Measures

实用软件度量

麦哲思科技（北京）有限公司

课程的目的

Measures

- 本课程旨在帮你解决如下的问题：
  - 为什么需要度量？
  - 应该度量什么？
  - 如何准确定义度量数据？
  - 如何采集数据？
  - 如何验证数据？
  - 如何分析度量数据？
  - 如何将度量数据应用到实际项目中？

主　题

Measures

- 概述
- 需要采集什么数据？
- 如何展示数据?
  - 五种基本图形
  - 如何设计指示器
- 如何采集与校验数据?
- 如何定义度量体系?
- 如何分析数据?
  - 基本数据分析技术
  - 如何编写分析报告
  - 过程稳定性分析技术
- 度量应用实例
  - 如何分析与使用工作量数据？
  - 如何跟踪项目进展？
  - 如何分析与使用生产率?
  - 如何使用度量数据管理评审过程？
  - 如何设定量化管理目标？
- 小结

为什么需要度量？

Measures

	问题	感性回答	理性回答
了解	项目进展如何	进展顺利	工期延迟3天 工期延迟1%
评价	代码走查是否有效	可以发现比较多的BUG，但是发现的严重与致命的BUG不多，花费的工作量还是比较大的	代码走查缺陷检出效率是系统测试的4倍，严重与致命的缺陷是系统测试的2倍
预测	为了国庆献礼，10.1之前本项目必须上线，可以吗？	这不可能完成啊！	如果需求砍掉40%，人员每天工作9小时，有80%的把握在10.1之前完成
控制	要减少交付后的缺陷个数，我们应该怎么办？	多加强系统测试吧，系统测试的轮次太少，测试时间太短。	通过静态检查我们发现A和B两人写的代码警告个数占了所有代码的70%以上，因此我们应该优先对这2个人编写的代码执行单元测试
改进	公司今年要提升质量水平，提高效率！	恩，好的！	我们去年的缺陷逃逸率已经低于5%，达到了业内的标杆水平，而工期延迟率平均达到30%，因此我们应该重点提升效率！

如何获得度量的成功

Measures

目标驱动	• 收集有用的度量数据
高层支持	• 将度量的结果真正用于决策
合理考核	• 不要用员工自己提供的度量数据考核自己
最低负荷	• 每天的工作量不要超过10分钟
统一定义	• 明确定义度量元的含义
持之以恒	• 坚持就是胜利

度量是有成本的

Measures

- 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总成本的10-15%
- 国外的一些数据：

组织	数据收集的目的	成本占整个项目 (%)
NASA/SEL	研究	8 – 25
NEC(Japan)	控制质量	10 – 20
STC (UK)	项目管理	3 – 5
NASA/SEL	项目管理	2.5
AT&T	项目管理	1.5
Motorola	项目管理	1

Measures

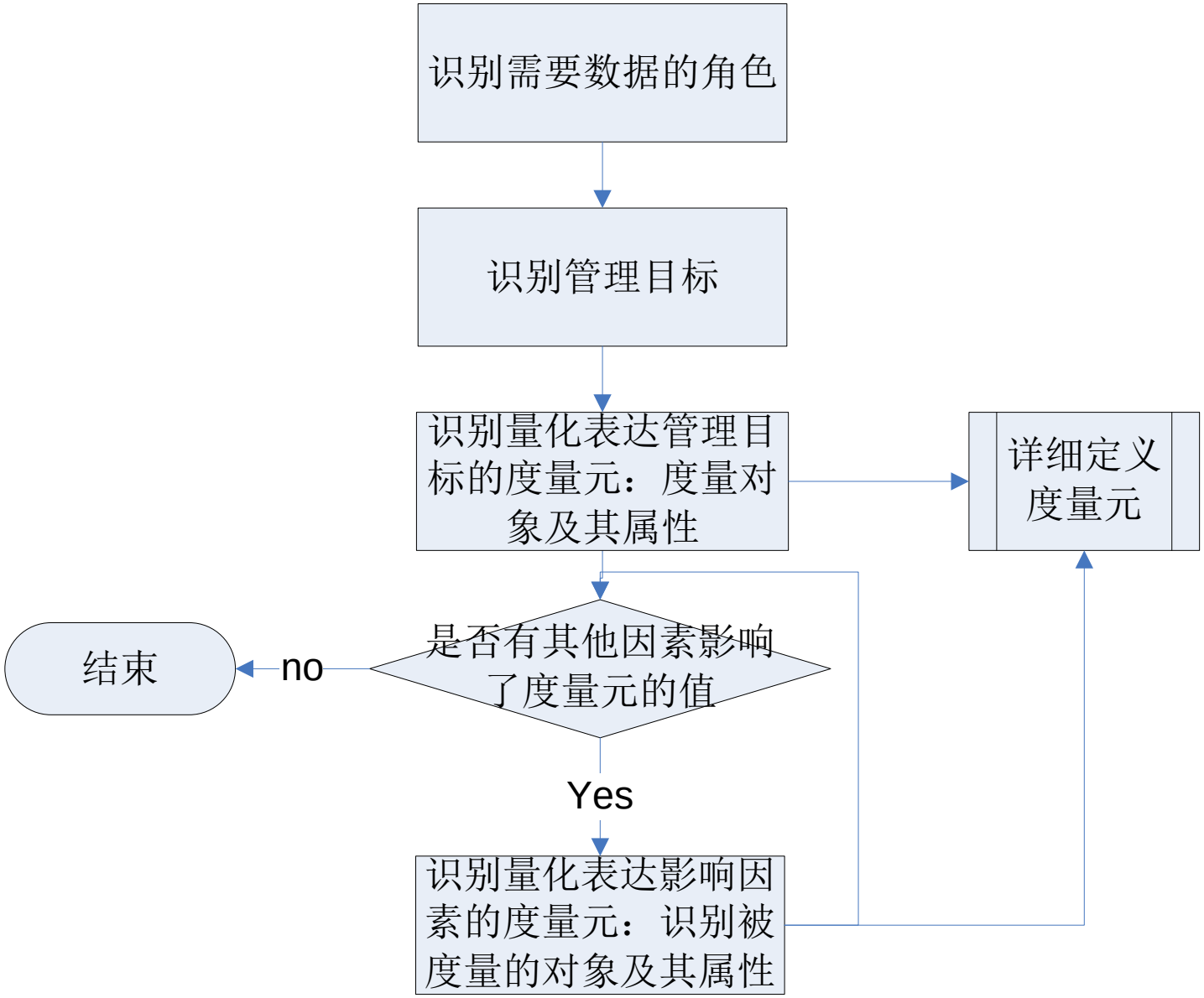
需要采集什么数据？

需要采集什么数据？

Measures

- 目标导向：
  - 谁需要数据？
  - 需要数据做什么？
  - 需要哪些类的数据？
  - 需要哪些数据？

识别度量元的流程


识别需要数据的角色
识别管理目标
识别量化表达管理目标的度量元：度量对象及其属性
详细定义度量元
是否有其他因素影响了度量元的值
no
结束
Yes
识别量化表达影响因素的度量元：识别被度量的对象及其属性

谁需要度量数据？

Measures

- 客户
- 高层经理
- 项目经理
- 测试人员
- QA人员
- EPG成员
- 开发人员
- CM人员

需要度量数据解决什么问题？

Measures

- 要了解什么？
  - 获得对过程、产品、资源等的理解；是后续目的的基础
- 要预测什么？
  - 通过建立预测模型，进行估算和计划
- 要评价什么？
  - 产品的质量、过程改进的效果等
- 要控制什么？
  - 识别偏差，采取纠正行动
- 要改进什么？
  - 根据得到的量化信息，确定潜在的改进机会

度量数据的需求者及其度量目的

Measures

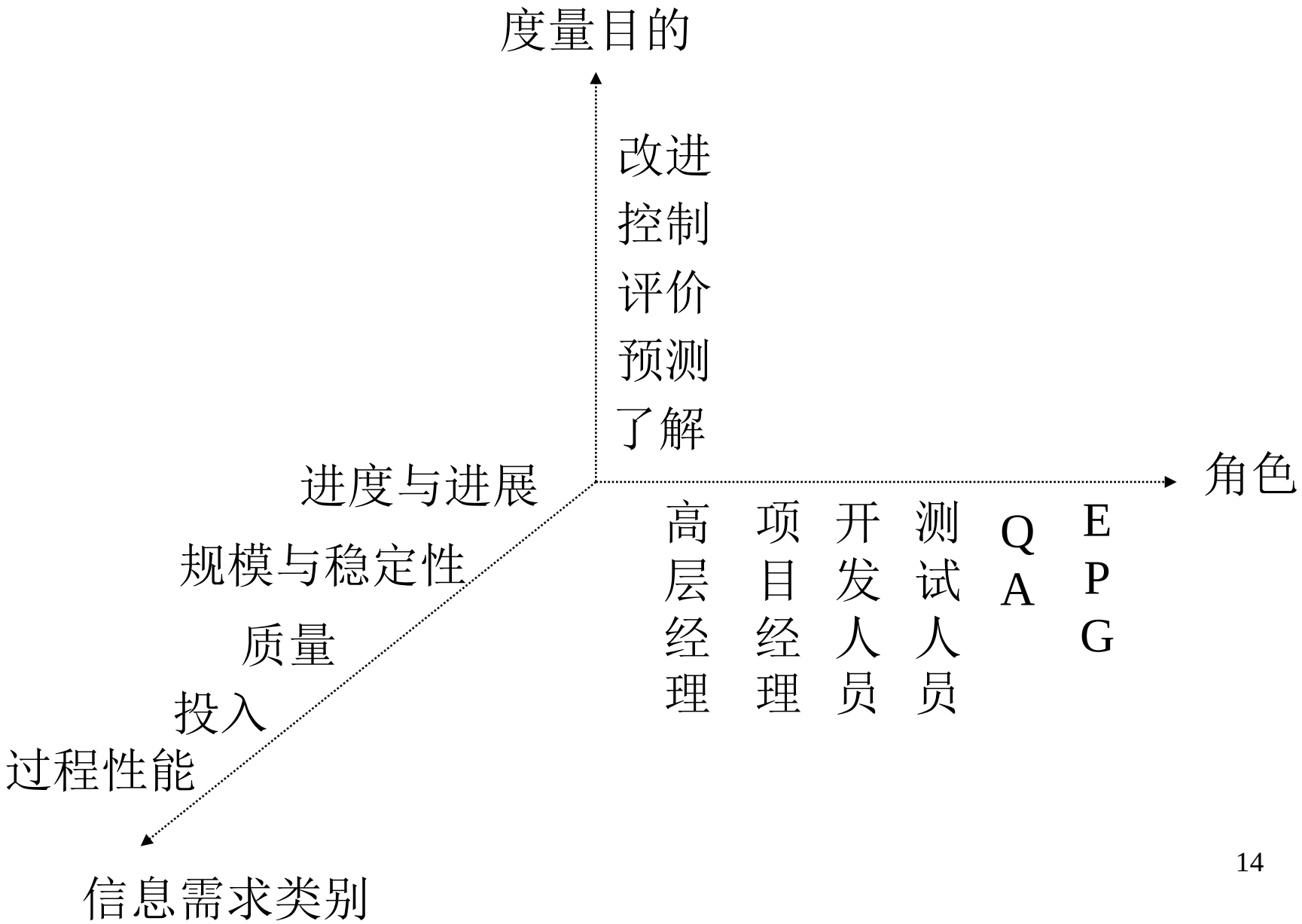
- 度量数据的需求者
  - 客户
  - 高层经理
  - 部门经理
  - 项目经理
  - 开发人员
  - EPG
  - 测试人员
  - 质量保证人员
- 度量的目的
  - 理解
  - 预测
  - 评价
  - 控制
  - 改进

需要哪些数据----五项基本度量

Measures

- 五项基本度量
  - 产品规模和稳定性
  - 进度和进展：
    - 进度：工期的完成情况
    - 进展：任务的完成情况
  - 产品质量
  - 资源和费用
  - 过程性能
    - 过程的效率
    - 过程的质量
- 第六项：
  - 客户满意度

度量需求：角色-度量目的-信息需求类别



案例一：了解顾客服务质量

Measures

- 度量需求者:服务经理
- 管理目标:了解 售后服务质量
- 度量元
  - -如何量化售后服务质量?
    - 未解决的问题总计（TOP）= 新的、发布后未解决的、且在月末还未解决的问题总计
    - 未解决的问题的平均驻留时间(AOP)= 新的、发布后未解决的、且在月末还未解决的问题的总时间/新的、发布后未解决的、且在月末还未解决的问题总计
    - 已解决的问题的平均驻留时间（ACP) = 在月内已解决的发布后问题的问题延续总时间/在月内已解决的发布后问题的问题数

案例二:控制项目进展

Measures

- 度量需求者:项目经理
- 管理目标:控制项目的进度
- 度量元
  - 如何量化项目进度?
    - SPI (进度性能指标) =EV/PV
    - CPI(成本性能指标) = EV/AC
  - 如何控制 ?
    - 当SPI或CPI小于1时，应采取控制措施 ?

案例三：提升项目策划的能力

Measures

- 需要度量数据的角色：EPG
- 解决的问题/度量目的：提升 项目策划的能力
- 度量元：
  - 如何量化项目策划的能力？有几种量化的方法？
    - 项目估算过程：
      - 进度估计偏差率=（实际工期-计划工期）/计划工期
      - 工作量估计偏差率=（实际工作量-计划工作量）/计划工作量
- 影响度量元的因素有哪些？
  - 估计方法
  - 估计人员的水平
  - 客户的水平
  - 项目的规模
- 影响因素的量化：
  - 估计方法:0 宽带DELPHI方法 1 COSMIC FFP方法
  - 估计人员的水平：
    - 技术与管理经验年限
  - 客户的水平:0 不成熟的客户 1 有历史经验的客户 2 经验丰富的客户
  - 项目的规模:功能点

案例四:提升公司的质量水平

Measures

- 度量数据需求者:公司高层经理
- 管理目标:提升公司的质量水平
- 度量元:
  - 如何量化的刻画公司的质量水平
    - 缺陷逃逸率:发布后发现的缺陷个数/(发布前发现的缺陷个数+发布后发现的缺陷个数)
- 影响缺陷逃逸率的因素:
  - 质量的投入
  - 开发人员的技能
- 影响因素的量化
  - 质量的投入
    - 需求评审的投入%=需求评审的工作量/项目总工作量
    - 设计评审的投入%=设计评审的工作量/项目总工作量
    - 代码评审的投入%=代码评审的工作量/项目总工作量
    - 单元测试的投入%=单元测试的工作量/项目总工作量
    - 系统测试的投入%=系统测试的工作量/项目总工作量
  - 开发人员的技能
    - 开发人员的业务经验年限
    - 开发人员的技术经验年限

度量元识别练习

Measures

- 练习:参考度量元识别方法,站在PM的角度,针对至少3个度量需求，识别其需要的度量元。
- 分组:5人一组
- 时间:30分钟
- 输出的格式要求:
  - 度量数据需求者:
  - 管理目标
  - 度量元
  - 影响因素
  - 影响因素的量化

Measures

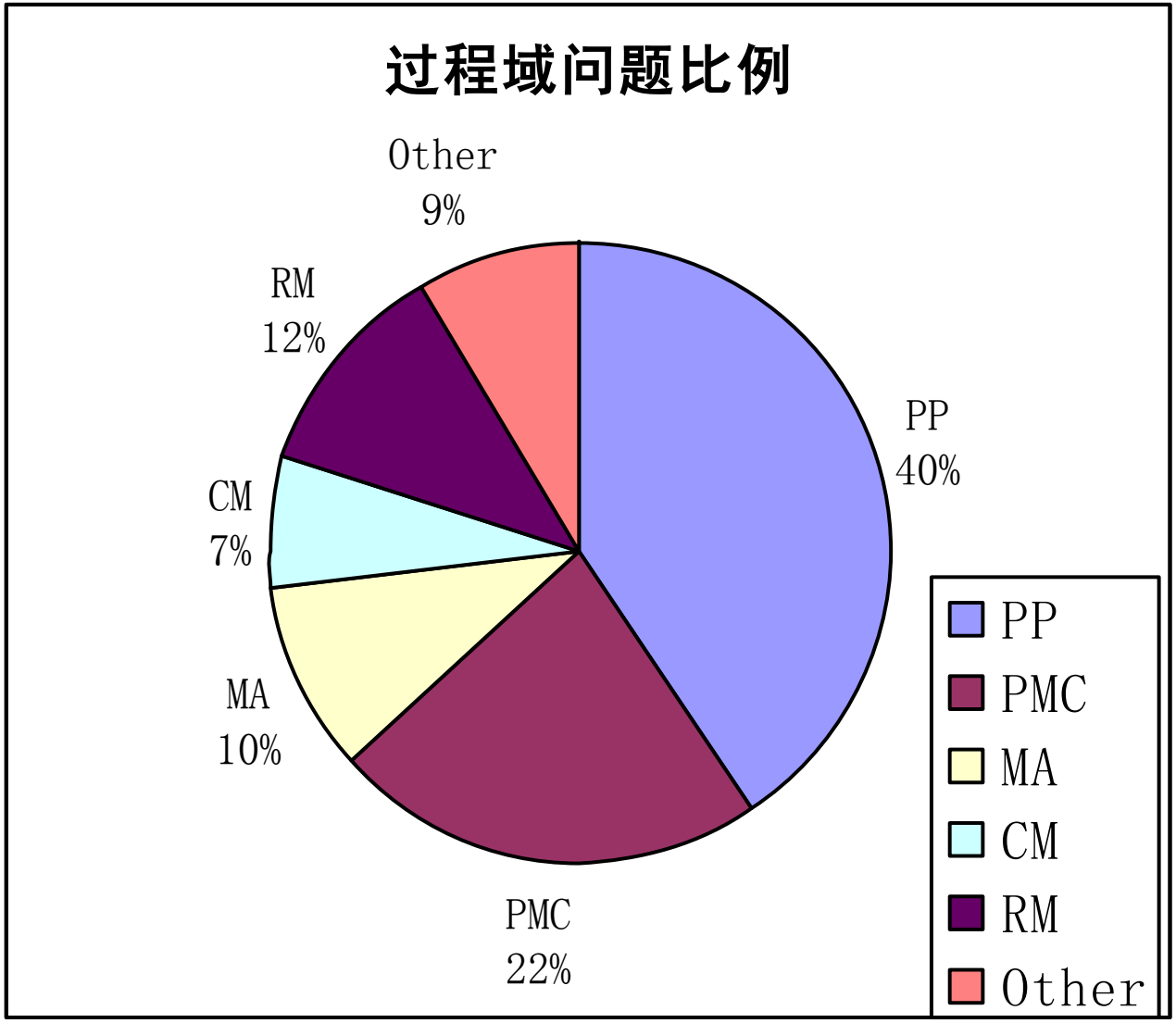
如何展示数据?

