、跟踪需求状态

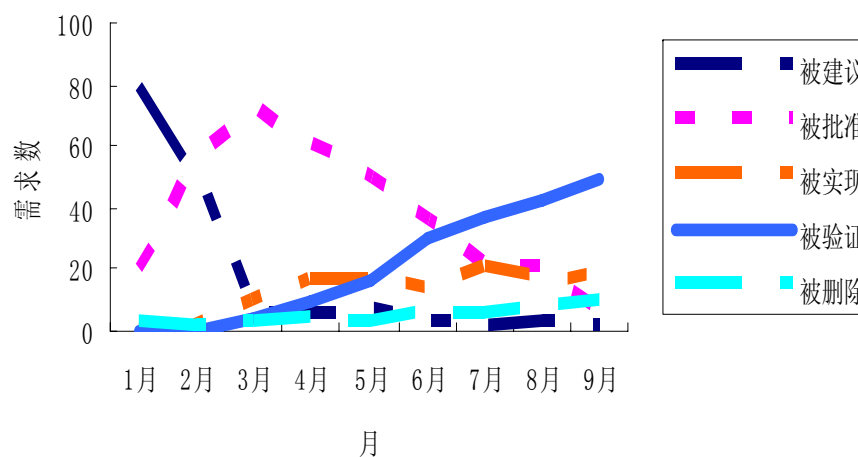
- 一个需求经提出、评审、设计、编码实现和测试，它的状态将发生变化。
- 可以将需求分被建议、被拒绝、被批准、被实现、被验证、被废除、被交付等状态
  - 被建议：该需求已被有权提出需求的人建议
  - 被拒绝：该需求被建议后，评审未通过
  - 被批准：该需求已被分析，估计了其成本和对项目其他部分的影响，而且通过评审
  - 被实现：已实现需求的设计、编码和单元测试
  - 被验证：使用所选择的方法（如测试）已验证了实现的需求，该需求现在被认为完成
  - 被废除：被批准的需求已从基线中废除，但记录了原因说明和做出废除决定的人员
  - 被交付：需求已通过客户的验收。



# 对需求状态的跟踪



前三季度需求状态分布



# 需求变更分析

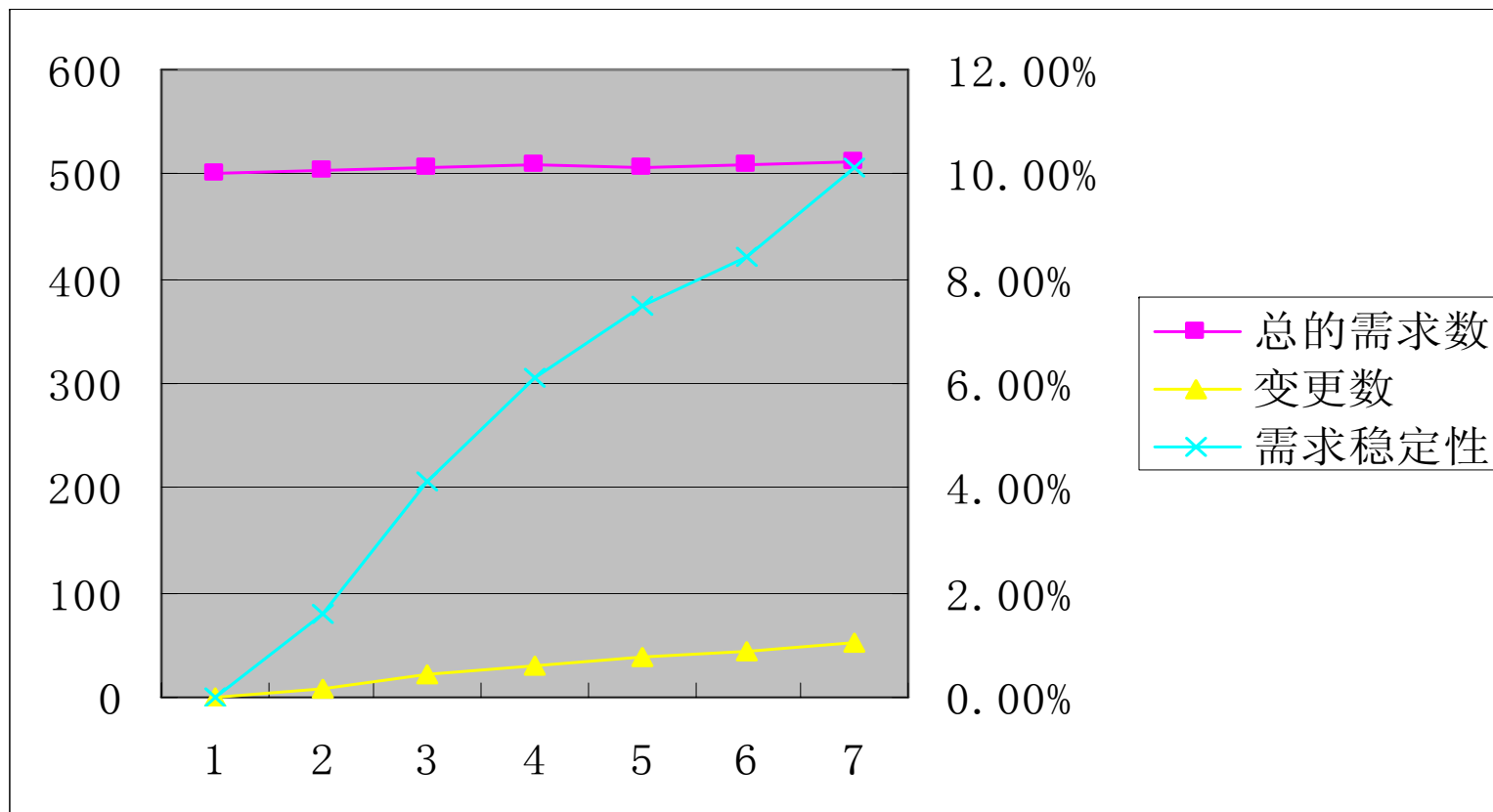
- 增加、修改或删除需求时，要进行影响分析：
  - 对开发进度的影响
  - 对发布进度的影响
  - 对人员安排的影响
  - 对成本的影响
  - 对现有约定（Commitment）的影响
  - 变更引起的风险分析
  - 对其他相关的工作产品的影响
  - 等等