5种基本图形

Measures

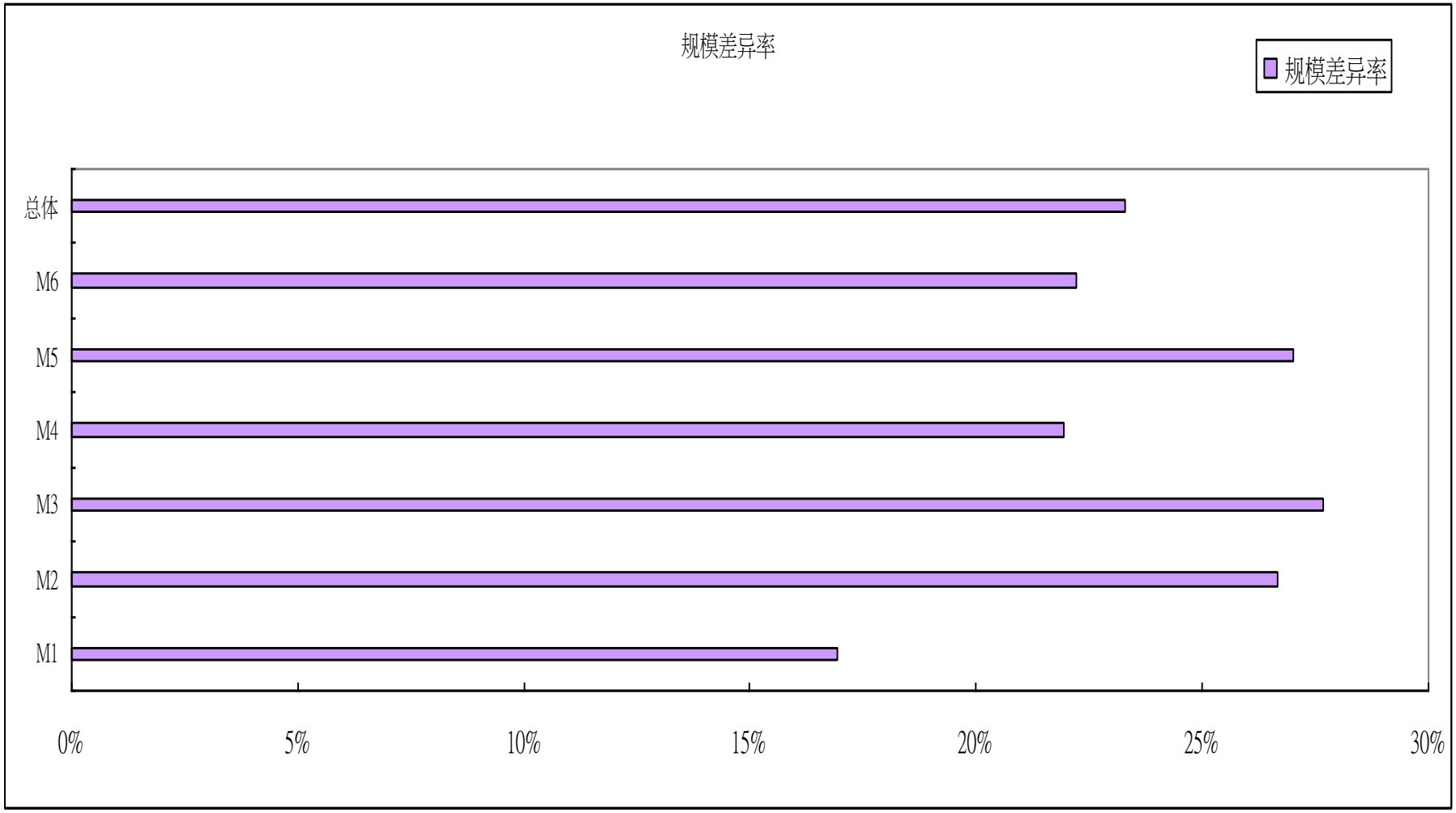
- 饼图
- 条形图
  - 每个类别的名字比较长
- 柱状图
  - 可以有时序对比关系
- 线性图
- 散点图

饼图



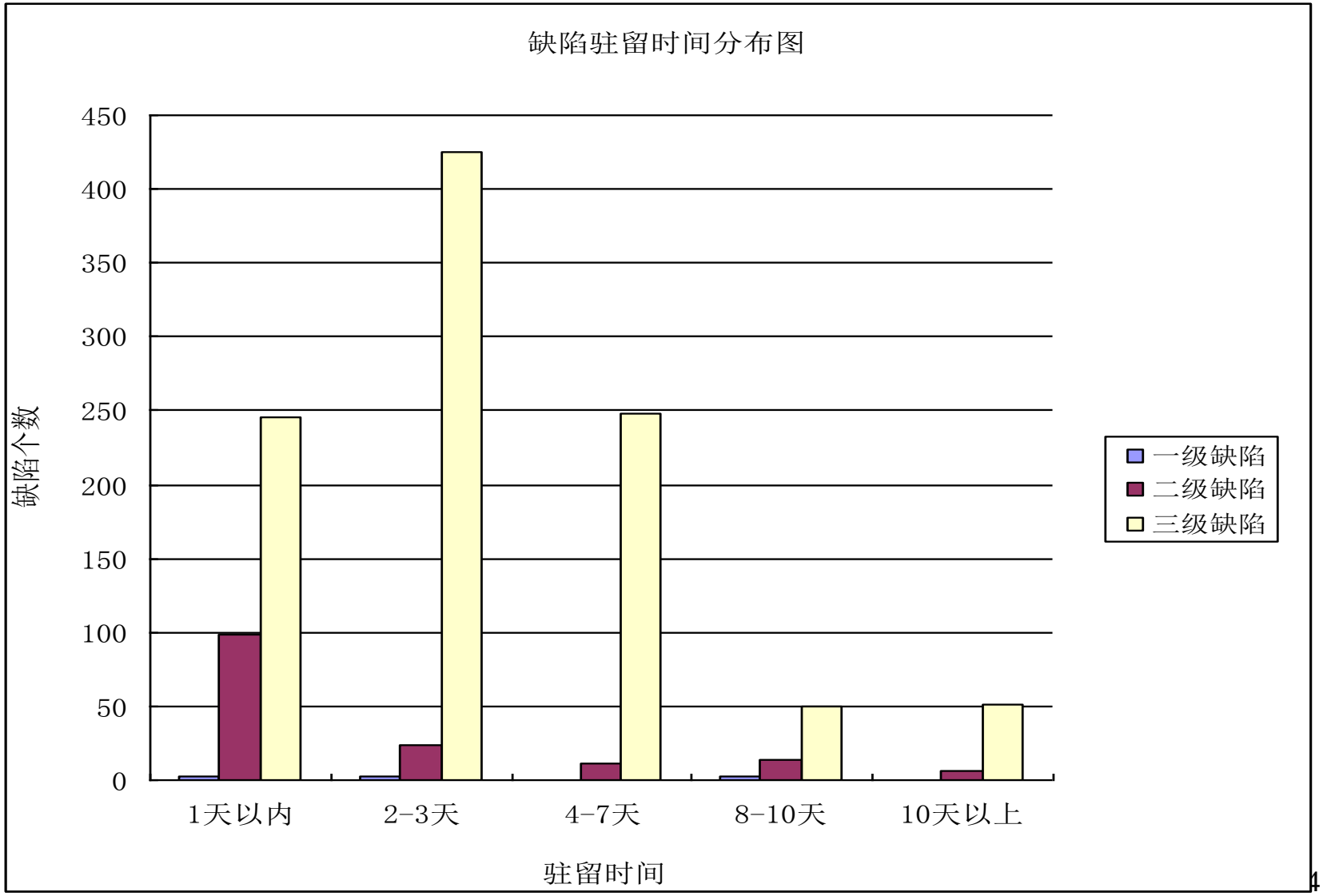
条形图

Measures



直方图

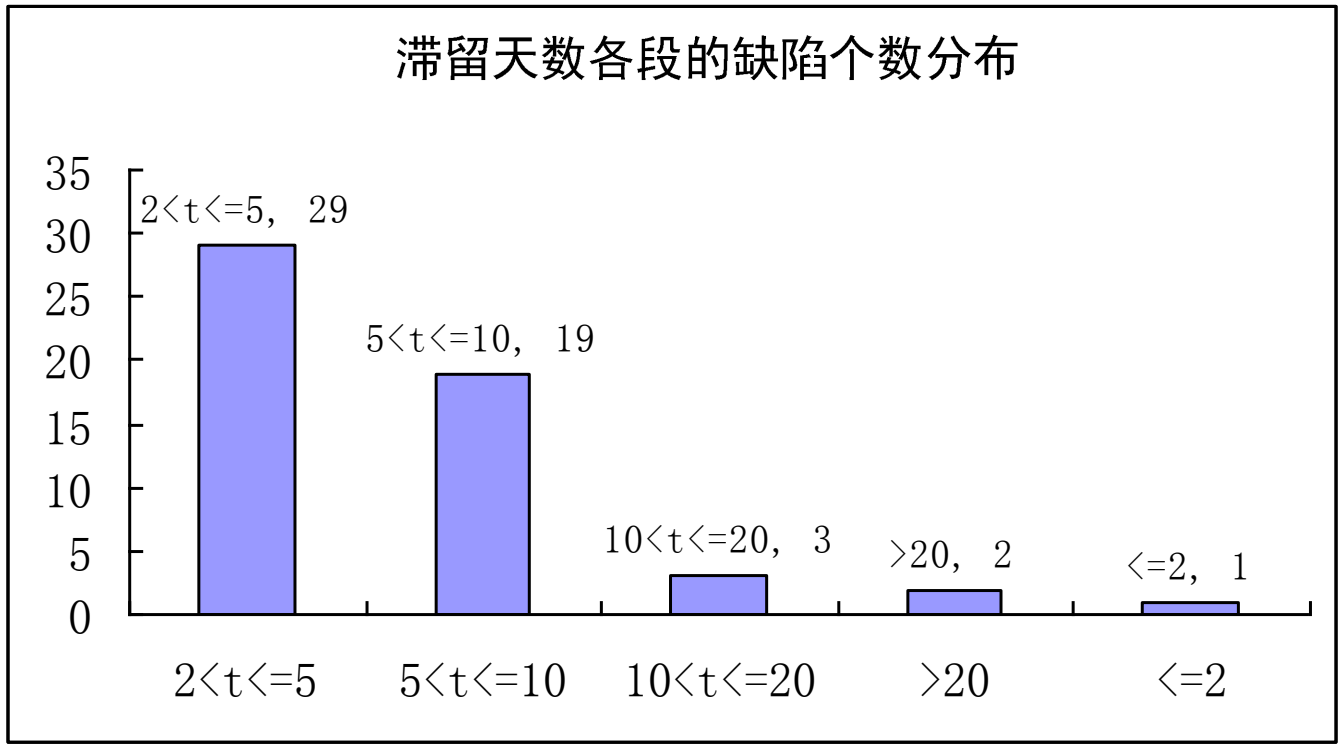
Measures



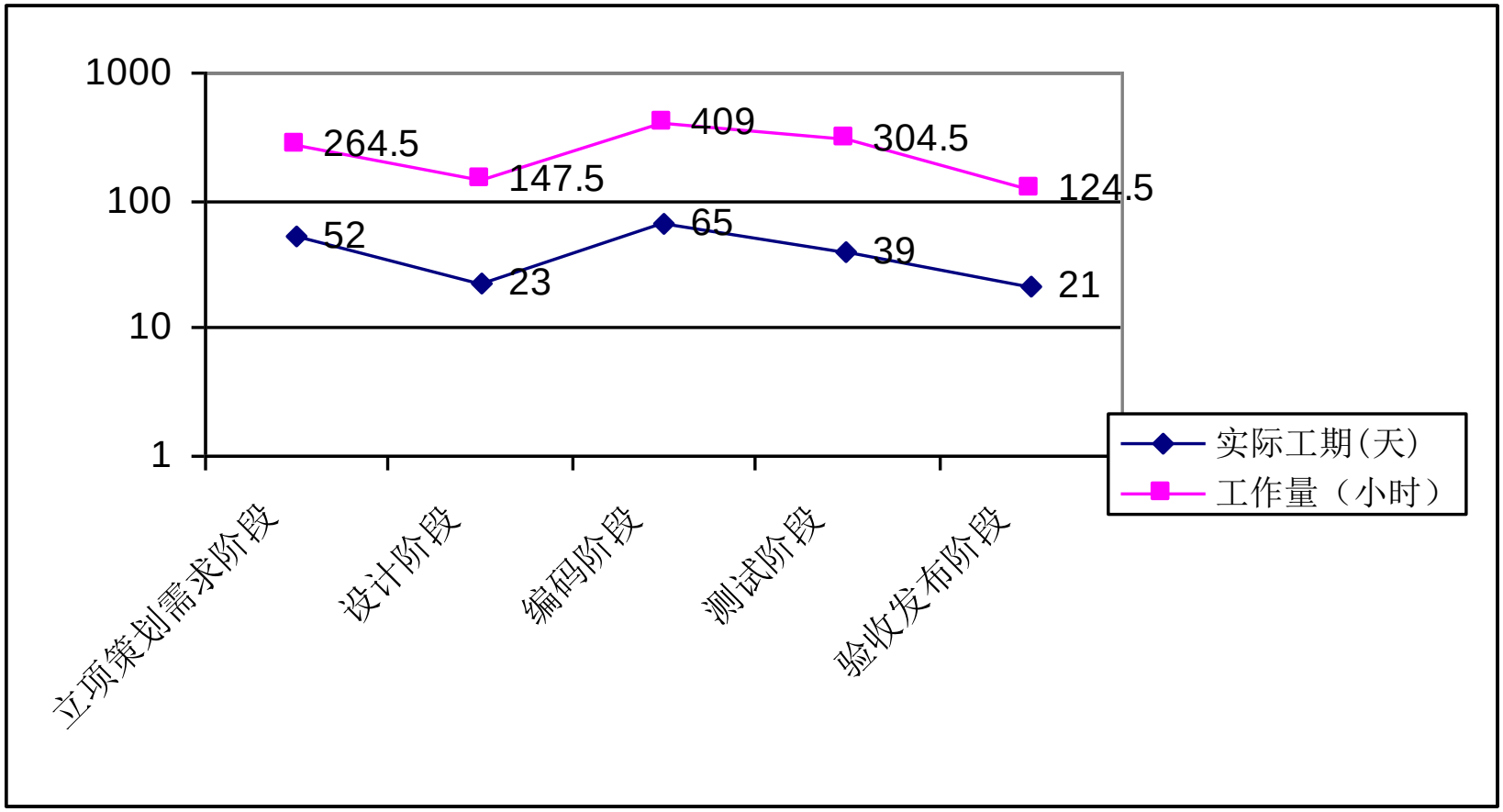
Pareto图

Measures

- Pareto图是直方图或条形图的一种特殊形式。该图根据问题、原因和操作数量、发生的频率或经济影响把它们分为不同的等级，所以有助于集中调查研究对象、找到解决方案。



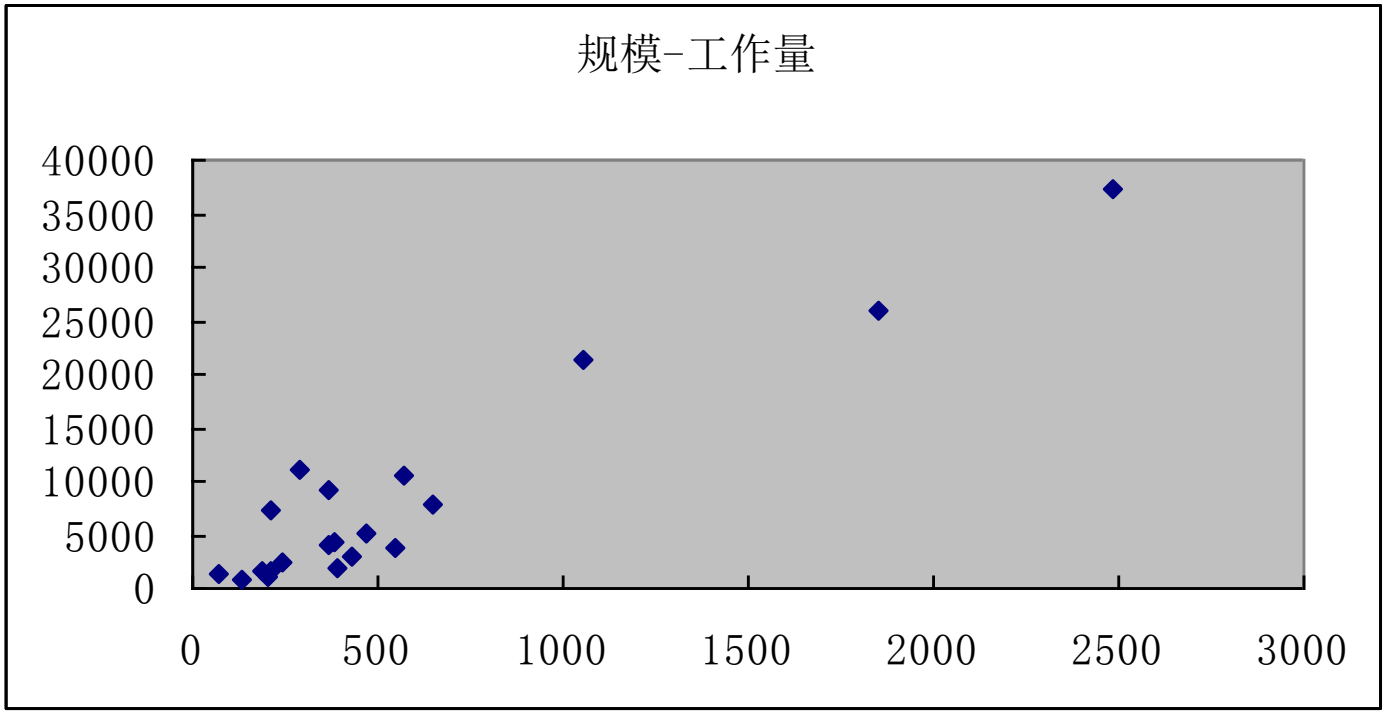
折线图


1000
100
10
1
264.5
147.5
409
304.5
124.5
52
23
65
39
21
立项策划需求阶段
设计阶段
编码阶段
测试阶段
验收发布阶段
实际工期(天)
工作量（小时）

散点图

Measures

- 散点图是用来表示两个因素间可能关系（如因果关系）的一种方法



5种对比关系

Measures

- 成分对比关系
  - 关注每一部分的大小占总数的百分比
- 类别对比关系
  - 关注类别的大小, 高低.
- 时间序列对比关系
  - 关注随时间的变化
- 频率对比关系
  - 关注有多少类别符合一个数字发展的范围
- 相关性对比关系
  - 关注几个变量之间的关系