# 对变更状态的跟踪

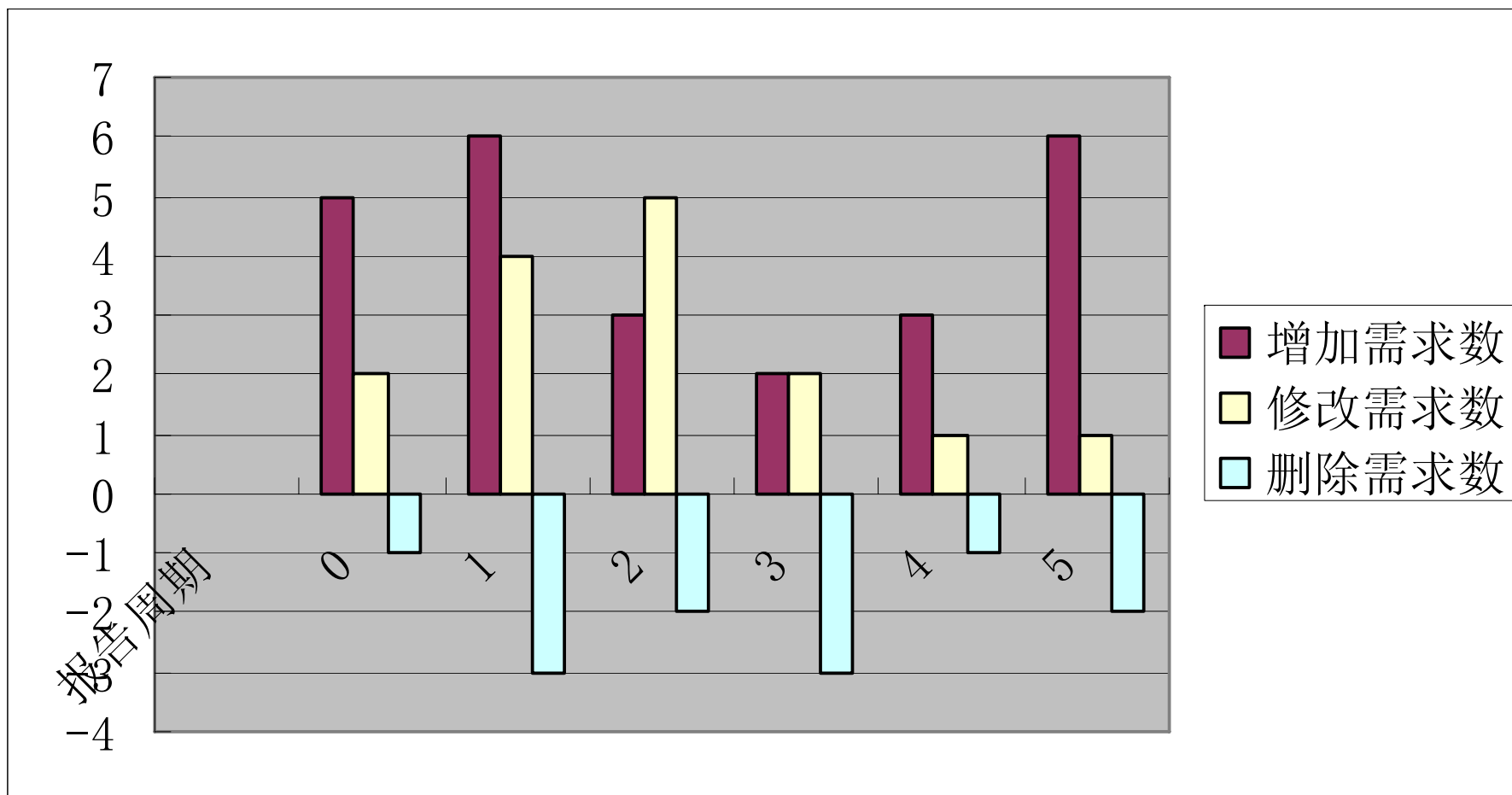
基线需求数	500					
报告周期	增加需求数	修改需求数	删除需求数	总的需求数	变更数	需求稳定性
0	0	0	0	500	0	0.00%
1	5	2	-1	504	8	1.59%
2	6	4	-3	507	21	4.14%
3	3	5	-2	508	31	6.10%
4	2	2	-3	507	38	7.50%
5	3	1	-1	509	43	8.45%
6	6	1	-2	513	52	10.14%

- 总的需求数=（基线需求数+增加的需求数-删除的需求数）
- 需求变更数=（增加的需求数+删除的需求数+修改的需求数）
- 需求稳定性=（需求变更数/基线需求数）

# 需求稳定性



# 按变更类型的需求稳定性



## 5. 跟踪缺陷数据——缺陷跟踪表

- 项目编号
- 缺陷ID
- 症状描述：发现缺陷时的系统行为
- 缺陷类型
- 严重程度
- 模块分布：缺陷是在哪个模块发现的
- 注入阶段：引入缺陷的阶段
- 发现阶段：发现缺陷的阶段
- 发现日期：发现缺陷的时间
- 发现方式：评审/审查、测试或使用等
- 计划解决日期
- 实际解决日期
- 解决用时
- 修复位置：修复缺陷的位置
- 评审/审查检查单-检查项
- 评审/审查人
- 测试用例
- 测试人

# 缺陷的度量（一）

- 缺陷按注入阶段的分布
- 缺陷按发现阶段的分布
- 缺陷按类型的分布
- 缺陷按严重程度的分布
- 缺陷按模块的分布
- 缺陷密度 = 缺陷数 / 实际规模

## 缺陷的度量（二）

- 残余数：进入该阶段时已有的缺陷数
- 注入数：在该阶段注入的缺陷数
- 清除数：在该阶段清除的缺陷数
- 剩余数：在该阶段剩余的缺陷数
- 注入率：在该阶段中每KLOC或每功能点注入的缺陷数
- 过程阶段的清除率： $\text{清除数} / (\text{残余数} + \text{注入数})$
- 总效率：总清除数/总注入数
- 审查缺陷：由审查发现的缺陷数
- 开发缺陷：由审查发现的缺陷数 + 由测试发现的缺陷数



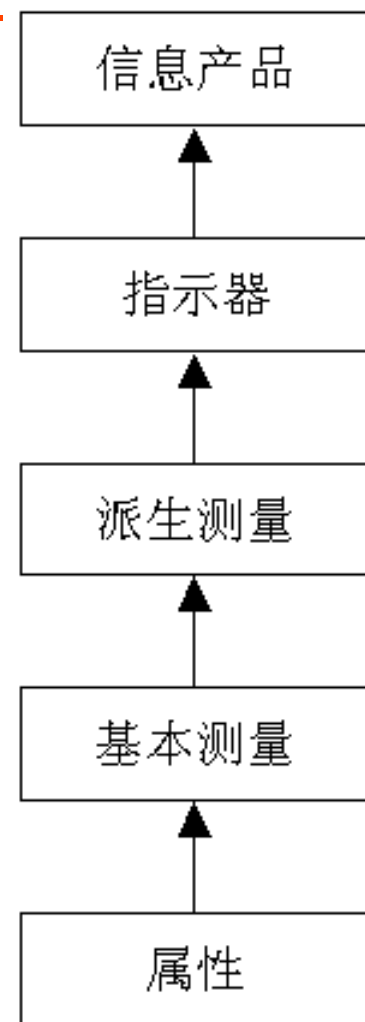
## 6、 SEI建议的最小度量元集

- 进度性能（里程碑，不一致情况）
- 费用性能（实际的与计划的对照，不一致情况）
- 工作量性能（实际的与计划的对照，分配情况）
- 需求管理（总数，增长，追踪性）
- 程序规模（源码行，页数 - 实际的与计划的对照）
- 测试性能（需要的测试，通过的测试）（功能、语句、分支、路径覆盖率）
- 缺陷数据状态（未解决和解决的问题，缺陷密度，缺陷来源）
- 过程性能（完成任务，行动项数）
- 计算机资源利用率（内存占有量，CPU占有量等）
- 管理计划项目过程的性能（对照实际进展作估计，重计划，项目总结数据）