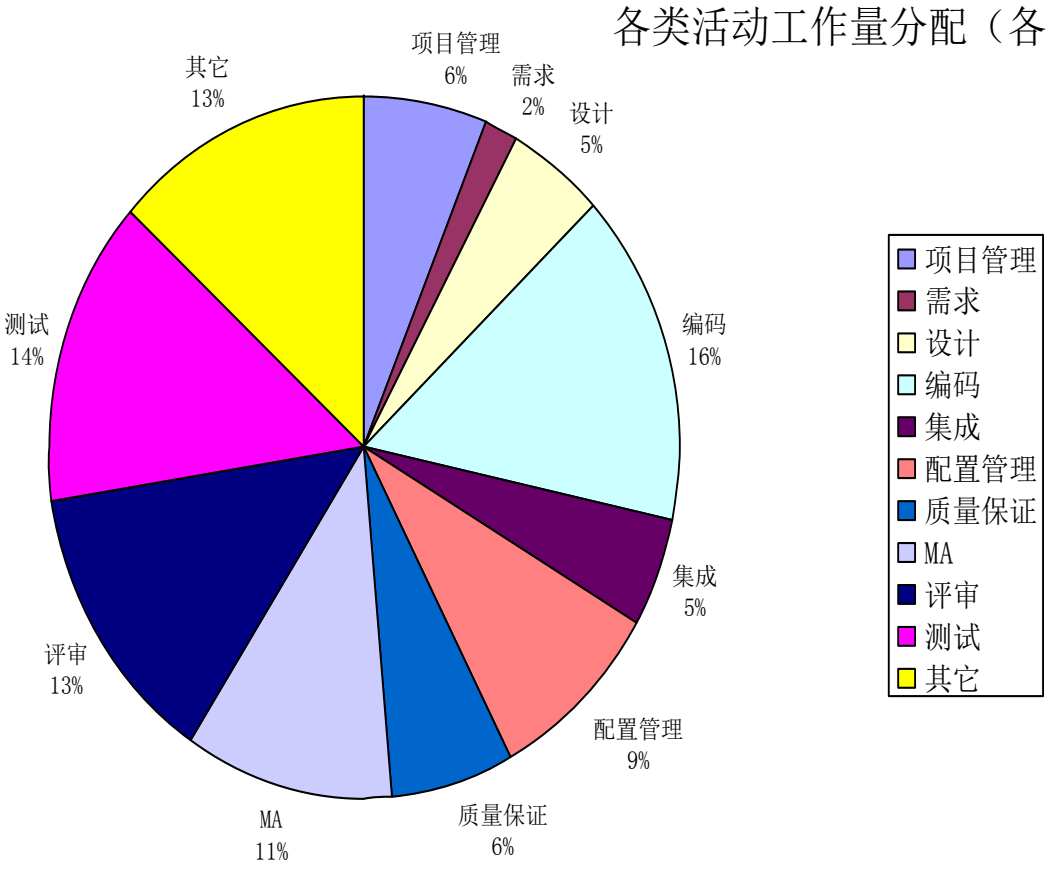
成分对比关系

Measures

- 采用饼图表示
- 比较的项一般不超过6种，如果超过6种则归为其他
- 最重要的项置于12点附近的位置，且用显著的颜色表示
- 如果各个部分相差不大，按从大到小或从小到大的顺序排列
- 在比较单一整体的各部分占整体的的比例时，饼图最有效，当比较多个整体的各部分的比例时，选择柱状图或条形图

指出下图中的缺陷

Measures

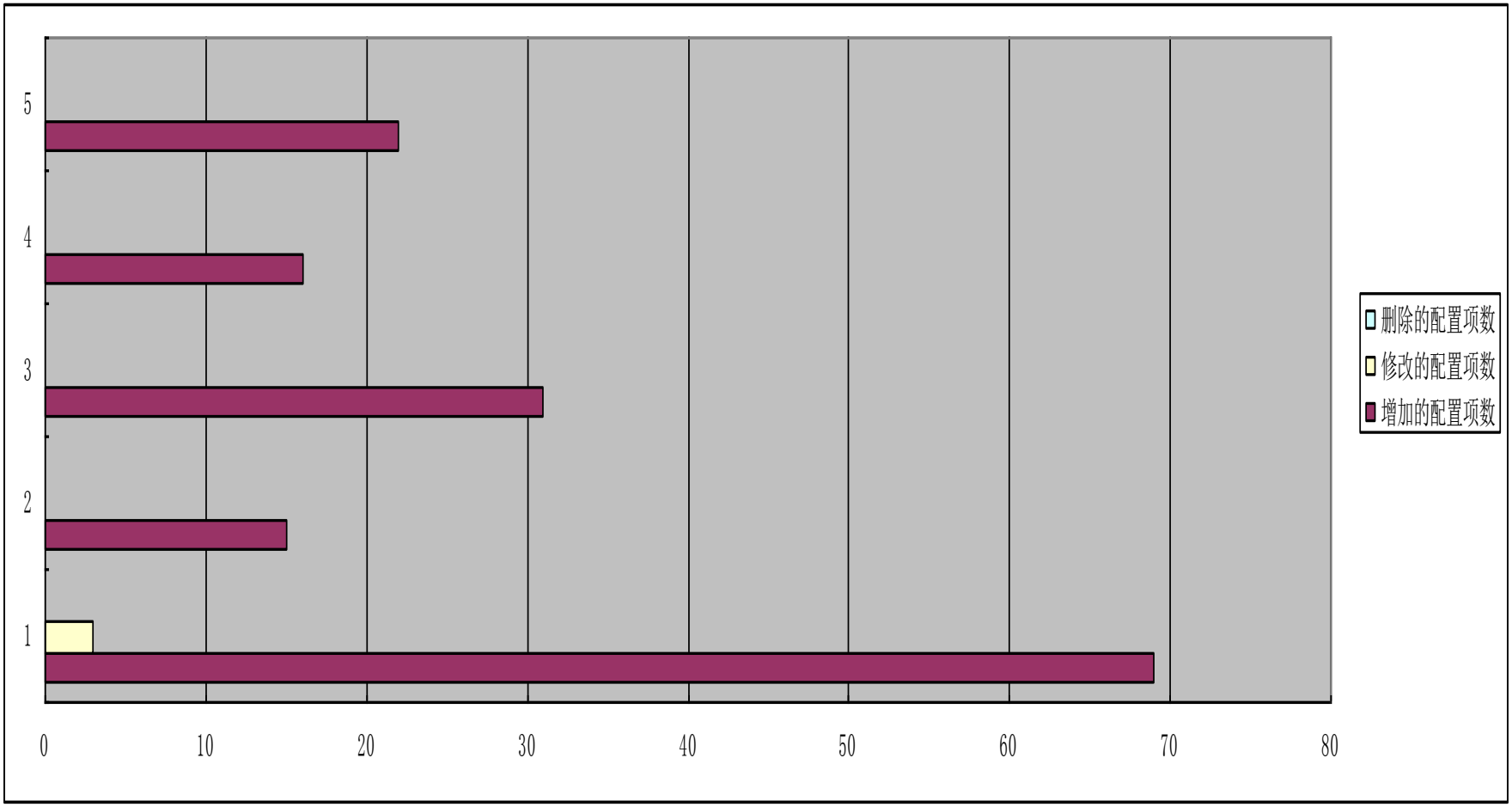


类别对比关系

Measures

- 一般采用条形图表示
- 要对各项次进行排序
- 条形之间的空间要小于条形的空间
- 在表示数字时，为了突出数字可以忽略小数点后的数字
- 尽量避免采用条形图表示时间变化，否则采用柱状图或线形图

指出下图中的缺陷



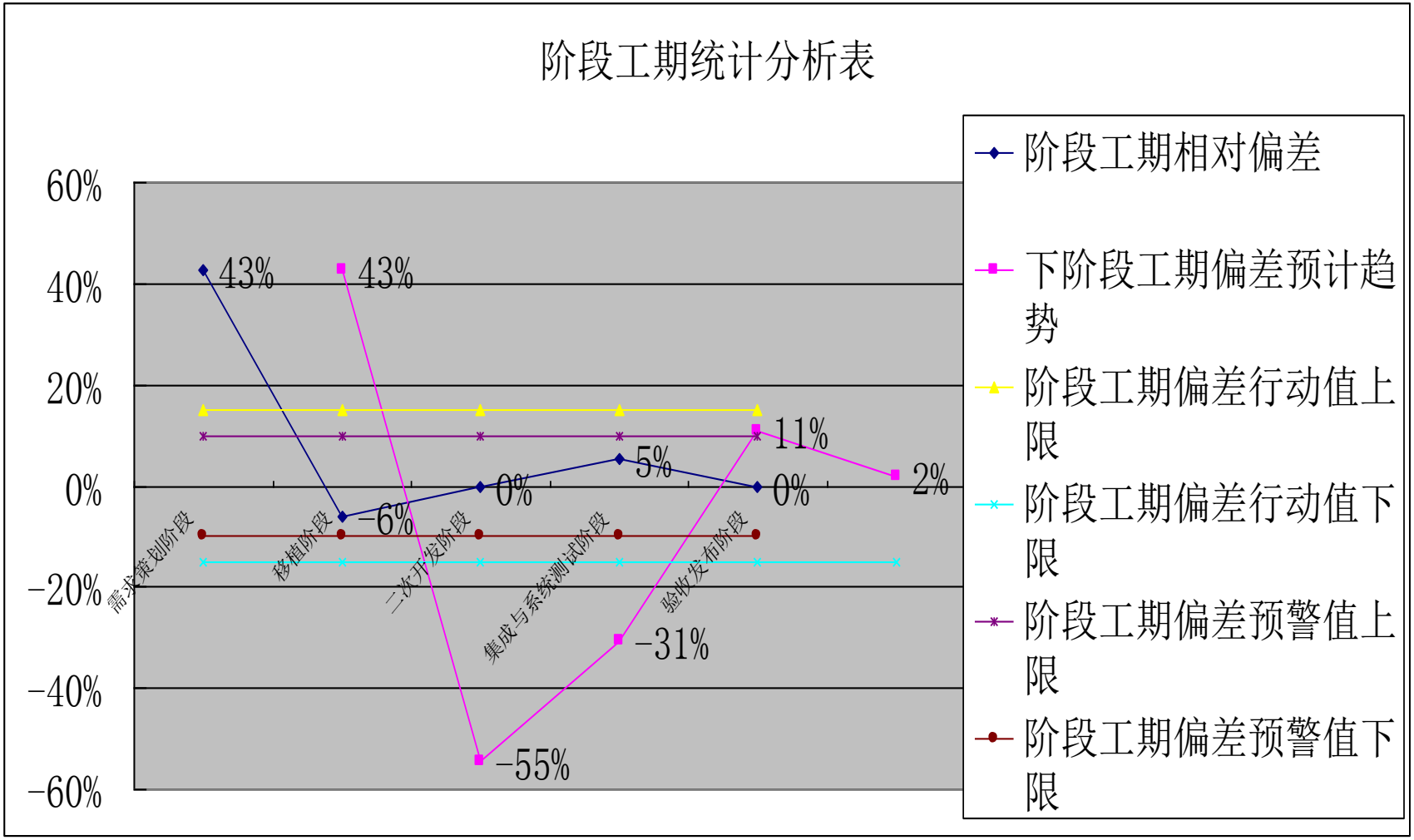
时间序列对比关系

Measures

- 可以用柱状图表和线形图表表示
- 柱状图注重程度和数量，线形图注重变化和变化趋势
- 如果时间点不超过8个，可以采用柱状图，否则采用线形图
- 柱体之间的距离要小于柱体本身的宽度
- 在画线形图时，趋势线要比底线粗，底线要比坐标线粗
- 在线形图中，尽可能少的在同一张图上有多条线

请指出下图中的缺陷

Measures



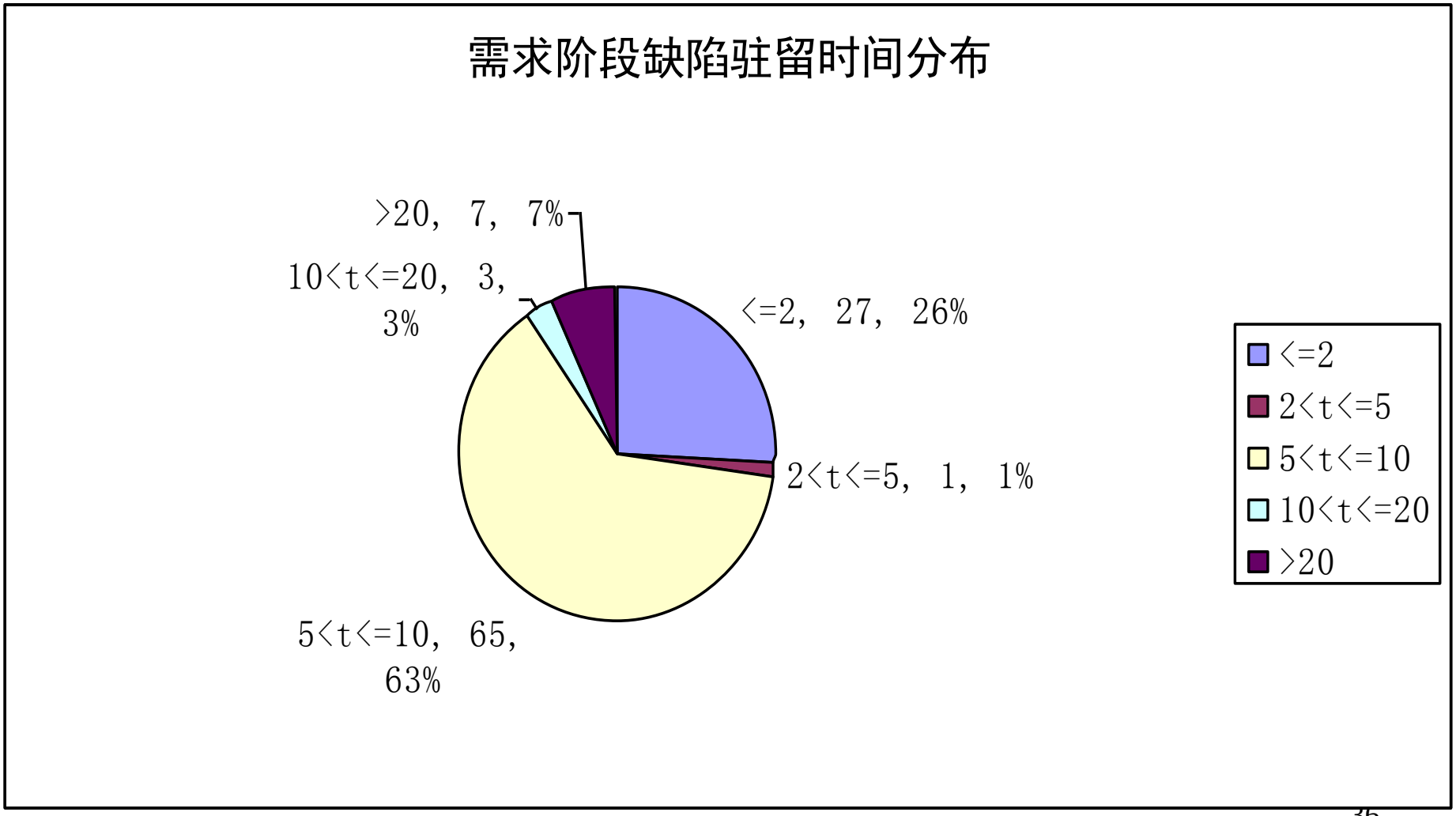
频率分布对比关系

Measures

- 表明在一个数字范围内有多少类别
- 可以通过柱状图或者线形图来表现，当范围数量比较少时，采用柱状图，当数量较多时采用线形图
- 分组的个数一般保持在5到20之间比较合适
- 分组的大小要相同

请指出下图中的缺陷

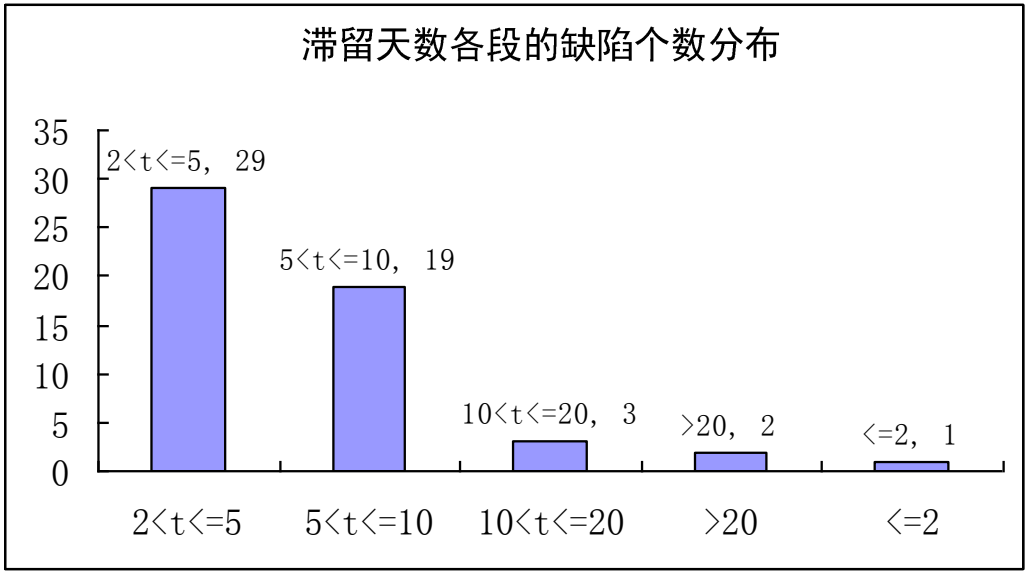
Measures



缺陷的滞留时间分析—单项目

Measures

缺陷滞留时间(天)	滞留天数各段的缺陷个数	所占比例
2<t<=5	29	53.70%
5<t<=10	19	35.19%
10<t<=20	3	5.56%
>20	2	3.70%
<=2	1	1.85%
总合	54	

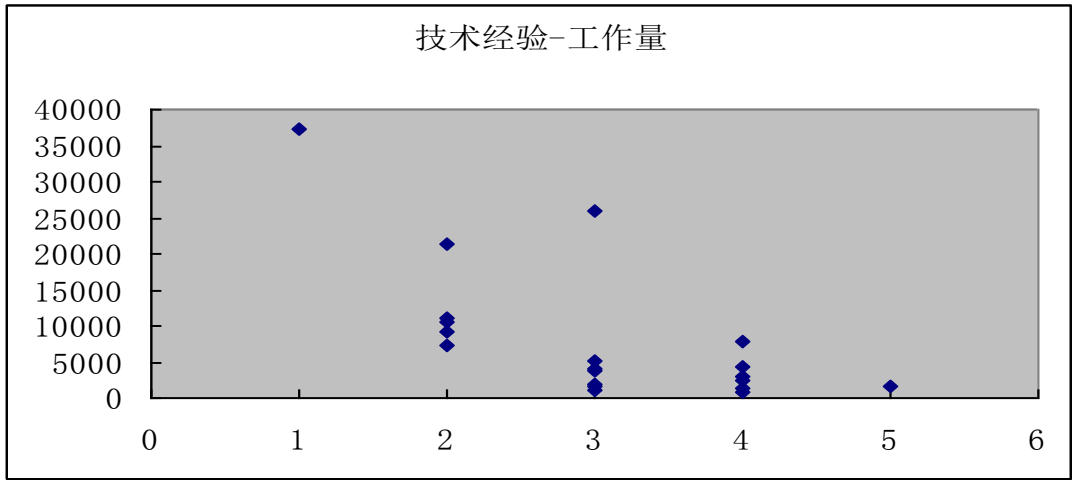


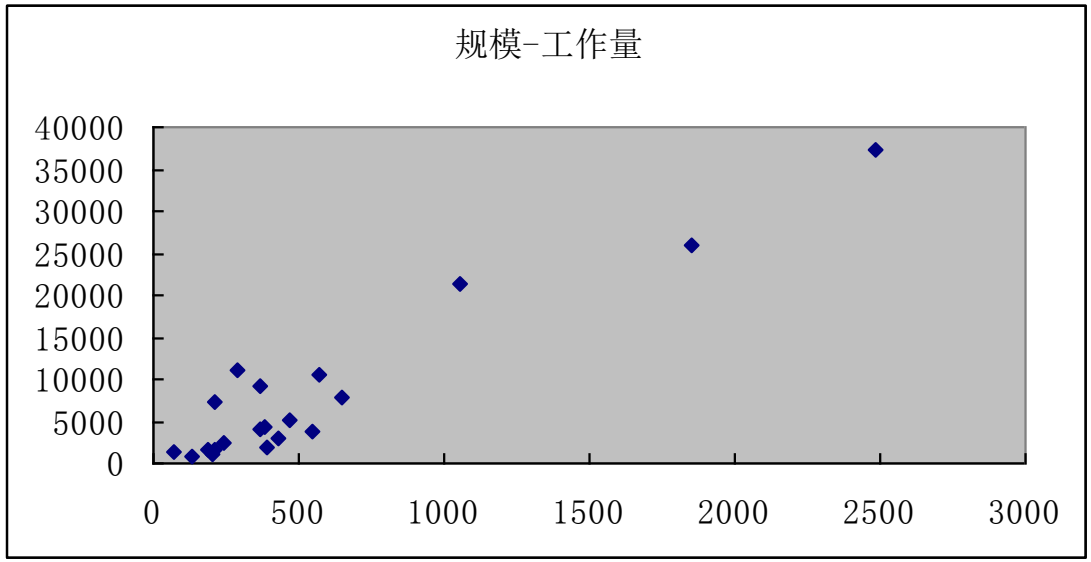
相关性对比分析

Measures

- 表明的是2个变量之间的关系符合或不符合某种模式
- 可以采用散点图或双条形图
- 如果数字本身看不到明显的相关性，观察其对数的相关性

变量相关性分析


技术经验-工作量
40000
35000
30000
25000
20000
15000
10000
5000
0
0
1
2
3
4
5
6


规模-工作量
40000
35000
30000
25000
20000
15000
10000
5000
0
0
500
1000
1500
2000
2500
3000

Measures

指示器的设计要点

指示器

Measures

- 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度量元和/或派生度量元而产生的。
- 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：
  - 支持指定的信息需求的分析
  - 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、指定的限定值或阀值范围。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。

指示器的设计要点

Measures

- 选择合适的图表类型
  - 根据目的选择图表类型
  - 根据数据量的大小选择图标类型
- 先排序再分析
- 选择合适的数据分组
- 设置合适的时间刻度
- 设置合适的控制线
- 减少网格线
- 每个表的标题，系列名称，类别名称，%显示等要设置全面准确

选择适合的图表类型

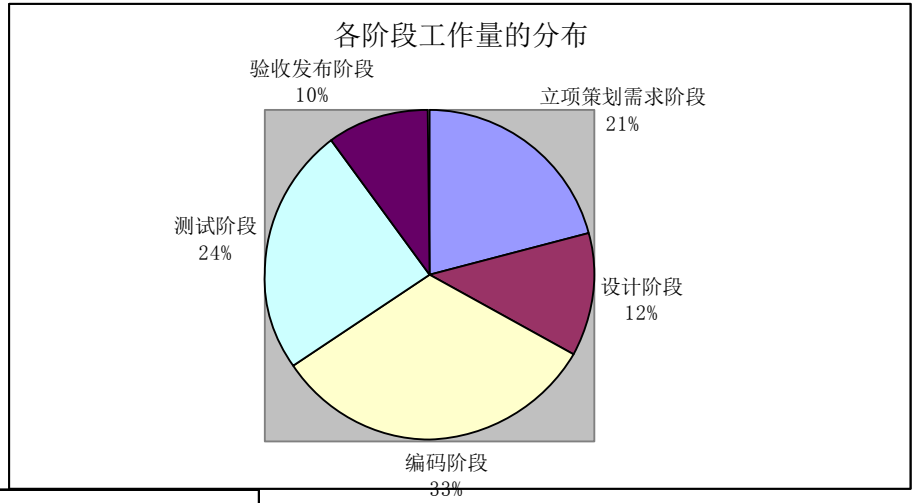
Measures

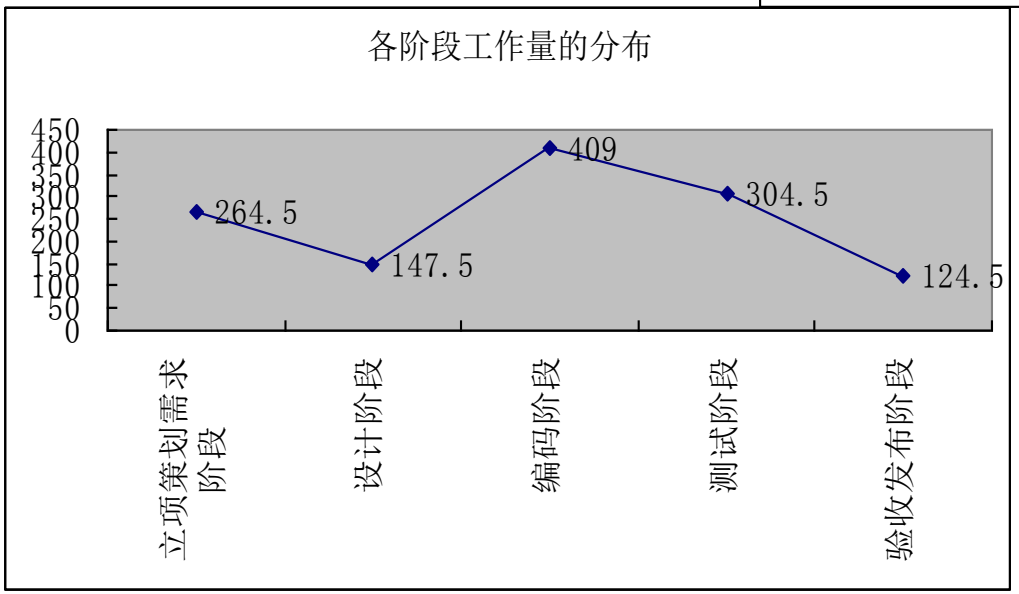
	成分	类别对比	时间序列	频率分布	相关性
饼图	✓				
条形图		✓			✓
柱状图			✓	✓	
线性图			✓	✓	
散点图					✓

根据分析目的选择分析图形-目的不同，图形不同

Measures

- 对于同样的数据,选择哪种图表更合适呢?
  - 你的分析目的是什么?

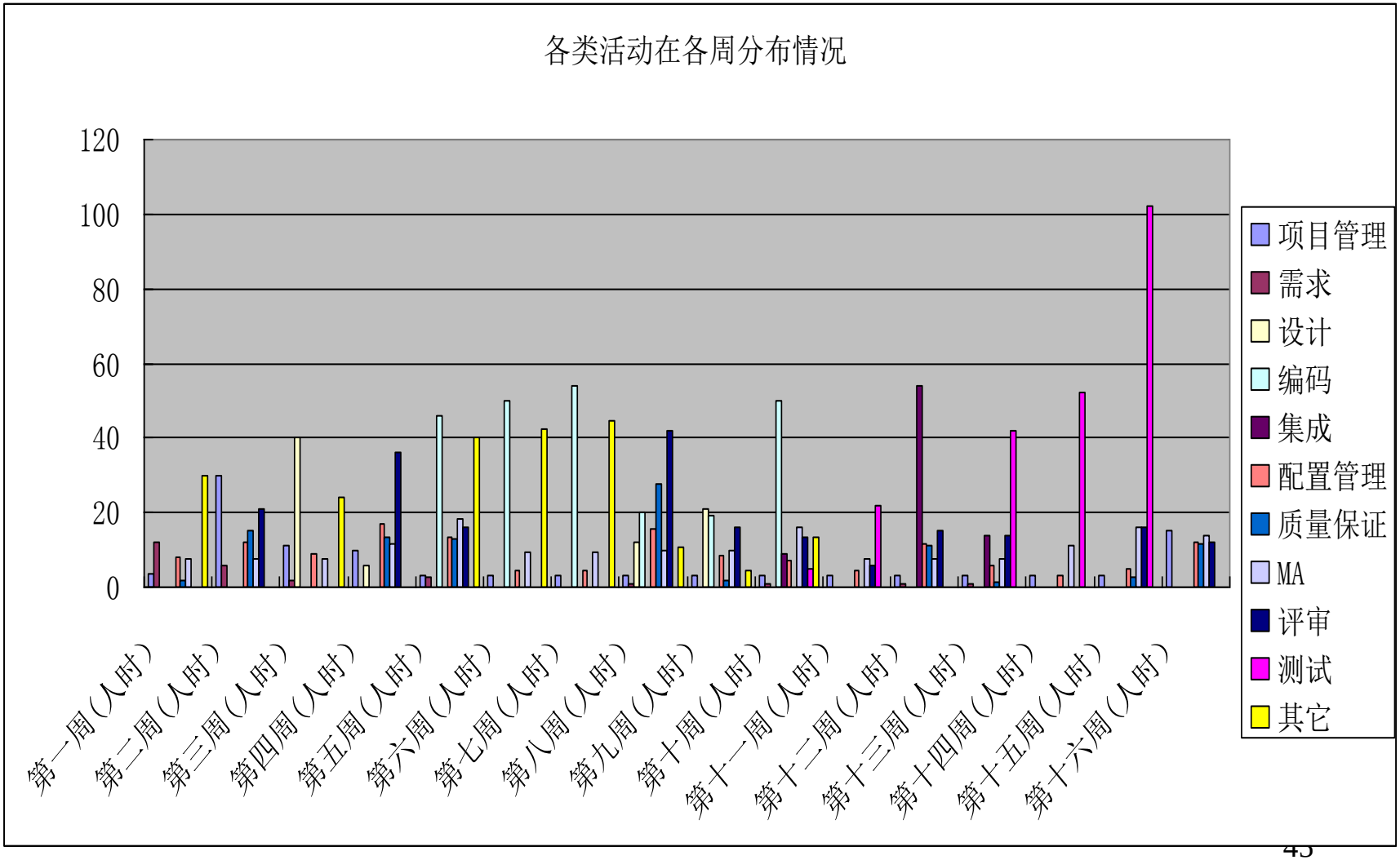




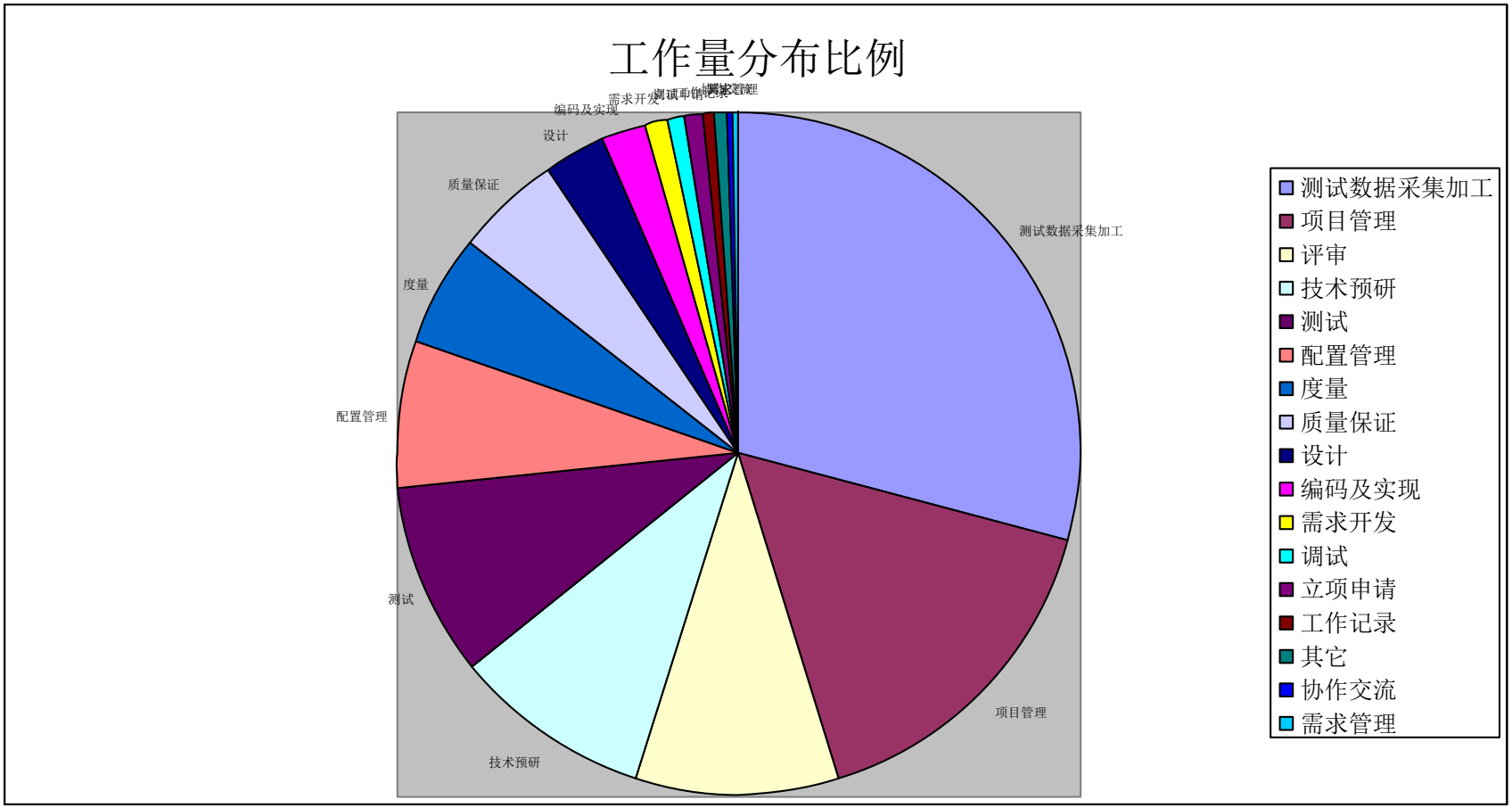
阶段名称	工作量（小时）
立项策划需求阶段	264.5
设计阶段	147.5
编码阶段	409
测试阶段	304.5
验收发布阶段	124.5

根据分析目的选择分析图形-目的要明确

Measures

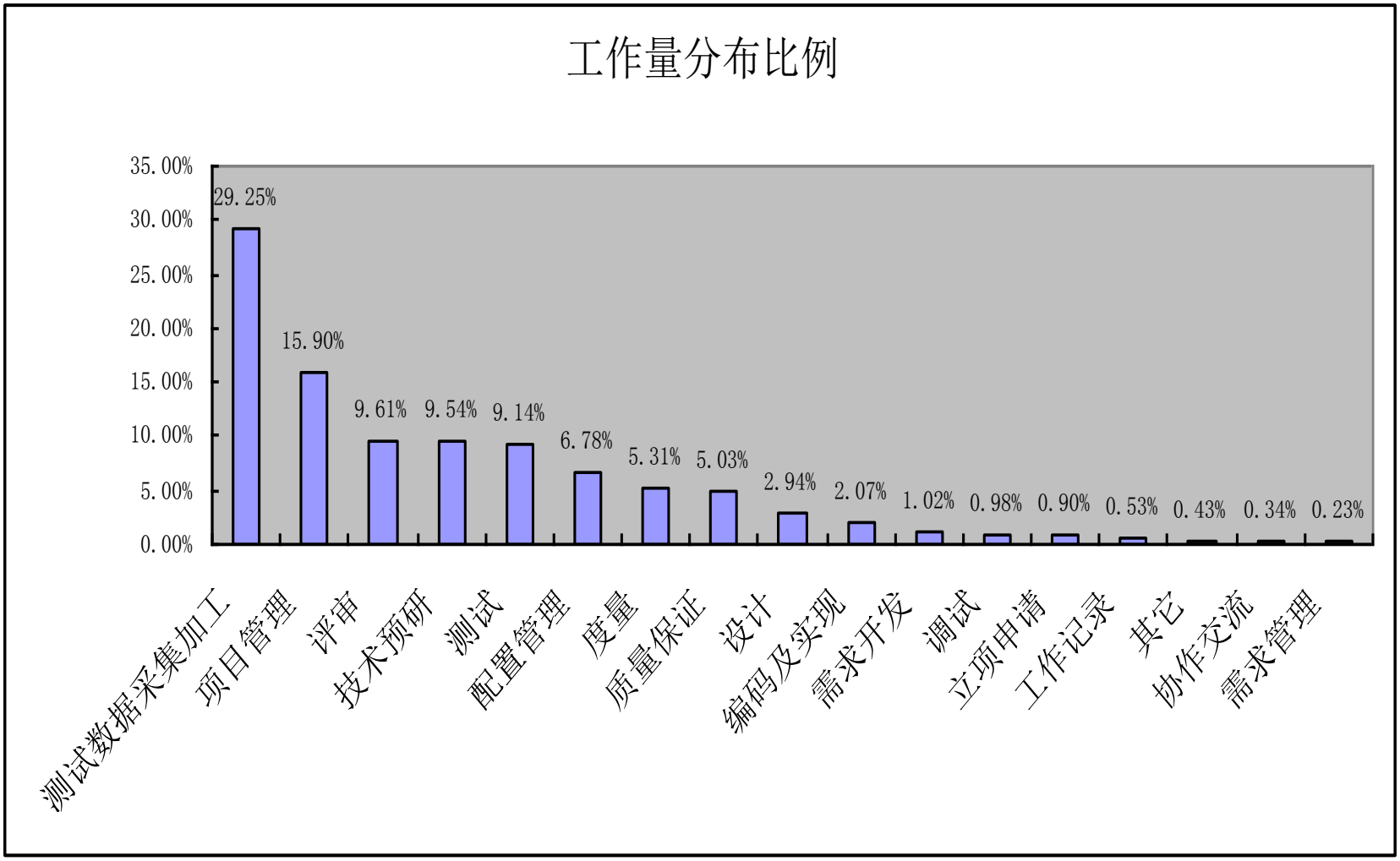


根据数据量的大小选择图形类型-1



- 当需要比较的项比较多时,可以考虑采用直方图而不是饼图

根据数据量的大小选择图形类型-2



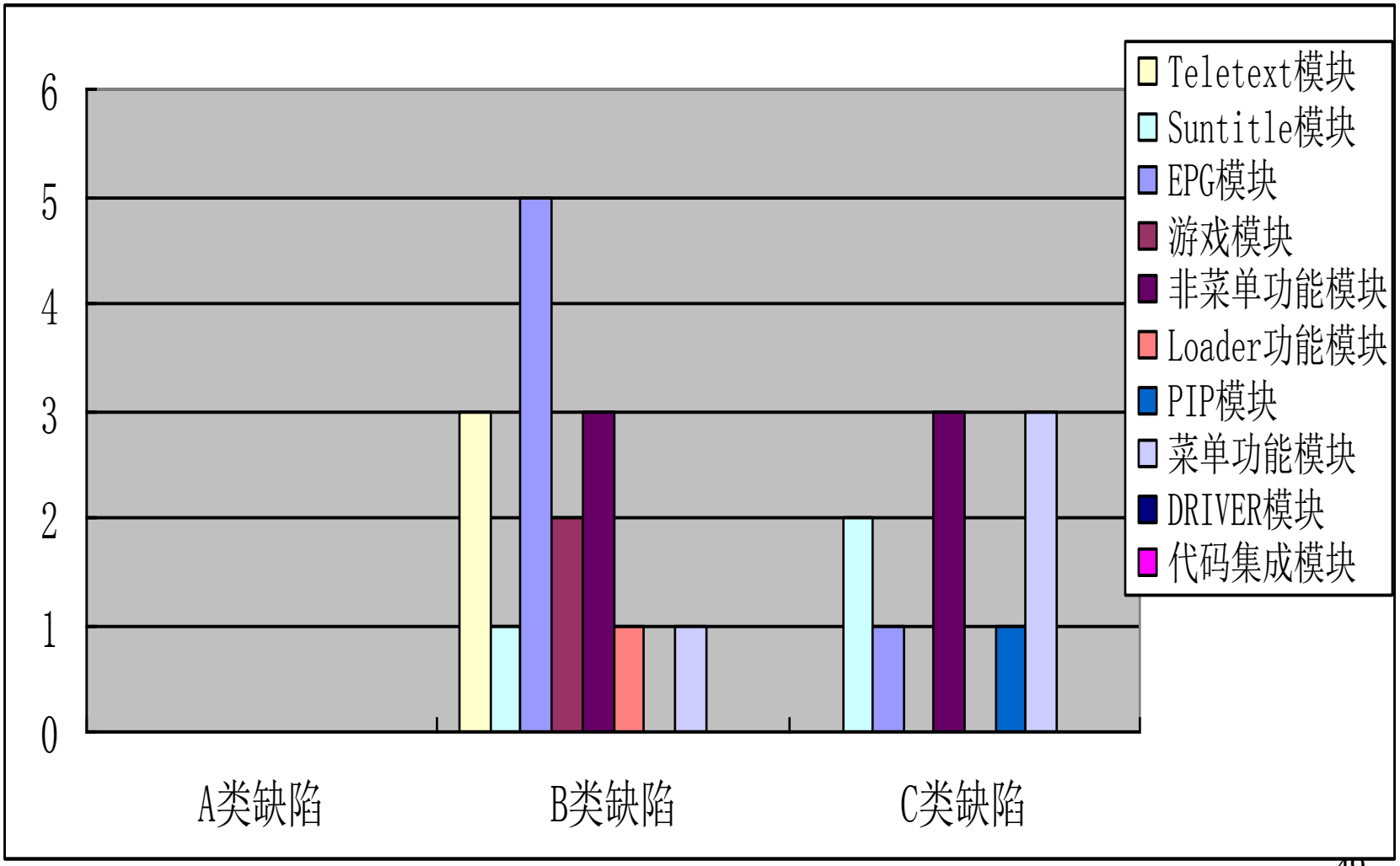
先排序再分析--原始度量数据

Measures

模块名称	A类缺陷	B类缺陷	C类缺陷	各模块的缺陷总数	各模块的代码行数	各模块的缺陷密度（千行bug数）
DRIVER模块	0	0	0	0	2800	0.00
Loader功能模块	0	1	0	1	2000	0.50
菜单功能模块	0	1	3	4	28251	0.14
非菜单功能模块	0	3	3	6	6500	0.92
EPG模块	0	5	1	6	5260	1.14
Teletext模块	0	3	0	3	1032	2.91
Suntitle模块	0	1	2	3	1500	2.00
PIP模块	0	0	1	1	5832	0.17
游戏模块	0	2	0	2	2116	0.95
代码集成模块	0	0	0	0	2120	0.00
整体的代码行数	57411					
整体的缺陷密度（千行bug数）	0.45					