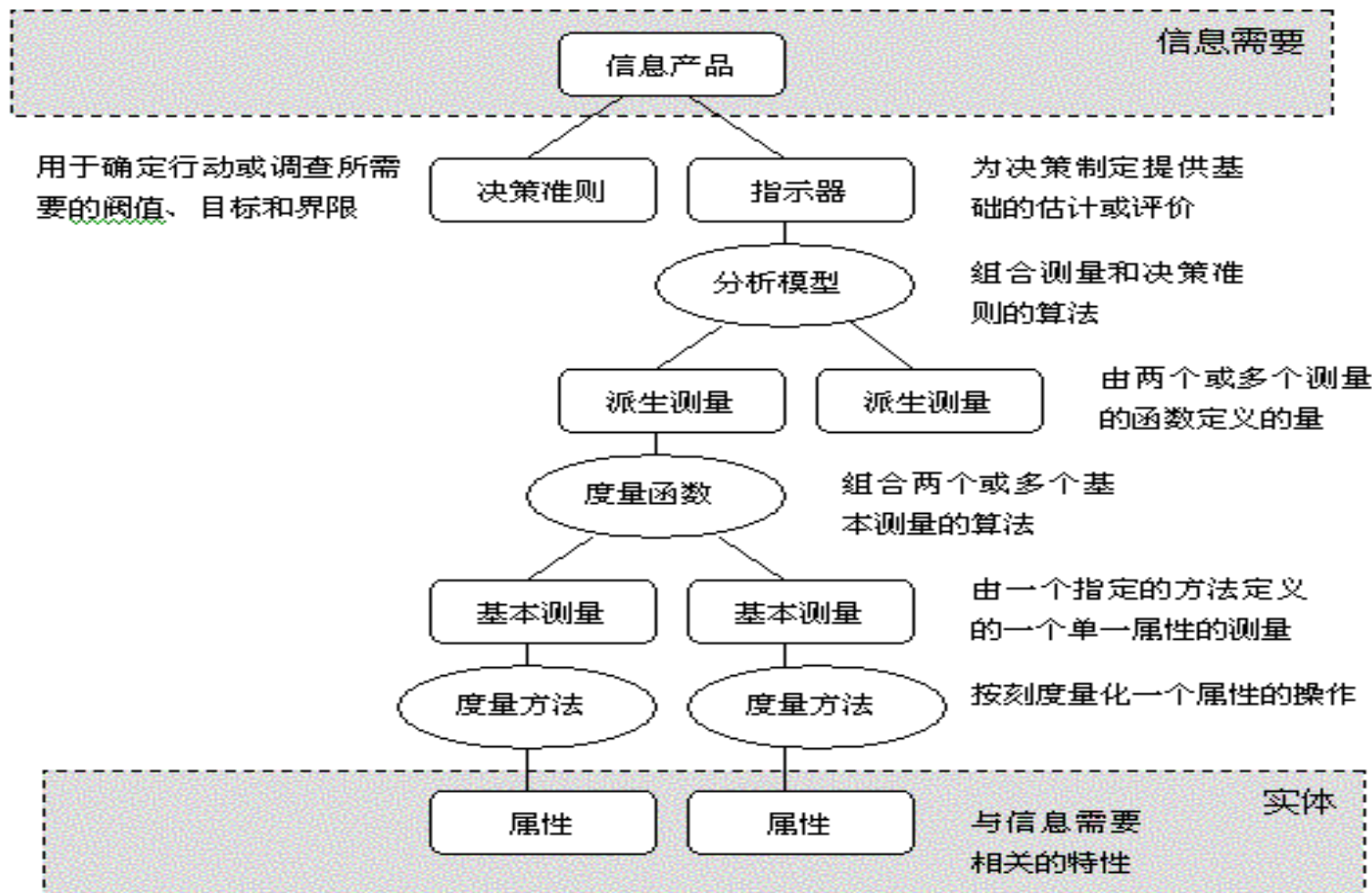
# 三. 定义度量

# 1、度量构造

- 选定了度量，还必须定义基本度量、派生度量及指示器，度量构造提供了度量的可操作定义
- 实际测量的对象包括指定的软件过程和产品的属性，如规模、工作量和缺陷数。
- 度量构造描述了相关软件属性是如何量化并转化成提供决策制定基础的指示器的。
- 一个单一的度量构造可能包括三种测量类型或层次：基本测量、派生测量和指示器。
- 度量构造设计得越合理，将要测量的软件属性与已标识的信息需要结合得就越好，项目经理也就越容易制定出可靠的客观的决策。



# 度量构造细节图

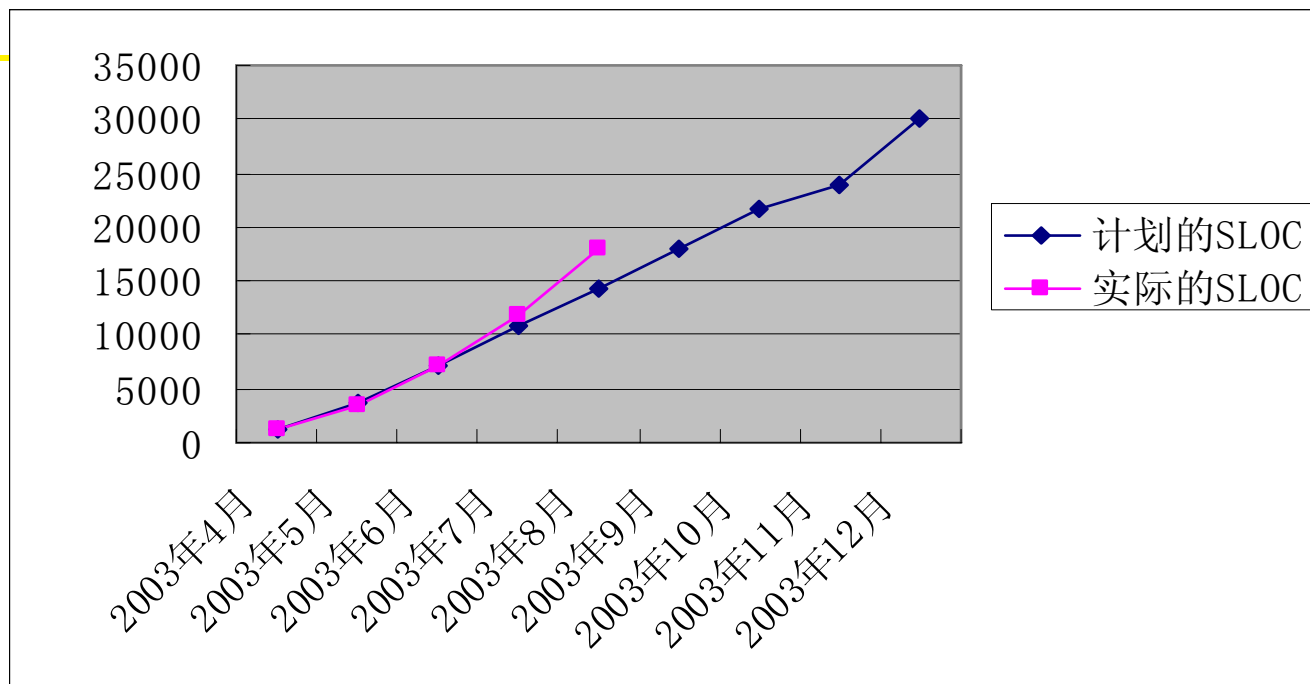


# 度量构造表——模板

信息需要	度量出来干什么用
信息分类	
可测量概念	
指示器	
分析模型	
决策准则	
派生测量	
度量函数	
基本测量	
度量方法	
方法类型	
刻度	
刻度的类型	
度量单位	
相关实体	
属性	