先排序再分析--改造前的分析表格

Measures



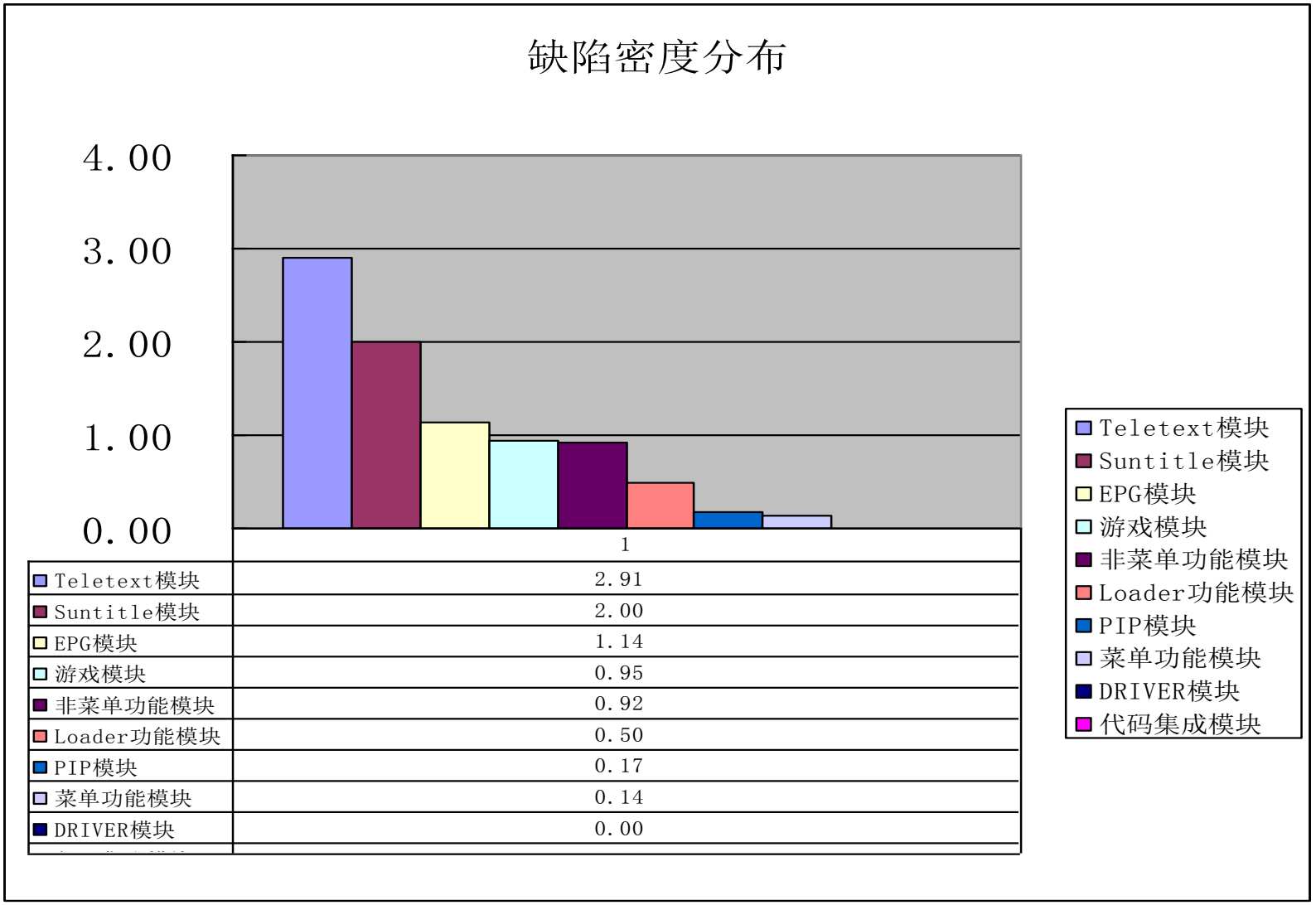
先排序再分析--数据排序后

Measures

模块名称	A类缺陷	B类缺陷	C类缺陷	各模块的缺陷总数	各模块的代码行数	各模块的缺陷密度（千行bug数）
Teletext模块	**0**	**3**	**0**	**3**	1032	**2.91**
Suntitle模块	**0**	**1**	**2**	**3**	**1500**	**2.00**
EPG模块	**0**	**5**	**1**	**6**	**5260**	**1.14**
游戏模块	**0**	**2**	**0**	**2**	2116	**0.95**
非菜单功能模块	**0**	**3**	**3**	**6**	**6500**	**0.92**
Loader功能模块	**0**	**1**	**0**	**1**	**2000**	**0.50**
PIP模块	**0**	**0**	**1**	**1**	5832	**0.17**
菜单功能模块	**0**	**1**	**3**	**4**	**28251**	**0.14**
DRIVER模块	**0**	**0**	**0**	**0**	**2800**	**0.00**
代码集成模块	**0**	**0**	**0**	**0**	2120	**0.00**
整体的代码行数	57411					
整体的缺陷密度（千行bug数）	0.45					