## 实例：物理规模和稳定性

- 在这个实例中，软件经理要评估原始规模估计的准确性。这个信息需要是通过比较计划与实际的代码量来处理的。
- 在下图的指示器中，绘制了计划和实际产生的代码行的基本测量。另外，计算了软件规模增长率的派生测量。该指示器似乎表明项目生产率比进度提前，但经过进一步调查后，表明一个组件的实际代码数要高于计划的代码数，因为遗漏的需求直到初始组件测试时才标出来。资源分配、进度、预算以及测试进度和计划都受到了这个无法预计的规模增长的影响。



	1-Apr	1-May	1-Jun	1-Jul	1-Aug	1-Sep	1-Oct	1-Nov	1-Dec
计划的SLOC	1200	3800	7200	10800	14400	18000	21600	24000	30000
实际的SLOC	1190	3500	7250	11900	18000				
比率	0.99	0.92	1.01	1.10	1.25				

信息需要	评价一个软件组件的规模来评估原始的预算估计
信息分类	产品规模和稳定性
可测量概念	物理规模和稳定性
指示器	软件规模增长趋势图和记录表
分析模型	软件规模增长率的增加表明达到成本和进度预算的风险的增加
决策准则	如果规模增长率超过1.2，则应该进行调查
派生测量	软件规模增长率
度量函数	将迄今为止完成的实际代码行除以计划的代码行
基本测量	<ol style="list-style-type: none"> <li>1、计划的软件源代码行数</li> <li>2、实际的软件源代码行数</li> </ol>
度量方法	<ol style="list-style-type: none"> <li>1、计算到当前时期为止计划完成的累计代码行数</li> <li>2、计算在目前已批准的软件源代码库中的源代码行数</li> </ol>
方法类型	<ol style="list-style-type: none"> <li>1、客观的</li> <li>2、客观的</li> </ol>



刻度	1、从0到无穷大的整数 2、从0到无穷大的整数
刻度的类型	1、比率 2、比率
度量单位	1、源代码行 2、源代码行
相关实体	1、软件开发计划或进度 2、基线化的软件源代码库
属性	1、在每个时期计划完成的代码行数 2、软件源代码行数

## 2、信息分类

大部分项目信息需求可以分成通用的组，称作“**信息分类**”：

- 进度和进展
- 资源和费用
- 产品规模和稳定性
- 产品质量
- 过程性能
- 技术有效性
- 客户满意度

## 进度和进展

- 这个信息分类针对的是项目里程碑的实现及工作单元的完成问题。
- 进度延迟的项目要想按时交付，通常只能削减功能或是牺牲产品质量。

# 进度与进展

可测量概念	要解决的问题	预期的测量
里程碑完成	项目符合预定的里程碑吗？	里程碑日期
关键路径性能	关键任务或交付日期延迟了吗？	缓冲时间
工作单元进展	特定的活动和产品进展如何？	已跟踪的需求、已测试的需求、已打开的问题报告、已关闭的问题报告、已完成的评审、已打开的变更请求、已解决的变更请求、已设计的单元、已编码的单元、已集成的单元、已尝试的测试用例、已通过的测试用例、已打开的行动项、已完成的行动项
增量式能力	要交付的能力像在增量式构造和发布中预定的那样吗？	已集成的构件、已集成的功能