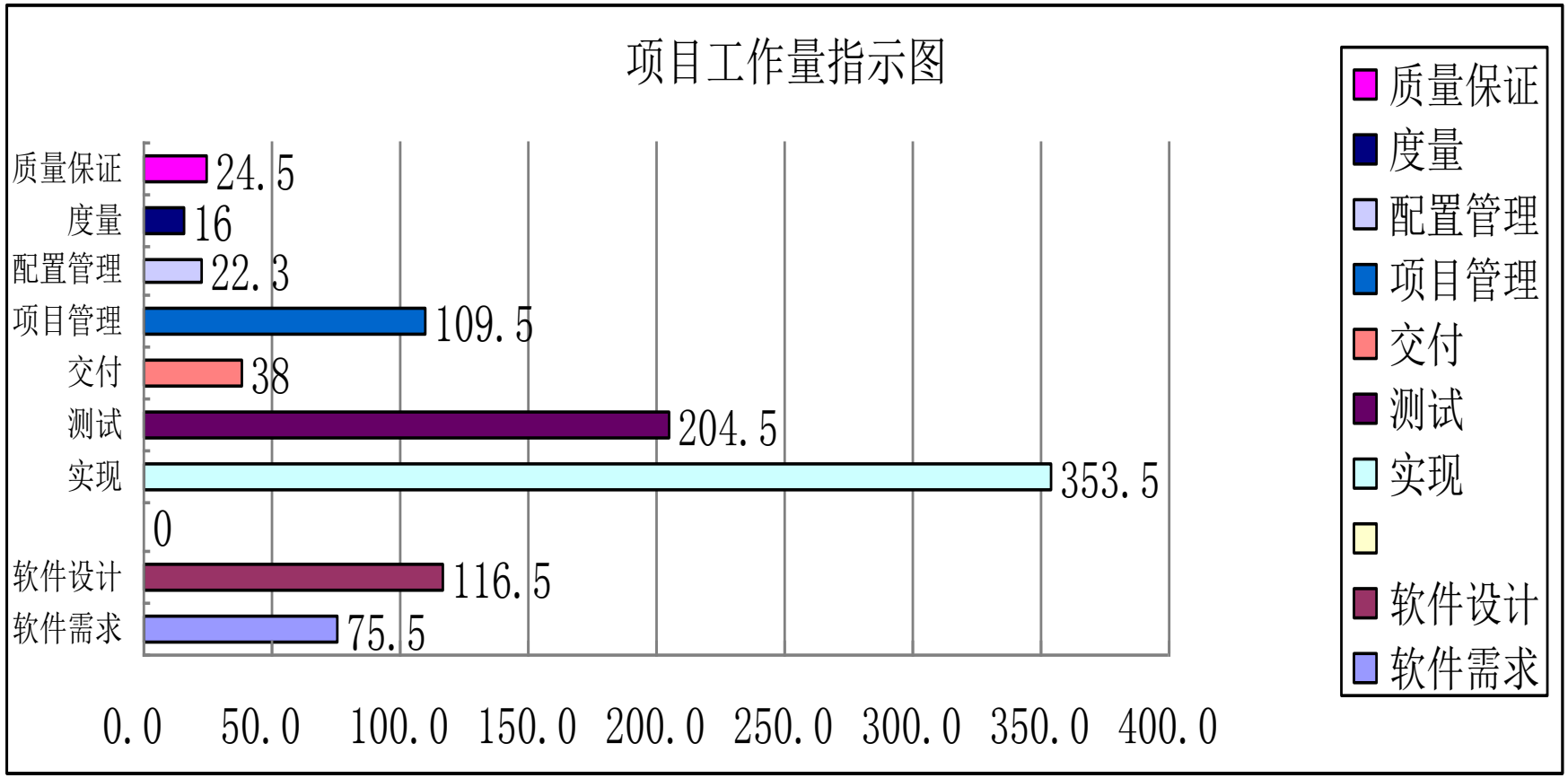
先排序再分析--重新分析的表格

Measures

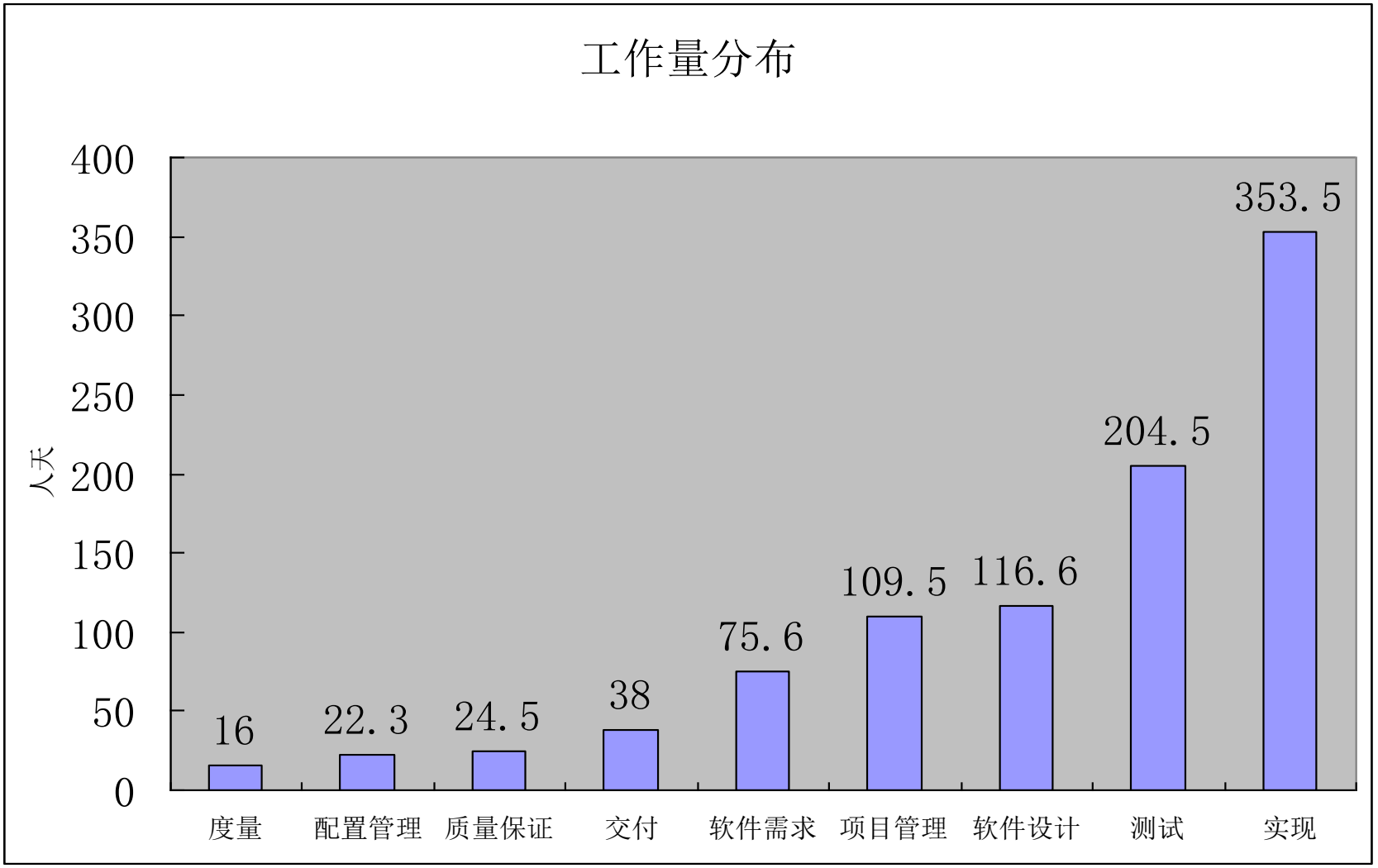


	1
Teletext模块	2.91
Suntitle模块	2.00
EPG模块	1.14
游戏模块	0.95
非菜单功能模块	0.92
Loader功能模块	0.50
PIP模块	0.17
菜单功能模块	0.14
DRIVER模块	0.00

先排序再分析案例-ZC_ZKZ项目工作量分布

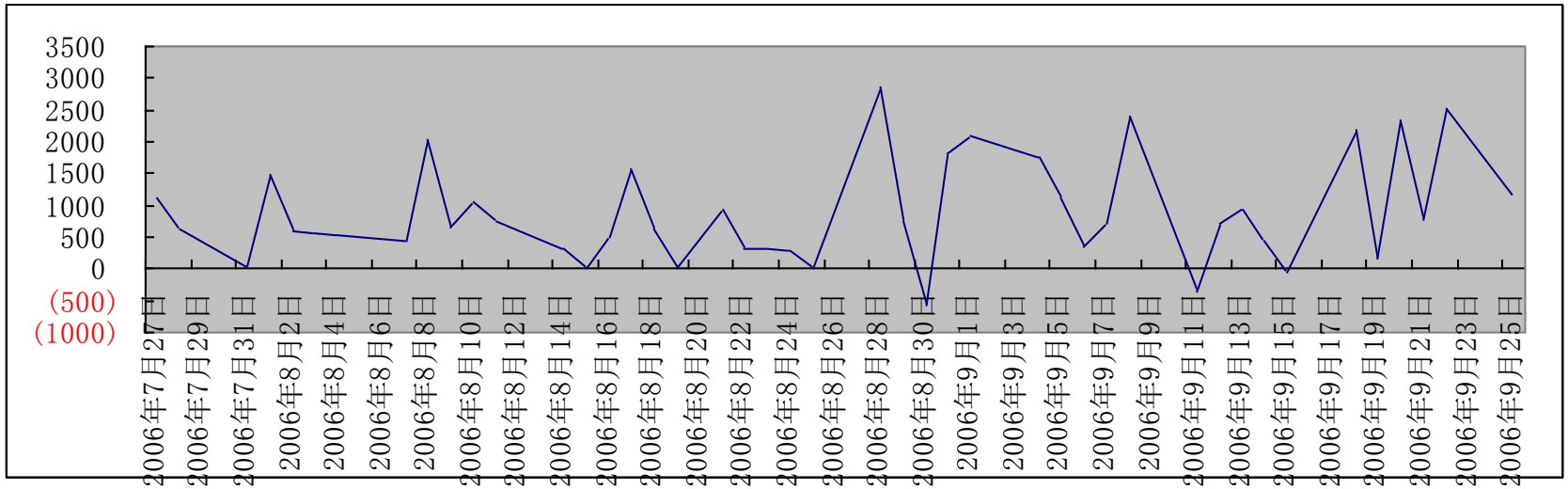


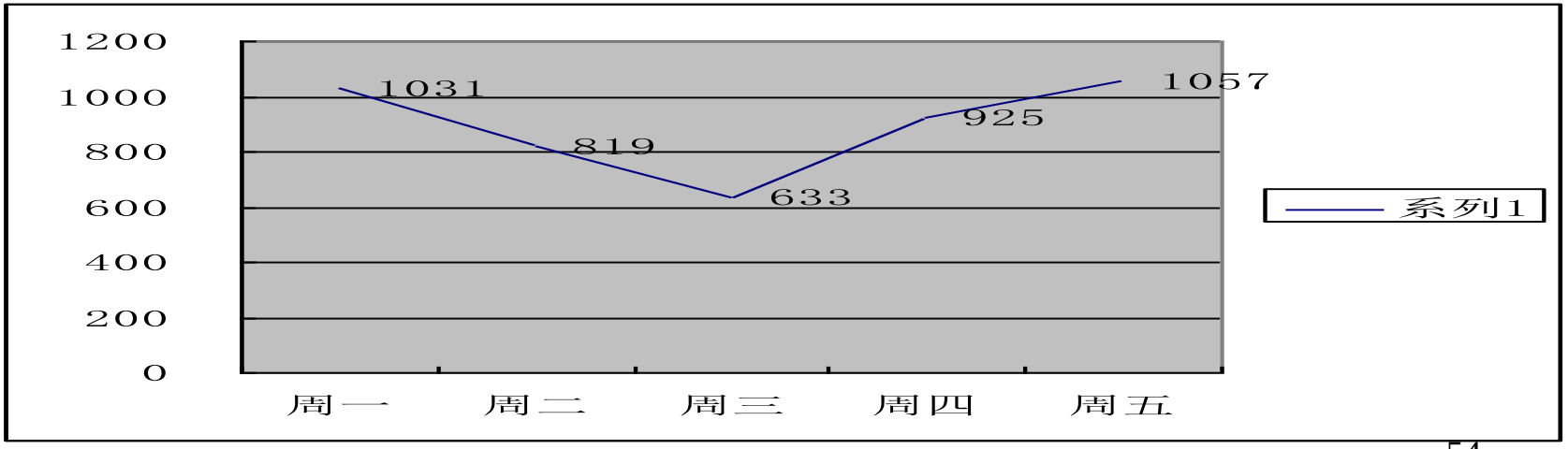
先排序再分析--重新分析的表格



选择合适的数据分组

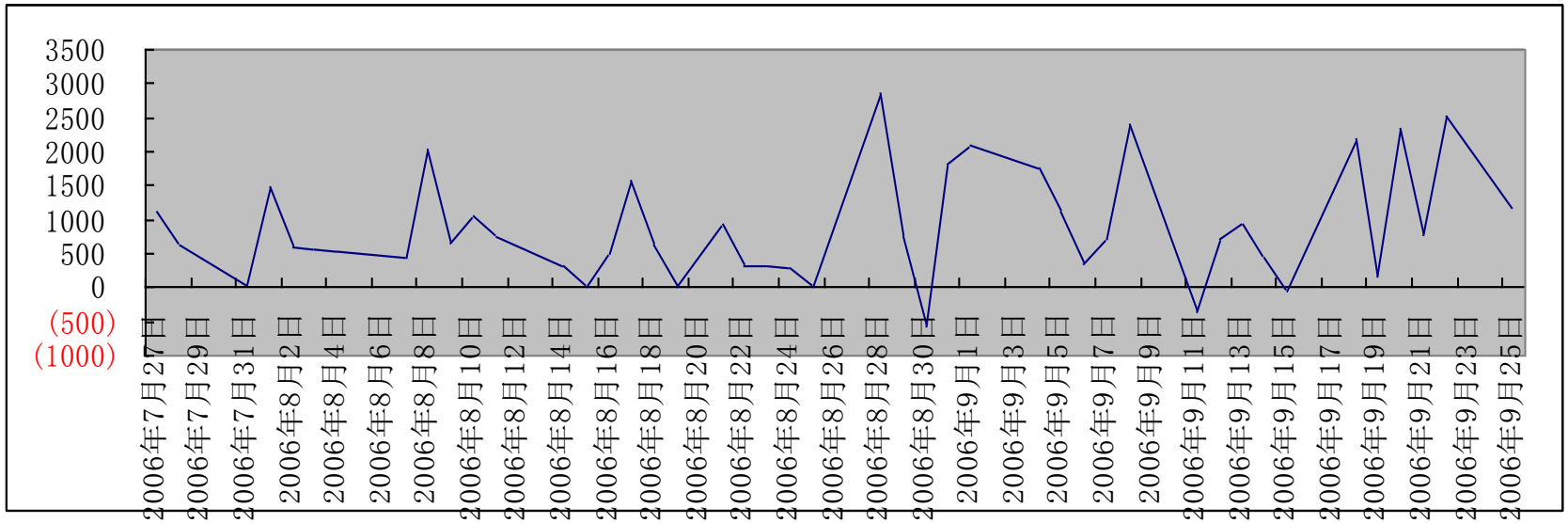
Measures

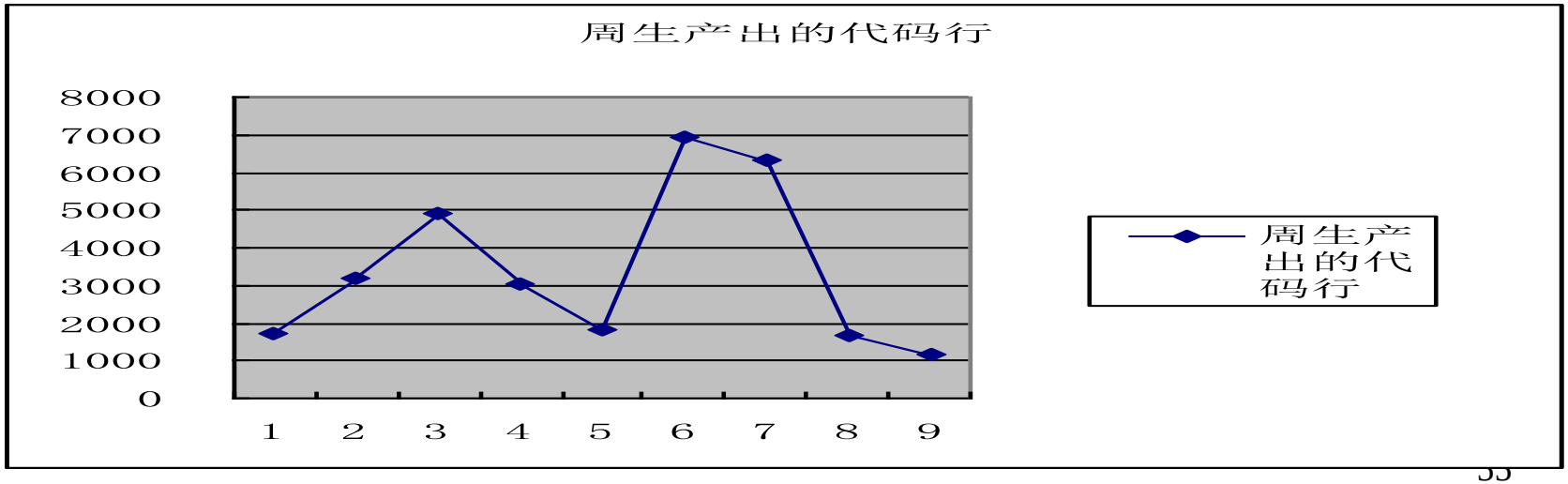




设置合适的时间刻度

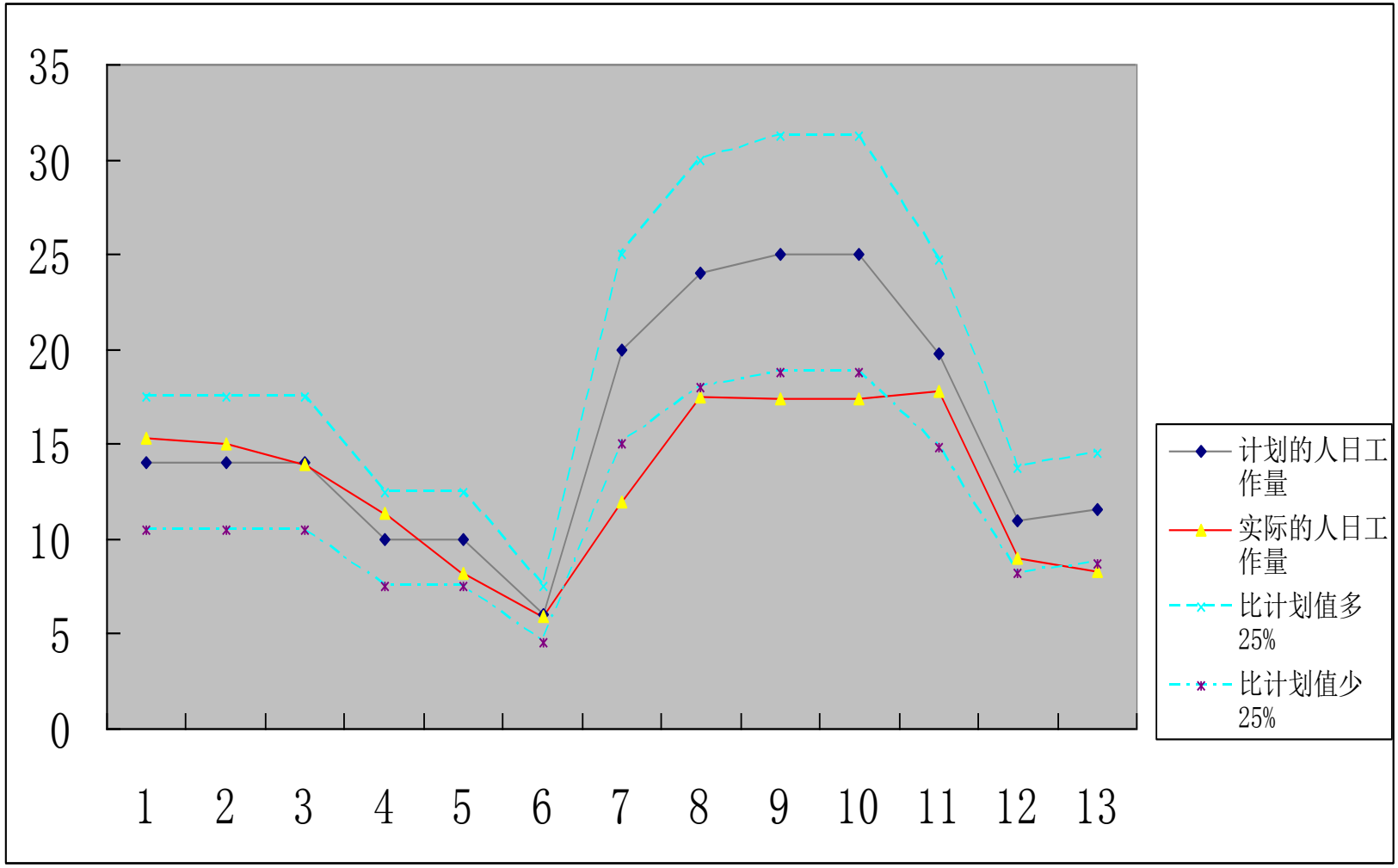
Measures



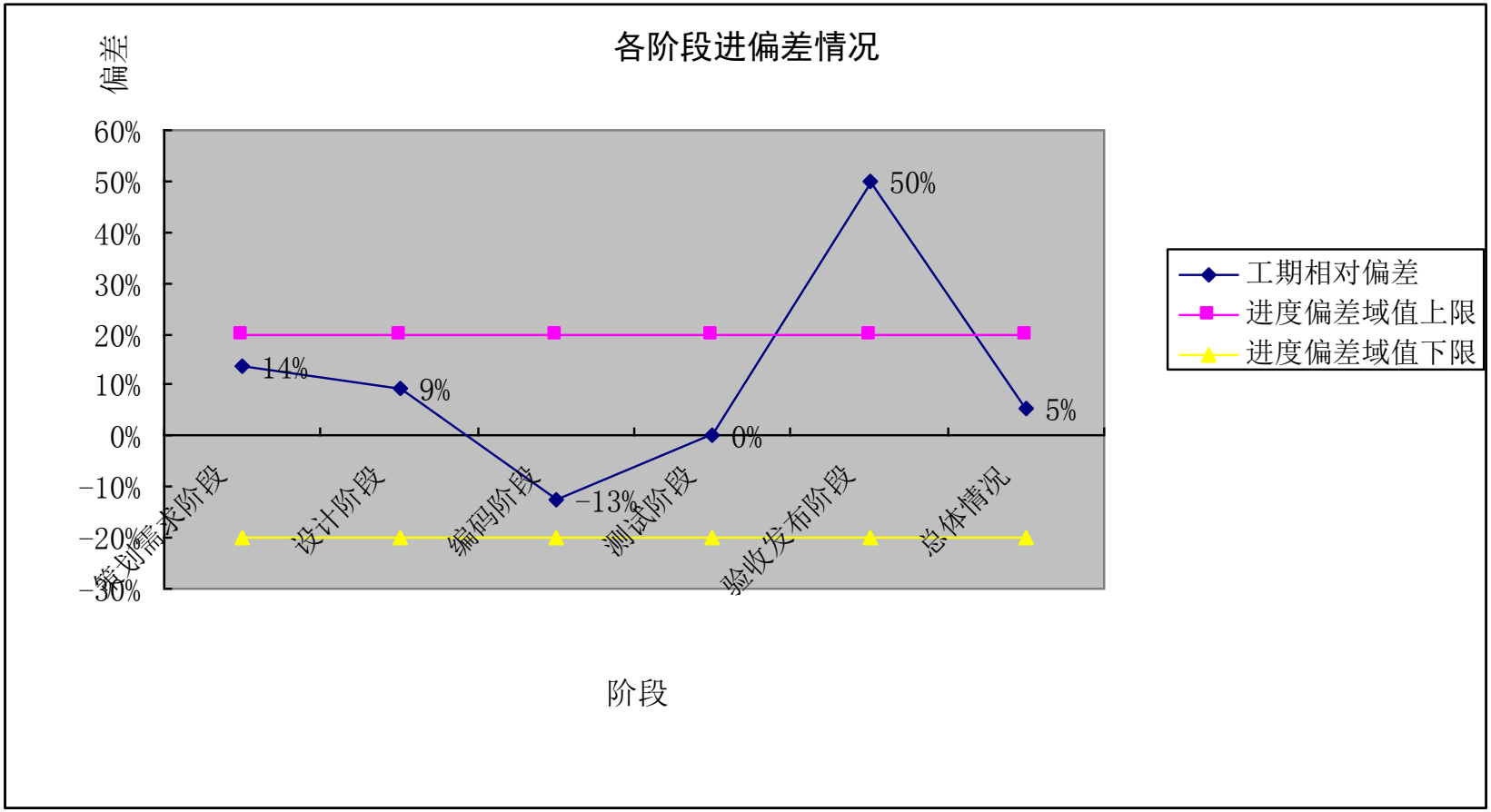


设置合适的控制线-1

Measures

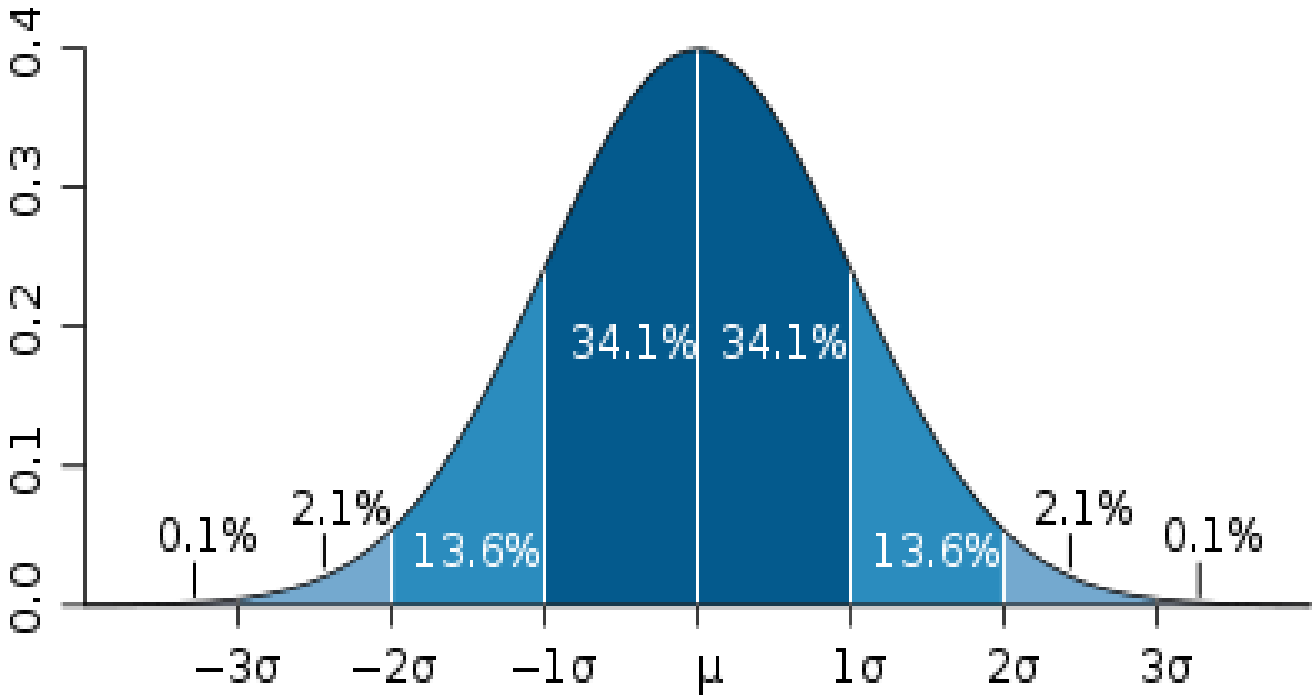


设置合适的控制线-2


各阶段进偏差情况
偏差
60%
50%
40%
30%
20%
10%
0%
-10%
-20%
-30%
14%
9%
-13%
0%
50%
5%
策划需求阶段
设计阶段
编码阶段
测试阶段
验收发布阶段
总体情况
阶段
工期相对偏差
进度偏差域值上限
进度偏差域值下限