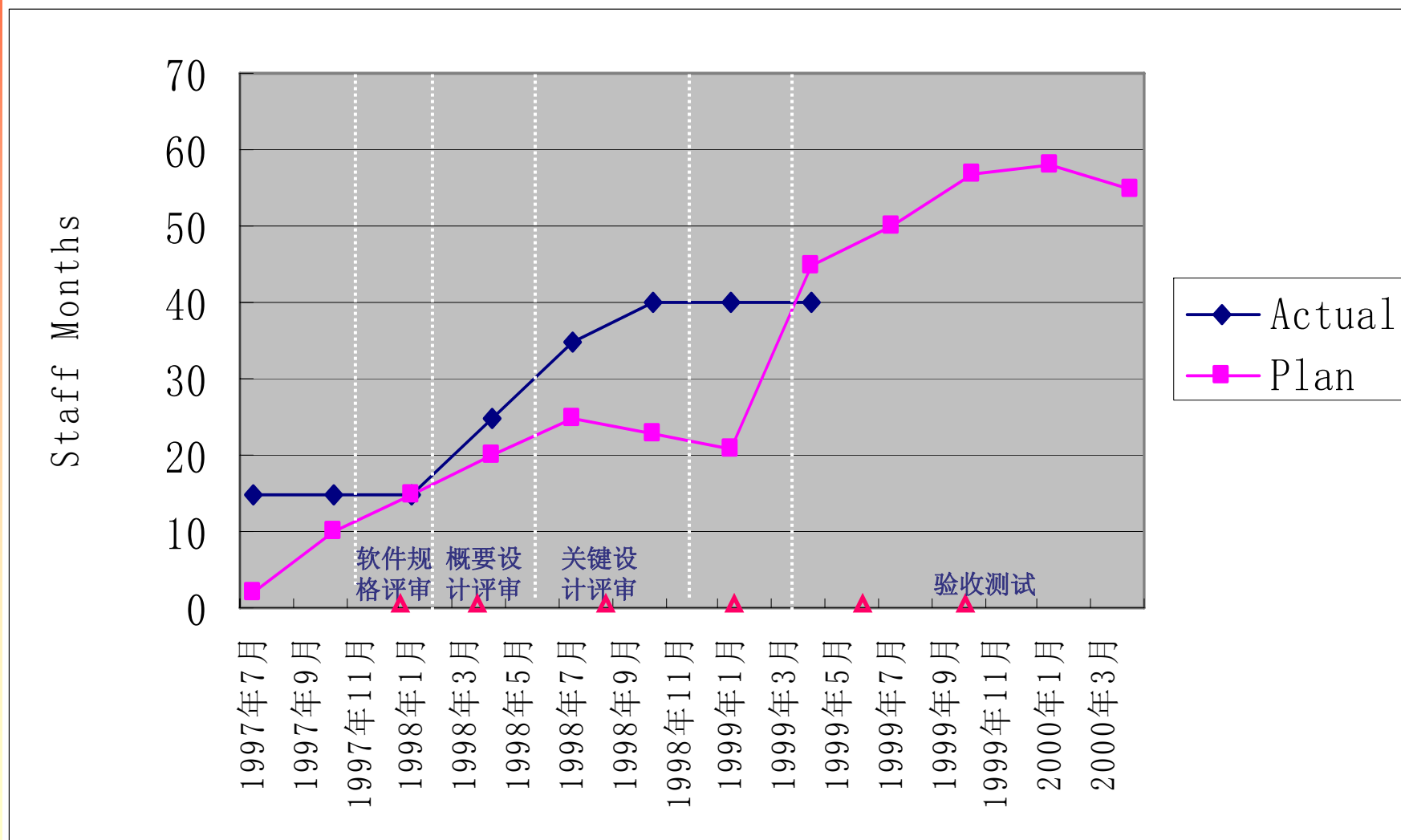
## 资源和费用

- 这个信息分类涉及在要执行的工作与项目分配的人力资源之间做出平衡。
- 超出预算工作量的项目通常只能通过削减软件功能或牺牲产品质量来恢复。

# 资源和费用

可测量概念	要解决的问题	预期的测量
人员工作量	<p>所花工作量是按计划的吗？</p> <p>是否有足够的具备所需技能的员工？</p>	<p>员工级别、开发工作量、经验水平、人员调整、BCWS（本月计划工作的累计预算成本）、BCWP（本月计划工作的累计实际成本）和ACWP（本月已完成工作的累计预算成本）</p>
财务性能	项目是否满足预算和进度目标？	预算、费用
环境和支持资源	需要的设施、设备和材料是否可获得？	需要的数量、可用的数量、可用的时间、已用的时间

# 人员配制情况－实例



## 产品规模和稳定性

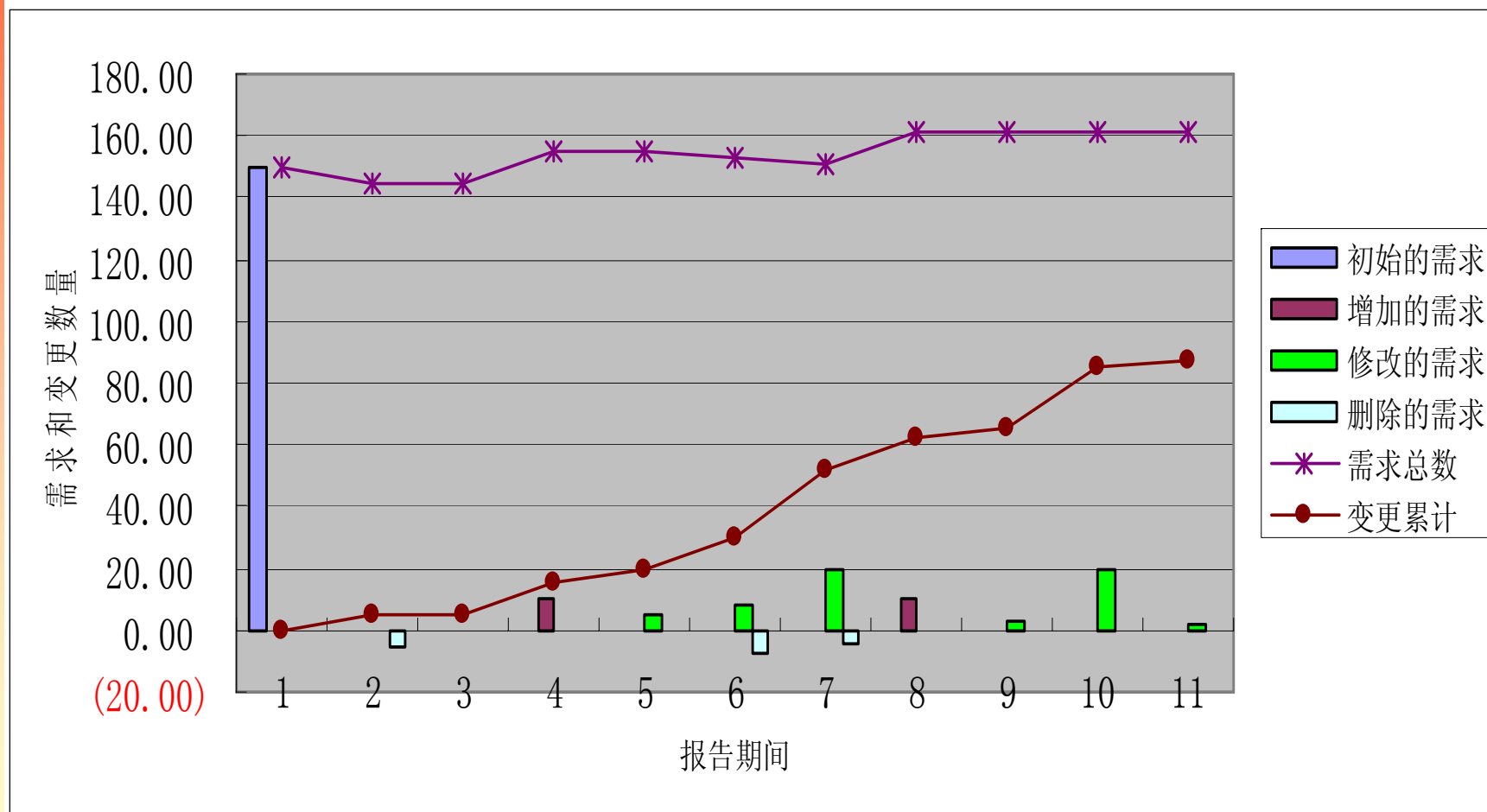
- 这个信息分类针对功能的稳定性及要求的软件能力。它还与提供所需能力的要交付的软件容量有关。
- 稳定性包括功能范围或数量的变更。软件规模的增长通常要求增加所用的资源或是延长项目的进度计划。



# 产品规模和稳定性

可测量概念	要解决的问题	预期的测量
物理规模和稳定性	产品的规模、内容、物理特性或接口变更有多少？	数据库大小、组件数、接口数、代码行数
功能规模和稳定性	需求和相关的功能变更有多少？	需求数、功能变更数、功能点数

# 需求变更的度量



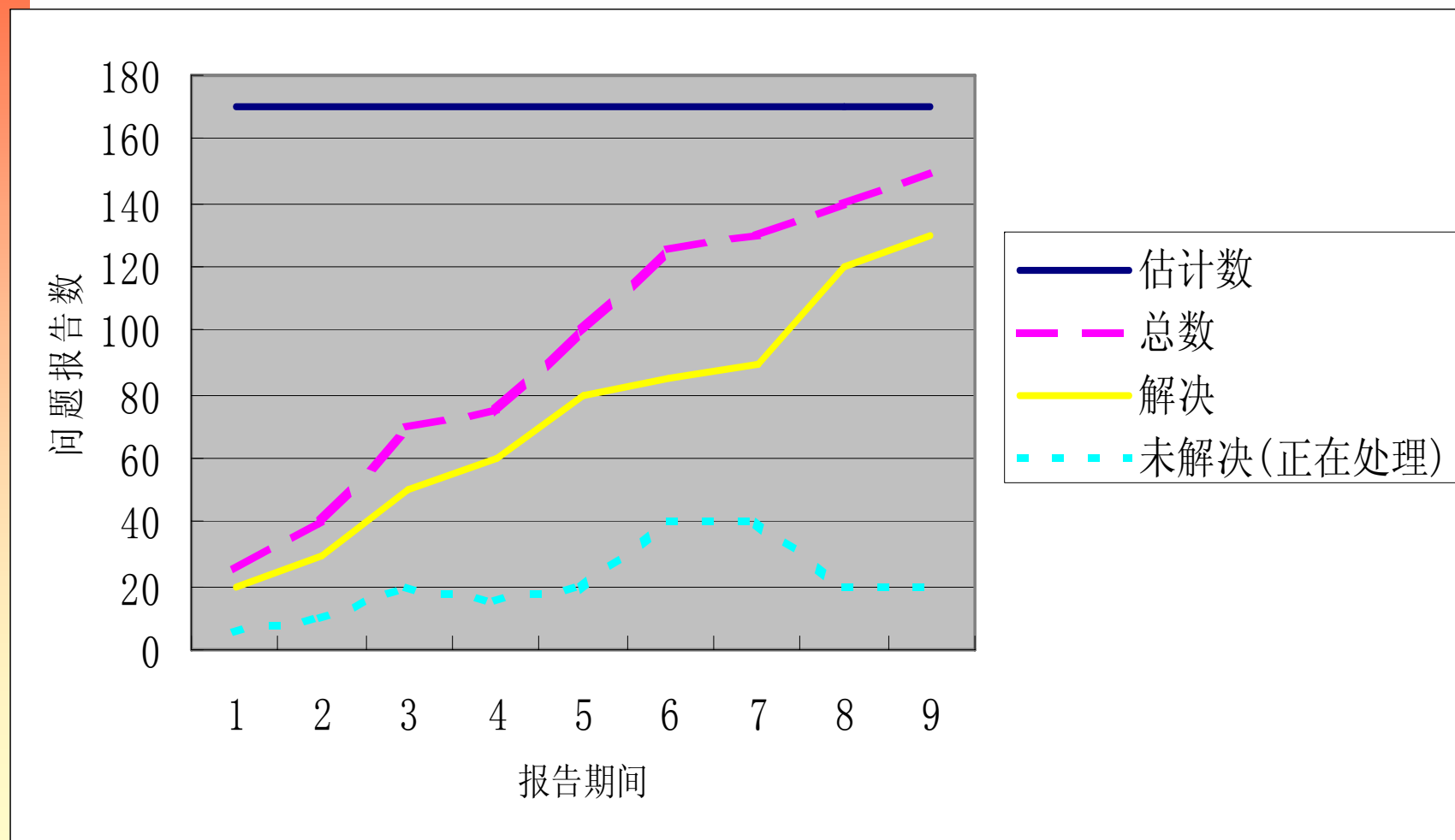
## 产品质量

- 这个信息分类针对已交付的软件产品支持用户的需要而不出错的能力。
  -
- 如果交付了低质量的产品，那么使产品正常运转的责任通常落在了负责维护的组织身上。

# 产品质量

可测量概念	要解决的问题	预期的测量
功能正确性	产品质量是否达到了交付给用户的水平？已标识的问题是否得到了解决吗？	缺陷、缺陷的持续时间、技术性能水平
可维护性	系统要求多少维护？维护的难度如何？	恢复的时间
效率	目标系统能有效地使用系统资源吗？	利用率、吞吐量、响应时间
可移植性	功能在另一平台上重新驻宿达到了什么程度？	标准的符合性
可用性	用户接口是否足够且便于操作？操作员的错误是在可接受的范围内吗？	操作员错误
可靠性	给用户的服务常常被中断吗？故障率是在可接受的范围内吗？	平均故障间隔时间（MTBF）

# 问题报告状态



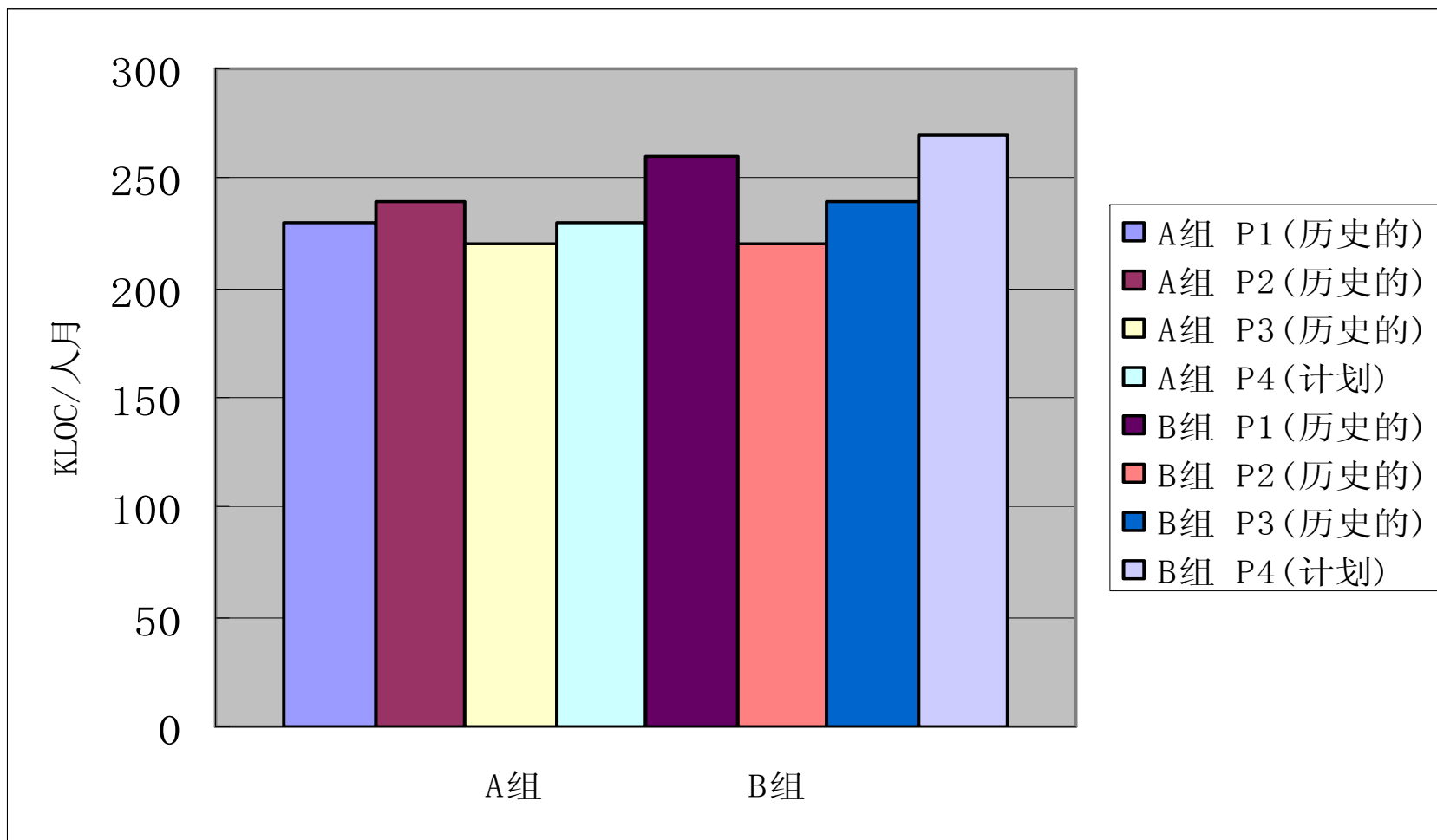
## 过程性能

- 这个信息分类涉及与项目需要相关的供应商的能力。
- 缺乏软件开发过程或低生产率的供应商可能难以满足积极进取的项目进度和费用目标。

# 过程性能

可测量概念	要解决的问题	预期的测量
过程符合性	项目实现已定义的过程的一致性如何？	基准成熟度评定、过程 审计发现
过程效率	过程效率是否达到了满足当前 委托和计划的目标？	生产率、循环时间
过程有效性	因返工需要花多少额外的工作 量？	已包括的缺陷、漏掉的 缺陷（Defects Escaping）、返工工作 量、返工组件