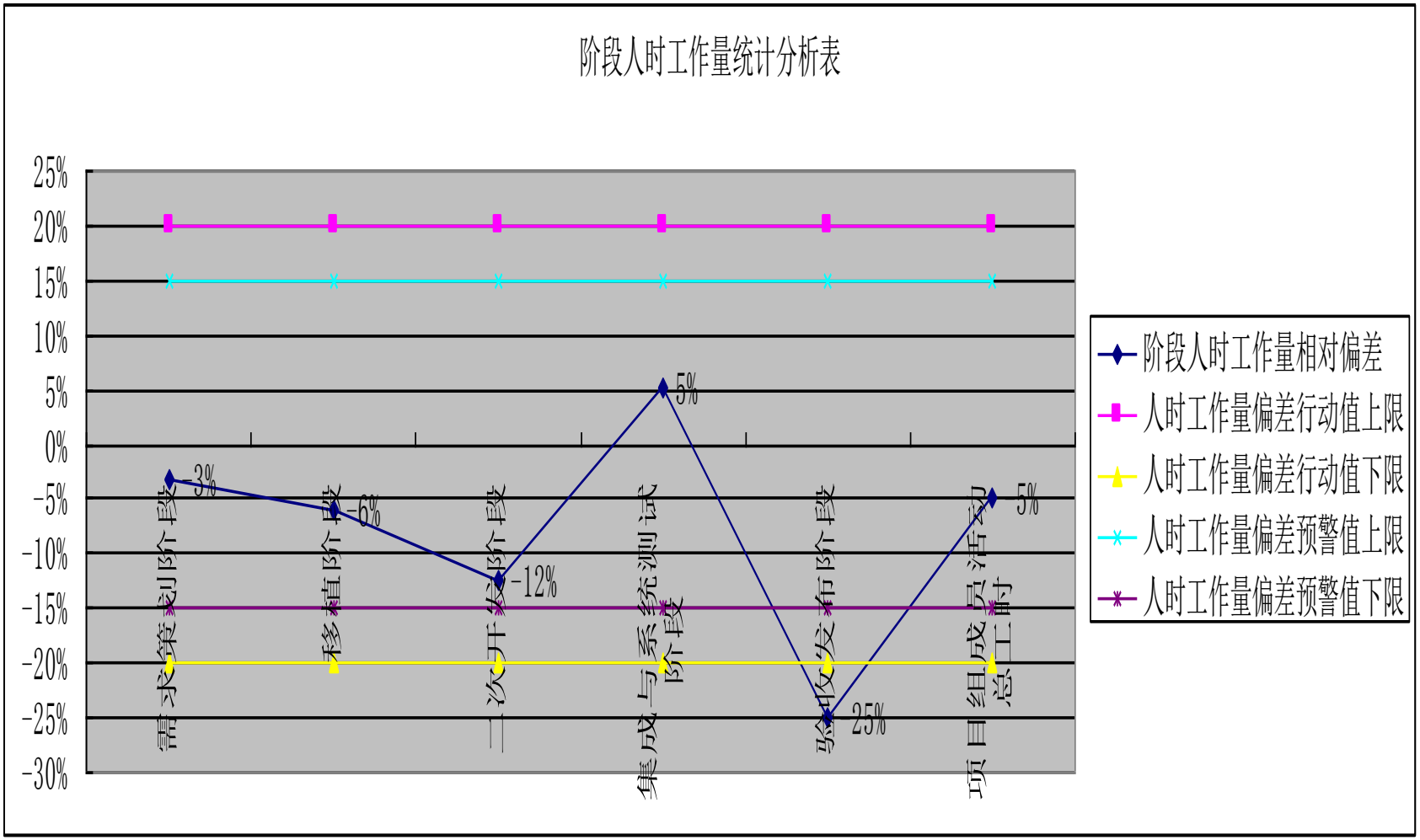
设置合适的控制线-3

正态分布时3Sigma的概率



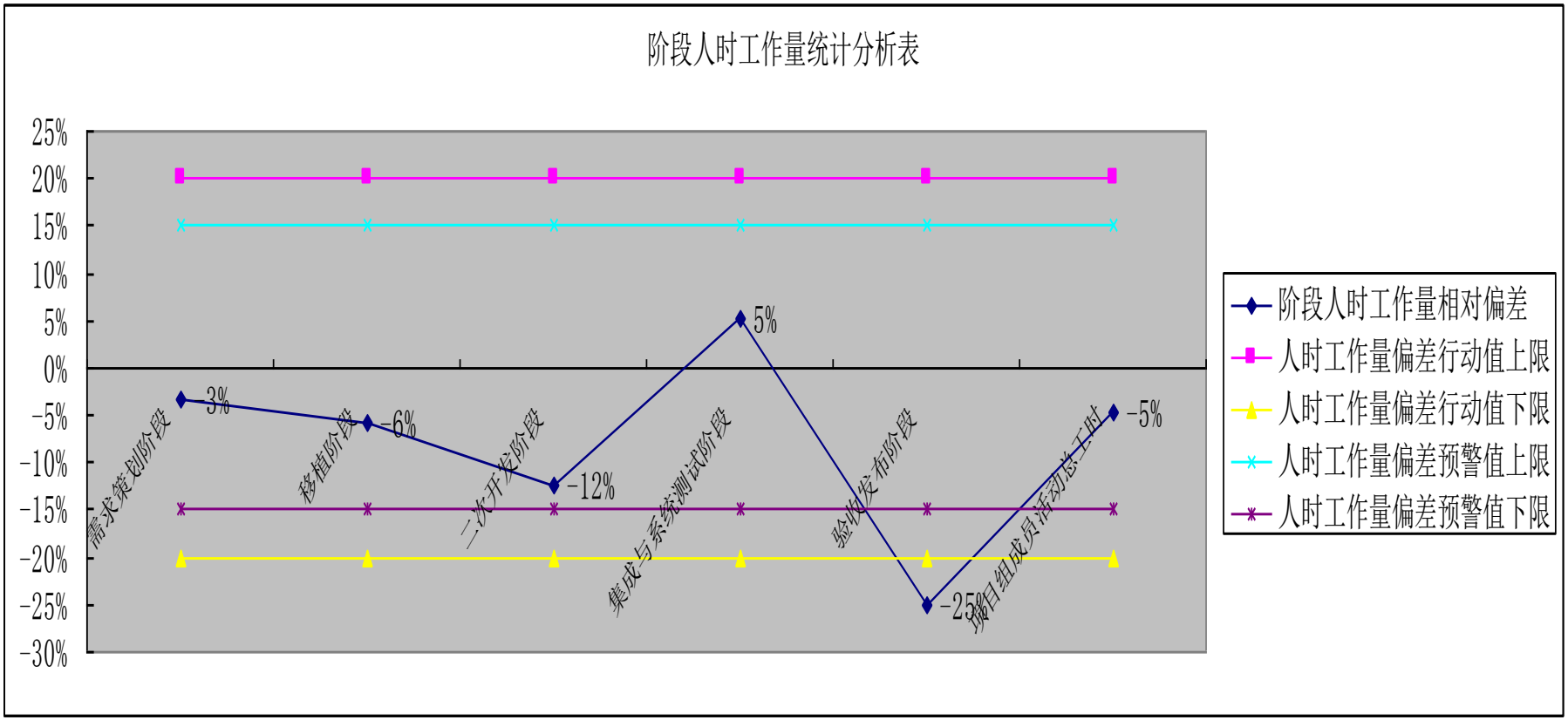
减少网格线-1

Measures



减少网格线-2

Measures



Measures

如何定义度量体系？

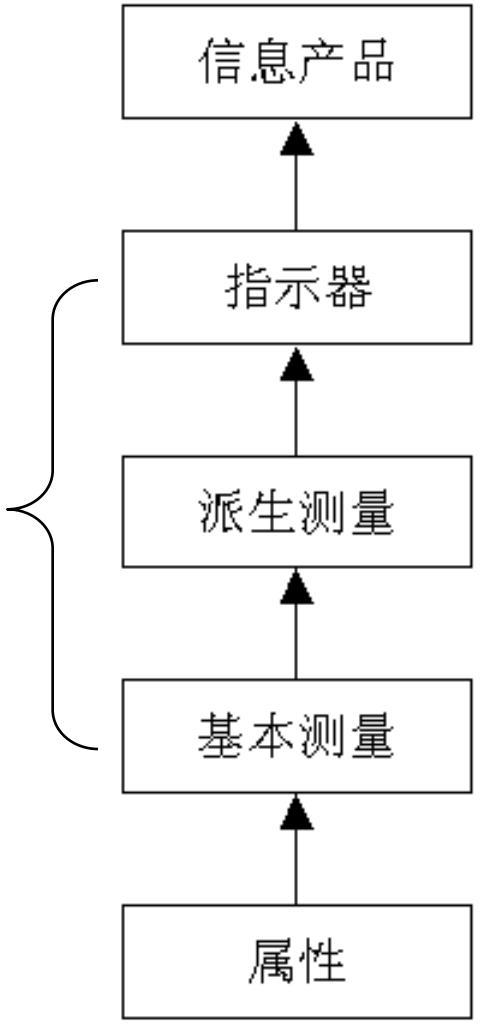
度量构造

度量元

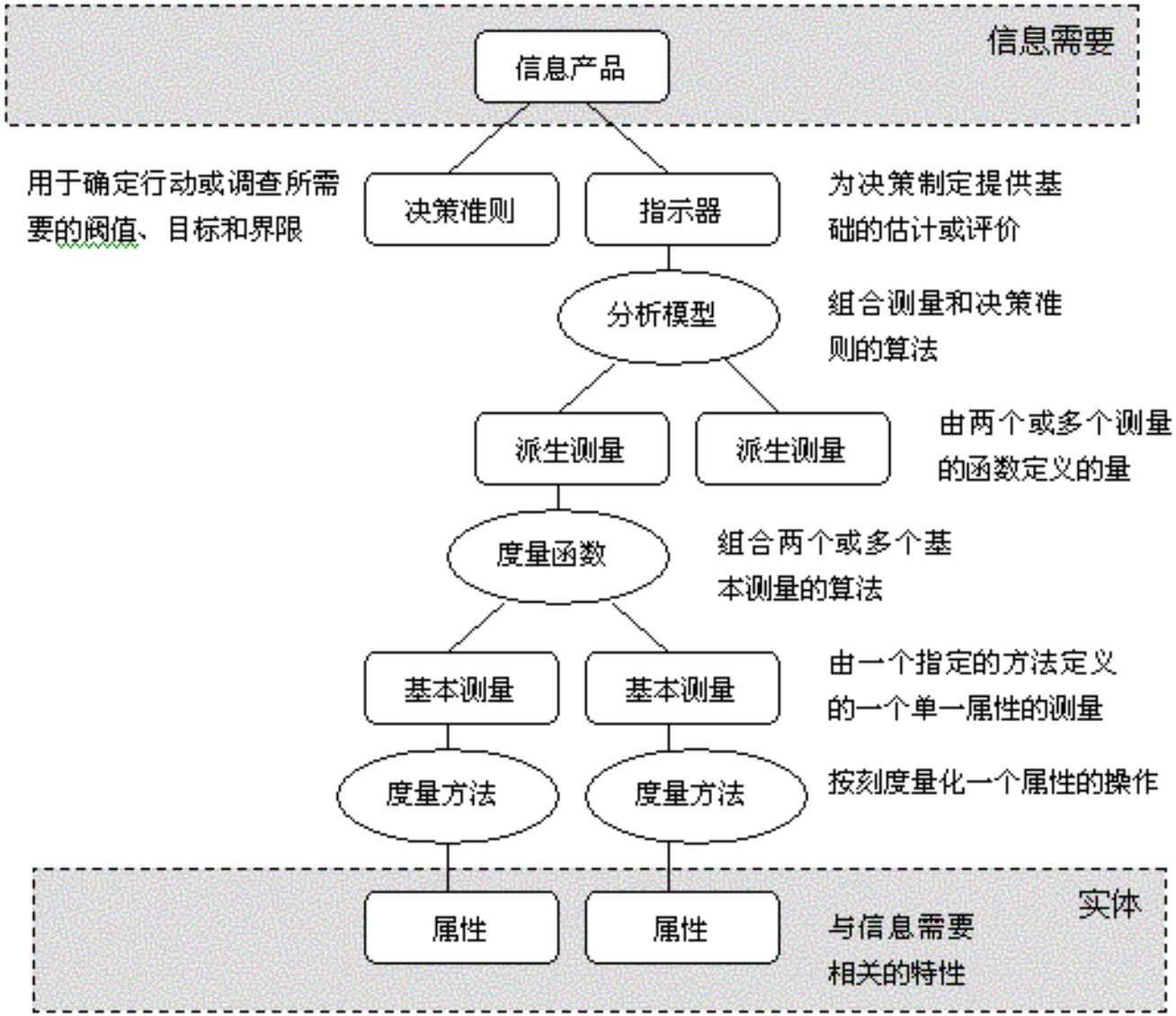
度量构造

Measures

- 度量构造是将用来满足指定信息需求所要测量的对象联系起来的详细结构。
- 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。
- 一个单一的度量构造可能包括三种测量类型或层次：基本度量元、派生度量元和指示器。
- 度量计划人员必须指定要使用的度量构造的细节以及在度量计划中数据采集、分析和汇报的步骤。
- 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求结合得就越好，项目经理也就越容易制定出可靠的客观的决策。

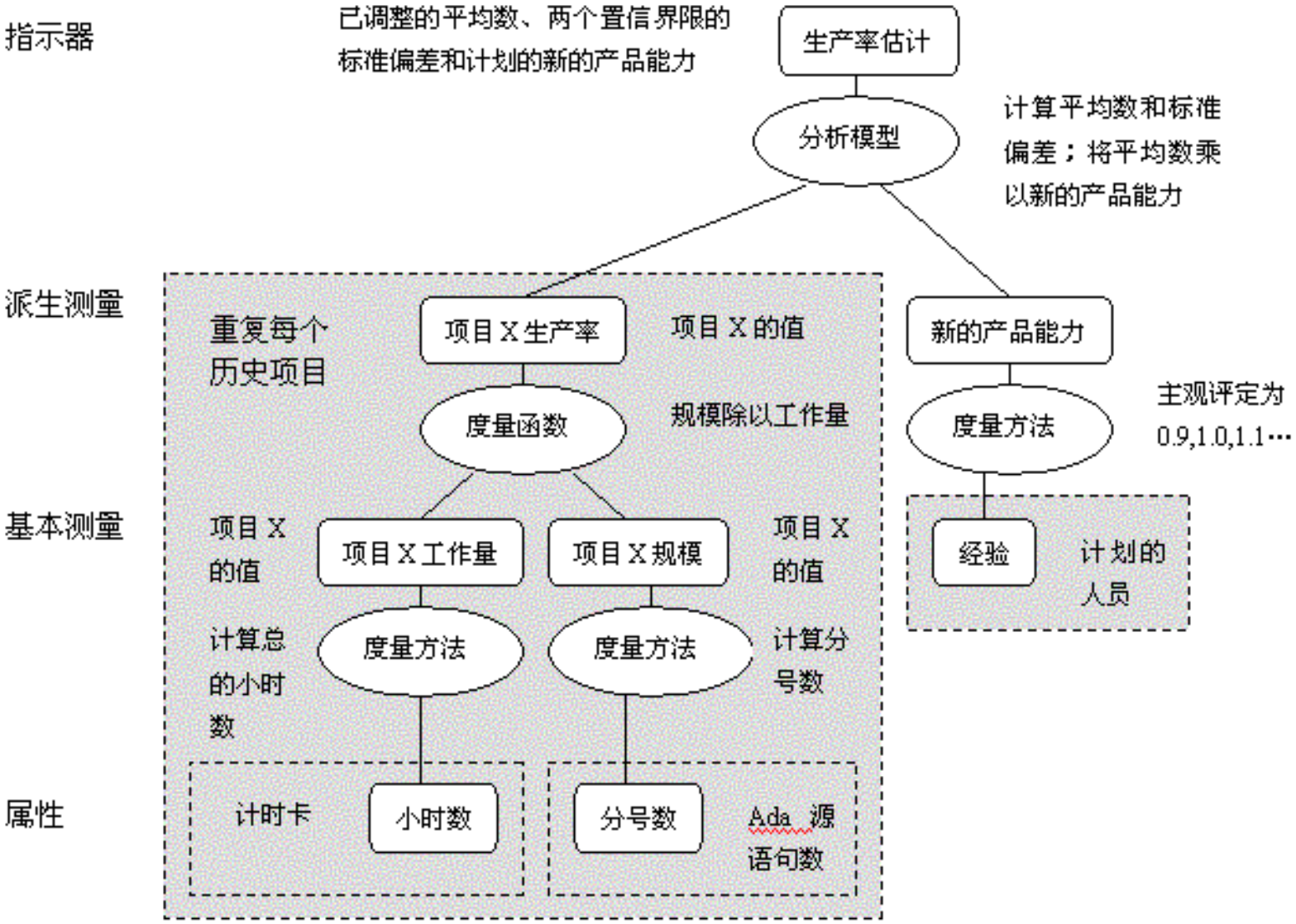


度量构造的细节


信息需要
信息产品
用于确定行动或调查所需要的阀值、目标和界限
决策准则
指示器
为决策制定提供基础的估计或评价
分析模型
组合测量和决策准则的算法
派生测量
派生测量
由两个或多个测量的函数定义的量
度量函数
组合两个或多个基本测量的算法
基本测量
基本测量
由一个指定的方法定义的一个单一属性的测量
度量方法
度量方法
按刻度量化一个属性的操作
实体
属性
属性
与信息需要相关的特性

度量构造的实例-生产率

Measures



基本度量元

Measures

- 一个基本度量元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，值被赋给基本度量元。基本度量元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。
- 每个基本度量元定义了如下特征：
  - 度量方法
  - 方法类型
  - 刻度
  - 刻度类型
  - 度量单位

基本度量元的度量方法

Measures

- 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度量元。
- 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。
- 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。

基本度量元的刻度

Measures

- 刻度是连续的或离散的值的有序集，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。

刻度类型

Measures

- 刻度的类型依赖于刻度上值之间关系的类型：
  - 可测量的数据
    - *比率(定比）：*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（定距）：*数字数据，相同的间距有相同的属性量，不使用0。例如，在一个软件单元中每个附加的逻辑路径有一个附加的圈复杂度值（cyclomatic complexity value）。
  - 不可测量的数据
    - *顺序（定序）：*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（定类）：*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。

标称刻度(nominal scale)或定类刻度

Measures

- 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，若感类趣的属性是宗教，则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)或定序刻度

Measures

- 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，不仅可以将主题分类，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)或定距刻度

Measures

- 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。

比率刻度或定比刻度

Measures

- 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。

刻度类型定义概要

Measures

刻度类型	次序是否有意义	等距离的值是否具有相同的含义	比例的计算是否有意义
标称刻度	否	否	否
顺序刻度	是	否	否
区间刻度	是	是	否
比率刻度	是	是	是

各个刻度类型所认可的操作类型

Measures

刻度类型	标称刻度	顺序刻度	区间刻度	比例刻度
平均数			0	0
中位数		0	0	0
众数	0	0	0	0
极差		0	0	0
方差和标准差			0	0

- 举例：
  - 开发人员的经验度量：
    - 熟练 3
    - 一般 2
    - 新手 1
  - 这是顺序度量，无法求平均。

基本度量元的度量单位

Measures

- 用于导出基本度量元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。

派生度量元

Measures

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

派生度量元的度量函数

Measures

- *度量函数：*用来合并两个或多个基本和/或派生度量元的算法或计算。派生度量元的刻度和单位取决于它所组合的基本度量元的刻度和单位，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度量元的函数，生产的产品总量除以总的工作量就可算出生产率。

指示器的定义

Measures

1. 度量目标
2. 解决的问题
3. 指示器
4. 需要的度量元
5. 决策准则
6. 分析时机
7. 异常的判断方法
8. 分析者
9. 信息需求者
10. 优先级

度量元的定义

Measures

1. 度量元类别
2. 度量元名称
3. 计量单位
4. 采集方法与工具
5. 计算规则
6. 验证方法
7. 采集周期、时间点
8. 采集人
9. 验证人
10. 数据存储位置
11. 对应的指示器
12. 优先级
13. 刻度及刻度类型

实例

度量构造模板

度量元定义练习一

Measures

- 根据上一个练习所确定的度量元，参照度量构造模板，构建其详细的度量构造
- 分组:5人一组
- 时间:30分钟

度量元定义练习二

Measures

- 练习:参考度量构造模板,站在EPG的角度,针对EPG的改进交付代码的质量的这个需求，识别并详细定义其需要的度量元。
- 分组:5人一组
- 时间:30分钟

Measures

如何采集与验证数据？

如何采集数据？

- 前提
  - 数据的含义进行了明确定义
  - 数据采集的方法、工具、步骤是可重复的
- 数据采集的方法
  - 工具自动采集
  - 手动采集
    - 计数
    - 访谈
- 数据采集的基本原则
  - 及时采集
    - 如果不及时采集数据，就无法保证数据的准确性。比如对与日志，如果当天记录工作量，误差为5%，如果第2天记录则误差为10%。
  - 基于统一定义的规范采集数据
  - 注明数据采集日期与采集人员
  - 采集的数据要纳入配置管理

采集后数据的存储

Measures

- 可获得性：
  - 保证度量数据应在一定的权限限制下可访问、可备份和可恢复；
- 可控性：
  - 基本度量和度量分析报告应该得到长期的管理和控制，但派生度量不长期存储；
- 完整性：
  - 通过配置管理来维护数据的完整性，限制读/写访问的权限，并对数据使用版本号；除了直接的度量数据外，还应采集和保存度量的定义和数据的背景描述（比如项目情况说明）；
- 可追溯性：
  - 所有数据都应标识上采集日期和来源，以便帮助度量分析人员进行调查并作比较和分析；
- 安全性：
  - 数据所在的软硬件环境应该是安全可靠的，必须建立定期杀毒和定期备份的机制。

如何验证数据

Measures

- 验证什么？
  - 真实性
  - 有效性
  - 一致性
  - 同步性
- 何时验证数据？
  - 采集时验证数据
  - 分析前验证数据
- 数据验证的方法：
  - 利用数据采集验证检查单
  - 重新采集数据
  - 与其他数据对比分析
    - 案例：ISStudio V6项目4月11日的进展分析报告中：挣值分析表中的工作量统计为149人天，在项目工作量数据跟踪中为：168.5人天，偏差为19.5人天