# 开发性能





## 技术有效性

---

- 这个信息分类针对已提议的技术方法的可行性。它针对工程方法，如软件重用、商业软件构件的使用、对高级软件开发过程的依赖以及通用软件构架的实现。
- 如果已提出的技术方法中的关键元素无法实现，那么可能导致费用的激增和进度的拖延。

# 技术有效性

可测量概念	要解决的问题	预期的测量
技术适合性	技术满足所有的已分配的需求吗或需要额外的技术吗？	需求覆盖
技术易变性	新的技术是否因太多的变更而造成风险？	基线变更

## 客户满意度

- 这个信息分类针对项目交付的产品与服务满足客户期望的程度。
- 满意程度的指示器可能来自客户的反馈以及所要求的客户支持的级别。
  -

# 客户满意度

可测量概念	要解决的问题	预期的测量
客户反馈	我们的客户多大程度上理解项目的性能？项目满足用户的期望吗？	满意度评定、奖励费用
客户支持	客户支持请求多快能得到处理？	支持的请求数、支持时间

### 3、信息需要

- **信息需要**：为了做出正确的决策，度量用户（如经理或项目组成员）需要知道什么。
- **信息分类**：信息需要的逻辑分组，为信息模型提供了结构。
- **可测量概念**：通过定义实体及其可测量的属性来满足信息需要的概念。

## 4、属性

- **相关实体**：要被测量的对象。实体包括一个项目的过程或产品元素，如项目任务、计划/估计、资源和可交付品。
- **属性**：属性是软件实体的特性或特征。一个实体可能有许多不同的属性，其中只有一部分是适合测量的。可测量的属性对于人或自动化手段都必须是可识别的，无论在定量还是定性上。

## 5、基本测量

- 一个基本测量是由一个指定的度量方法（例如，计划的代码行数，到目前为止的累计成本）定义的一个单个属性的测量。执行度量方法产生测量的一个值。当采集数据时，值被赋给基本测量。基本测量在功能上是独立于其它所有测量的，它获取单个属性的信息。
- 每个基本测量定义了如下特征：
  - 度量方法
  - 方法类型
  - 刻度
  - 刻度类型
  - 度量单位

## 6、度量方法

- 度量方法是操作的逻辑顺序，用于以一个指定的刻度来量化一个属性。操作可能包括诸如计算发生的次数或者观察时间的流逝这样的活动。
- 相同的度量方法可以应用于多个属性，但将不同的属性和方法组合在一起将产生一个不同的基本测量。
- 度量方法可能通过两种方式实现——**自动或手动**。度量规程描述了在给定的组织级环境内度量方法的特定实现。规程可能记录在度量计划中，也可能单独记录。



## 7、方法类型

- 度量方法的类型依赖于用来量化特定属性的操作本身。有两种类型的方法：
  - **主观方法**：量化包括了人的判断或估计。例如，依赖专家将功能的复杂性划分成高、中或低三等就是一种主观的度量方法。
  - **客观方法**：量化基于数字规则，如计数。这些规则可能由人工或者自动的方式来实现。例如，Ada代码的行数可以通过计算分号来量化。
- 客观的测量一般比主观方法更为准确和可重复。可能的话，客观的度量方法更为合适。

## 8、刻度

- 刻度是连续的或离散的值有序集，或者是一个属性被映射到的分类集。
- 刻度定义了执行度量方法产生的可能值的范围。度量方法将要测量的属性的大小映射为刻度上的一个值。一个度量的单位通常是与一个刻度相关的。

## 9、刻度类型

- 刻度的类型依赖于刻度上值之间关系的类型：
  - **比率**：数字数据，相同的间距有相同的属性量，数值0不对应任何属性。代码行的计算会产生从0到无穷大的取值范围。
  - **区间**：数字数据，相同的间距有相同的属性量，不使用0。
  - **顺序**：离散的等级排列（ranking）。例如，一组软件单元可能会根据预期的实现难度排序：最困难（most difficult）和很困难（second most difficult）等等。
  - **标称**：分类的数据。例如，一组问题报告可能根据问题的来源分类：需求和设计等等。
- 度量方法的类型通常影响可靠地应用于给定属性的刻度的类型。例如，度量的客观方法通常只支持顺序或标称刻度。

# 标称刻度 (nominal scale)

- 最简单的活动或最低层的测量就是分类。在分类中，我们把根据某个属性把元素划分为不同的类别。例如，对于宗教，可以将研究的科目分为天主教、新教、犹太教、佛教；又如，可以把产品开发过程分为瀑布开发过程、螺旋开发过程、迭代开发过程、面向对象开发过程等
- 在标称刻度中，对类别的关键需求是联合穷举和相互排斥。相互排斥指一个主题能够且只能被归到一个类别中；联合穷举指所有类别应该覆盖了所有可能存在的属性类别。如果一个属性有我们感兴趣的类别之外的类别，那么需要使用“其他”类别，以使类别达到联合穷举。
- 标称刻度中，类别名称和排列顺序没有对类别之间的关系作出假设。例如，瀑布开发过程列在螺旋开发过程之前，并没有说它们中的一个过程比另一个过程“要好一些”或“伟大一些”

## 顺序刻度 (ordinal scale)

- 顺序刻度指将主题按顺序进行比较的测量操作。例如，可以按社会经济地位对家庭进行分类，如分为“上层阶级”、“中层阶级”和“下层阶级”。又如，SEI成熟度等级分为“初始级”、“可重复级”、“已定义级”、“已管理级”和“优化级”
- 在测量层次上，顺序刻度比标称刻度要高一些。通过顺序刻度，不仅可以对主题分类，而且可以将类别排序。顺序刻度是不对称的，若 $A > B$ 为真，则 $B > A$ 为假。顺序刻度有传递性，即 $A > B$ ， $B > C$ ，则 $A > C$
- 顺序刻度不能体现元素之间的差别大小。例如，客户满意度调查中，通常采用5分制，即1=完全不满意，2=部分不满意，3=一般，4=满意，5=完全满意。我们仅仅知道， $5 > 4$ 、 $4 > 3$ 或 $5 > 2$ ，但不能说5分比4分好多少，也不能说5分与4分间的差别与3分和2分间的差别是一样的。因为，要使客户的满意度从满意（4分）增长到完全满意（5分），同从部分不满意（2分）增长到一般（3分），需要非常不同的改进活动和种类
- 将顺序刻度转为数学运算时，不能进行加减乘除这样的运算

## 区间刻度 (Interval scale)

- 区间刻度可以表示测量点之间的精确差值，可以对区间刻度进行加减数学运算。例如，若软件产品A的缺陷率是每千行代码5个缺陷，而产品B是每千行代码3.5个缺陷，则可以说产品A的缺陷级比产品B每千行代码高出1.5个缺陷。
- 测量区间刻度，需要建立一个严格定义的测量单位。给定了测量单位，就可以说两个值之间相差15个单位，或者说两个差值是相等的。例如，若产品C的缺陷率是每千行代码2个缺陷，则可以说产品A和B的缺陷率的差值与产品B和C的缺陷率差值是相同的。

# 比率刻度

- 如果在区间刻度中，定义绝对的或非随意的零点值，则区间刻度就成了比率刻度。
- 比率刻度是最高级的测量，可以进行所有的数学运算，包括乘除运算。例如，可以说产品A的缺陷率是产品C的缺陷率的2.5倍，因为当缺陷率为0时，意味着产品中不存在任何缺陷。
- 注意：零点不能随意设定。例如，传统的温度测量（华氏温度和摄氏温度）。我们说夏天平均温度80F，冬天平均温度16F，差值是64F，但不能说80F比16F热5倍。华氏温度和摄氏温度都是区间刻度，不是比率刻度。
- 除了少数特例外，几乎所有的区间刻度都是比率刻度。即建立单位的大小之后，就可以构想出零单位来。
- 注意：每个高级刻度具备低级刻度的所有属性。测量级别越高，对数据的分析就越有力。

## 10、度量单位

- 用于导出基本测量值的标准化的定量的数量，如一个小时或一行代码。
- 每个度量单位是一个特殊的量，同类的其它量通过它来进行比较，以表示它们相对于这个量的大小。只有通过同一个度量单位来表示的量才能直接比较。
- 度量单位通常不是为由主观方法所确定的测量而定义的，或者说不是为实体化的序号或标称刻度而定义的。



## 11、派生测量

- **派生测量**：被定义成两个或多个基本和/或派生测量的函数。派生测量获取多个属性的信息。
- 例如，生产率就是派生测量的一个示例，生产率通过代码行的基本测量除以工作小时的基本测量来导出。
- 基本测量的简单转化（例如，基本测量取平方根）不会增加信息，因而不会产生派生测量。
- 数据的规范化通常包括将基本测量转化为派生测量，以用于比较不同的实体。

## 12、度量函数

- **度量函数**：用来合并两个或多个基本和/或派生测量的算法或计算。派生测量的刻度和单位取决于它所组合的基本测量的刻度和单位，同时还取决于它们是如何由函数组合的。
- 例如，除法（Division）是用来产生生产率这个派生测量的函数，生产的产品总量除以总的工作量就可算出生产率。

## 13、指示器

- 指示器：一个或多个测量（基本或派生的）的显示，是使用预定义算法或模型组合基本测量和/或派生测量而产生的。
- 指示器支持用户为分析和制定决策而导出信息。
- 典型的指示器值是一个数字或一系列数字。指示器常用图形或图表来表示。
- 好的指示器可以：
  - 支持指定的信息需要的分析
  - 支持需要的分析类型（估计、可行性分析或性能分析）
  - 提供适当的细节等级
  - 提示一个可行的管理行动
  - 为做出决策和采取行动提供及时的信息
- 指示器通常用来将实际值与它们的期望值做比较。期望值可能基于历史平均数、计划数字、指定的限定值或阈值范围。模型和决策准则帮助决定实际数据与期望数据之间的差距是否过大。

## 14、分析模型

- 分析模型是一个具有相关决策准则的包括两个或多个基本测量和/或派生测量的算法或计算。
- 分析模型基于对组件测量及其随时间变化的行为之间的期望关系的理解和假设。
- 分析模型产生于已定义的信息需要相关的估计或评价。度量的刻度和方法会影响用来产生指示器的分析技术或模型的选择。

## 15、决策准则

- 决策准则用来决定实际数据和期望数据间的差距是否过大。它是数字阈值、目标和界限，用于确定行动或进一步调查的需要，或者在一个给定结果中描述置信等级（**level of confidence**）。决策准则帮助解释度量结果。
- 决策准则可能基于对期望行为的概念上的理解或从数据算出。决策准则可能源自历史数据、计划和直观判断或者作为统计控制界限（**control limits**）或统计置信界限（**confidence limits**）被计算。

## 四. 建立度量库

# 1、度量库数据

- 度量库的数据包括：
  - 项目特征信息
  - 工作产品规模的估计值
  - 工作量和成本的估计值
  - 规模、工作量和成本的实际值及偏差（要有趋势图）
  - 需求和配置项变更统计
  - 项目进度：预计的开始和结束日期，实际的开始和结束日期
  - 里程碑完成情况
  - 质量度量（如缺陷率、发现的缺陷数、缺陷的严重程度等）

## 度量库数据（续）

- 同行评审覆盖度
- 测试覆盖度
- 资源模型
  - 工作量的各阶段的工作量比例
  - 各种类型的任务的工作量比例
  - 各种类型的任务在各阶段的投入比例
- 生产率
- 等等
- 度量数据的采集、存储、更新和检索等的规程



## 2、项目特征信息

- 项目名称、项目经理及小组长的名字
- 项目的目标
- 项目选用的生命周期
- 项目的应用领域、硬件平台、所用的开发语言、所用的DBMS等
- 项目持续时间
- 项目的规模、工作量
- 项目的团队规模
- 人员经验水平
- 项目风险信息
- 是成功的项目，还是不成功的项目

### 3、资源模型 – 1

- 按项目(规模和类型)、阶段(生命周期)和活动类型等三个属性分类记录工作量
- 按分析、设计、实现、测试、项目管理、PPQA(产品和过程质量保证)、SCM(软件配置管理)、评审等活动类型统计工作量
- 各阶段的工作量比例
- 各种类型的任务的工作量比例
- 各种类型的任务在各阶段的投入比例
- 把组织内所有项目的资源模型图折算为各个等价的资源模型图,然后相加取平均值,就得到该组织的资源模型图
- 随着项目的进展,不断积累数据;然后根据积累的数据对该组织的资源模型图进行不断更新(一般一年一次)

工作日志											
姓名											
本周起始日期											
				工作时间（小时）							
项目名	活动	活动类型	阶段	周一	周二	周三	周四	周五	周六	周日	合计
正常工作时间											
PRO32	CDMIP1-Coding			8	8	8					24
PRO32	CDMIP1-Review						4				4
PRO32	CDMIP1-更改							4	8		12
	正常工作时间小计			8	8	8	4	4	8	0	40
加班时间											
PRO32	CDMIP1-更改						2	1	2		5
	加班小计			0	0	0	2	1	2	0	5
	合计			8	8	8	6	5	10	0	45

## 资源模型 – 2

- 坚持日填、周报、月分析的制度
- 日填数据的精确度为半小时
- 开发自动化的数据统计工具
- 按项目类型和项目规模建立组织级的资源模型
- 每年更新一次组织级的资源模型

# 资源模型 – 3

- 注意事项

- 数据真实
- 填写及时（日填与周填）
- 模型科学（度量元正确）
- 表格规范（以便同类相加）
- 同类可比（指数据的值）
- 位数适当（指数据的准确度）
- 注意汇总（记录、统计、汇总和分析）
- 定期分析（日填、周报、月分析）

谢谢!