验证真实性

Measures

- 即根据度量规格说明书中，对照每个度量的定义检查：
  - 是否具有正确的类型；
  - 是否具有正确的格式；
  - 是否在规定值域内；
  - 是否完整；
  - 在数学上是否正确。

验证有效性

Measures

- 验证采集的数据是否真实地反映对应的度量：
- 检查该数据的采集过程是否符合在项目中对应度量规格的定义；
- 数据是否是最新的；
- 计划值和实际值是否发生了变更并作了记录。

验证一致性

Measures

- 即只有证实一个值的定义与同类的其他记录值的定义一致时，才能将它们保存到一起并进行比较分析。在检查数据时，验证人应对过去的数据记录有充分的了解，以及时发现异常数据。异常数据可能反映一个具体的问题，也可能是由于这个项目的过程和工种发生了变化，导致此度量的定义也发生了变化。这时，应把新的定义在这个项目的度量规格中记录下来。同时，这个值不能与同类的其他记录值进行比较，除非能够换算达成一致。要对以下方面验证定义是否与过去一致：
  - 度量刻度（仅用于基本度量）
  - 度量所属阶段（仅用于基本度量）
  - 度量所属工种（仅用于基本度量）
  - 度量单位（适用于基本度量和派生度量）
  - 度量函数（仅用于派生度量）

验证同步性

Measures

- 即计算多个在时间上同步并相互关联的度量得到新的派生度量时，应该验证其时间框架是否匹配。比如生产率的计算公式为产出规模与工作量的比值，需要检验在这个时间段上，这个规模的产出是否是全部在这个工作量的时间段中完成的。如果两者的时间段不同步，则计算得到的生产率是没有意义的。

如何验证数据

1	数据当前性（Data Currency） **·数据与当前进行中的项目活动有关吗？** **·数据是最新的吗？** **·这个项目阶段有关的所有数据都已经采集和处理了吗？**
2	数据聚集结构和属性 **·用作汇集数据记录的字段中的值在记录中是否一致（如缺陷分类、软件子系统或模块标识符、项目活动/任务ID和成本帐户）？** **·将用作汇集数据记录的字段中的值在整个项目组或组织中是否一致？** **·正使用的度量单位在所有项目组或组织中是一样的吗（如小时与天、代码行与KSLOC）？**
3	数据内容 **·数据与计划中的度量规格说明匹配吗？** **·比率/间隔刻度数据——有落在可接受范围之外的值吗？** **·标称/序号刻度数据——所有值是有效的吗（如缺陷分类、严重编号、任务标识符）？** **·任何数据字段的格式是否不正确（如日期、百分比、货币符号、小数位数）？**
4	数据完备性（Data completeness） **·所有一致认可的数据都已经采集了吗？** **·有没有遗漏基本或派生的度量？** **·项目环境数据是否交付？**
5	对现有数据的变更 **·计划值是否意外地进行了变更（如计划的开始和结束日期，计划的成本、人员和工作量，以及计划的已完成的/已测试的组件）** **·对计划值的变更是否意味着一次主要的重新计划？** **·实际值是否意外地进行了变更（如实际的已经完成和活动的开始和结束日期，前一时期实际的成本/人员/工作量）？**
6	数据完整性（Data integrity） **·数据看起来太有规律吗（期望数据有些差异）？** **·有没有明显的例外（一个或多个完全不同于多数其他值的值）？**

Measures

如何分析数据

数据分析的三个层次

Measures

- 基本数据分析
- 过程稳定性
- 相关与回归分析

Measures

基本数据分析

对比分析的基准

Measures

- 横向对比
  - 和本项目的其他度量实体对比
    - 不同模块
    - 不同阶段
    - 不同人员
  - 和本项目的其他度量元对比
    - 规模
    - 工作量
  - 和其他项目对比
  - 和组织的基线对比
  - 和标杆数据对比
- 纵向对比
  - 和本项目的历史对比
    - 历史日期、周次、月份、阶段
  - 和本项目的计划对比

横向对比--与其他项目对比

Measures

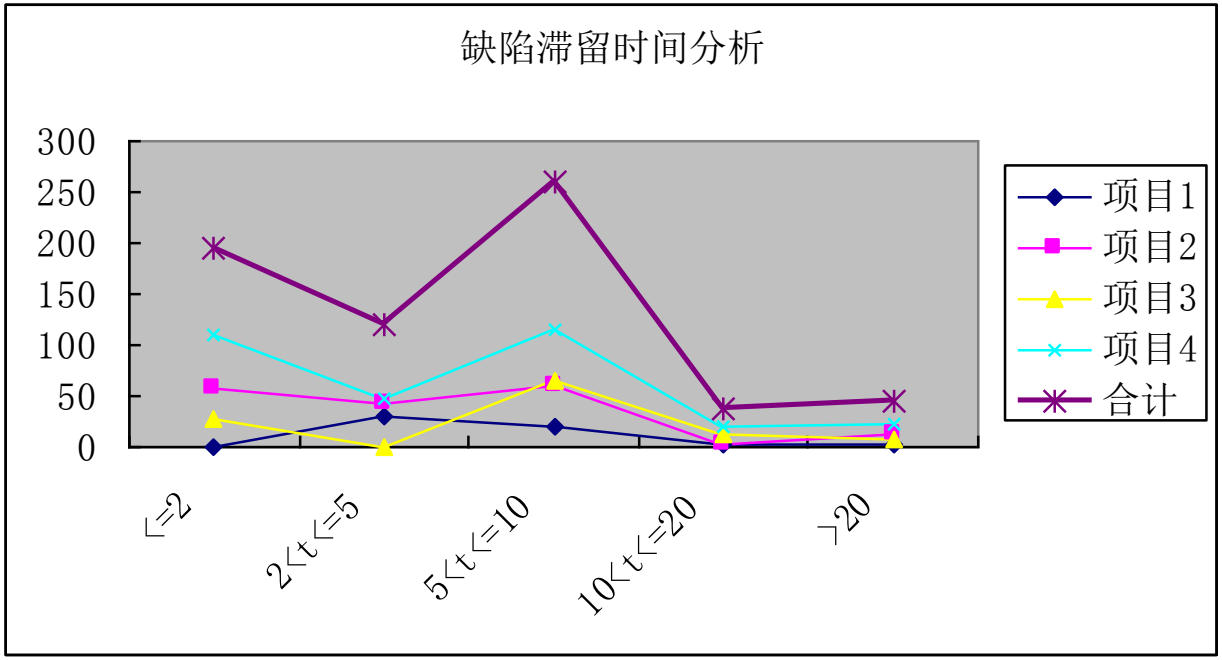
项目名称	项目总规模（代码行）	编码阶段的总工作量（人日）	编码阶段生产率（行/人日）	项目总工作量（人日）	整体生产率（行/人日）
A	3200	11.50	278	332.00	10
B	10700	51.00	210	156.00	69

- 同其他项目对比
  - 为什么2个项目编码阶段的生产率的差别比项目整体生产率的差别大那么多?而且还是逆向的?
- 同标杆数据对比
  - 为什么项目B的整体生产率与业内的标杆数据差别那么大?

横向对比分析—多项目的缺陷驻留时间分析

Measures

缺陷滞留时间(天)	项目1	项目2	项目3	项目4	合计
<=2	1	58.00	**27**	110	196
2<t<=5	29	42.00	**1**	47	119
5<t<=10	19	59.00	**65**	116	259
10<t<=20	3	3.00	**12**	20	38
>20	2	13.00	**7**	22	44



横向对比：评审与测试的效率差异

Measures

项目名称	A	B
规模(行)	3200	10700
同行评审发现的缺陷	139	127
同行评审工作量(人时)	181.7	119.5
同行评审的效率(个/小时)	0.76	1.06
测试发现的缺陷	9	27
测试的工作量(人时)	185	115.5
测试效率(个/小时)	0.05	0.23
效率差异(评审/测试)	15.72	4.55

横向对比--与标杆数据对比

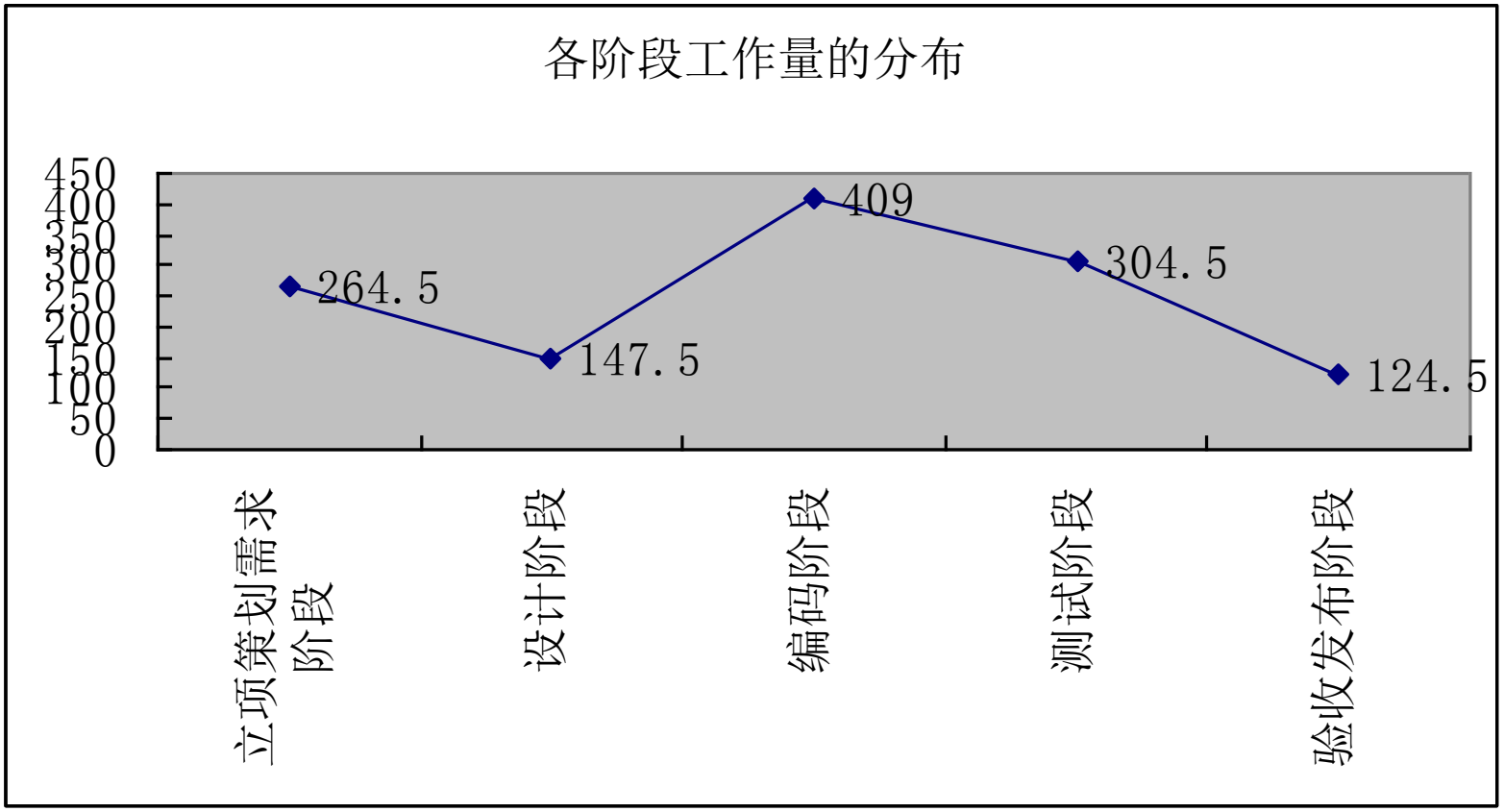
Measures

项目名称	A	B
测试发现的缺陷个数	9	27
所用的测试时间（小时）	185	115.5
测试效率（个/小时）	0.05	0.23
测试用例个数	33	73
测试速率（个/小时）	0.18	0.63
项目总规模（代码行）	3200	10700
测试用例密度(个数/KLOC)	10.3	6.8
缺陷密度(个数/KLOC)	2.8	2.5

- 测试用例密度和业内的标杆数据为什么差别那么大?这种差别是否是合理的?

纵向对比—与历史数据对比

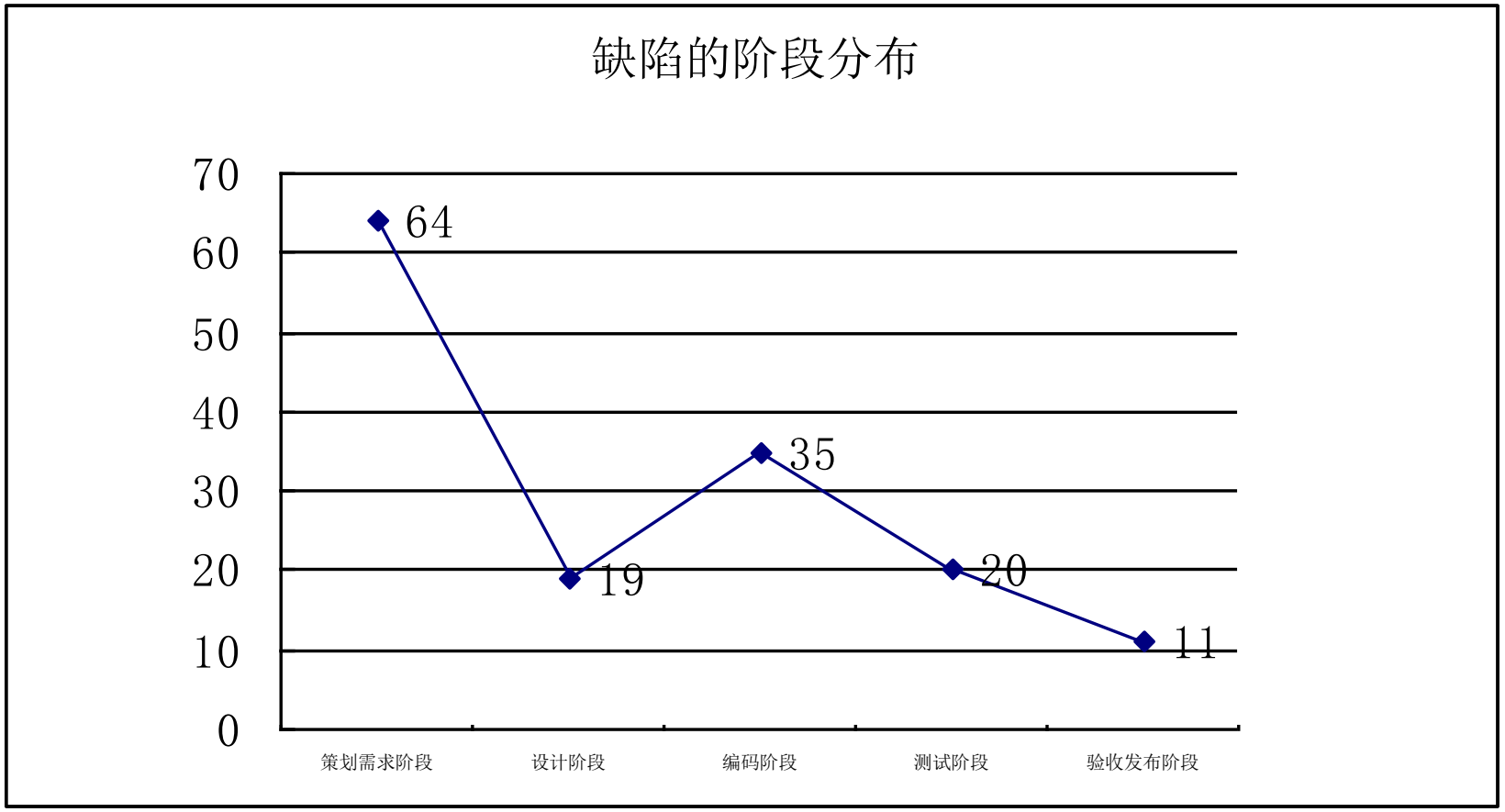
Measures



- 为什么设计阶段的工作量那么少?
- 还需要哪些数据来在证明你的推测?

纵向对比：测试与同行评审发现的缺陷分布

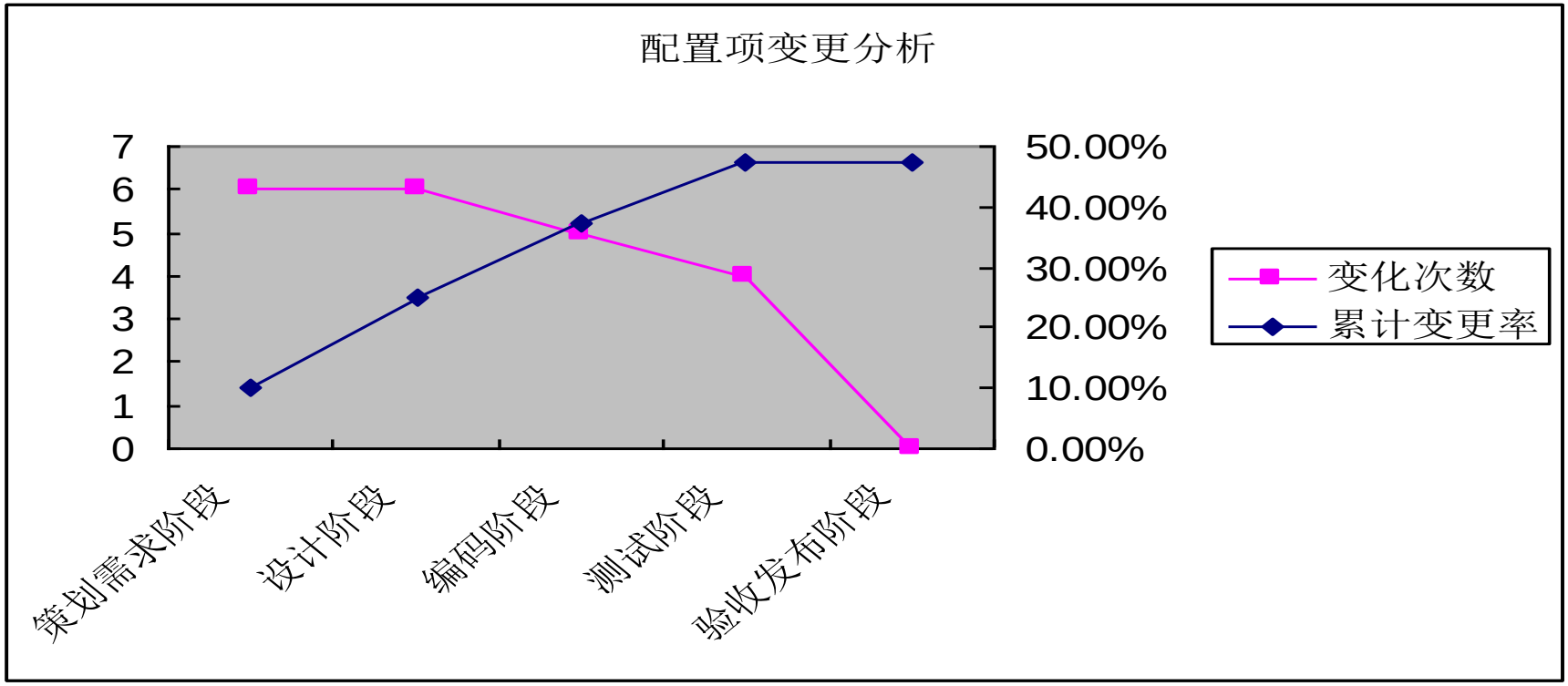
Measures



纵向对比：双轴折线图

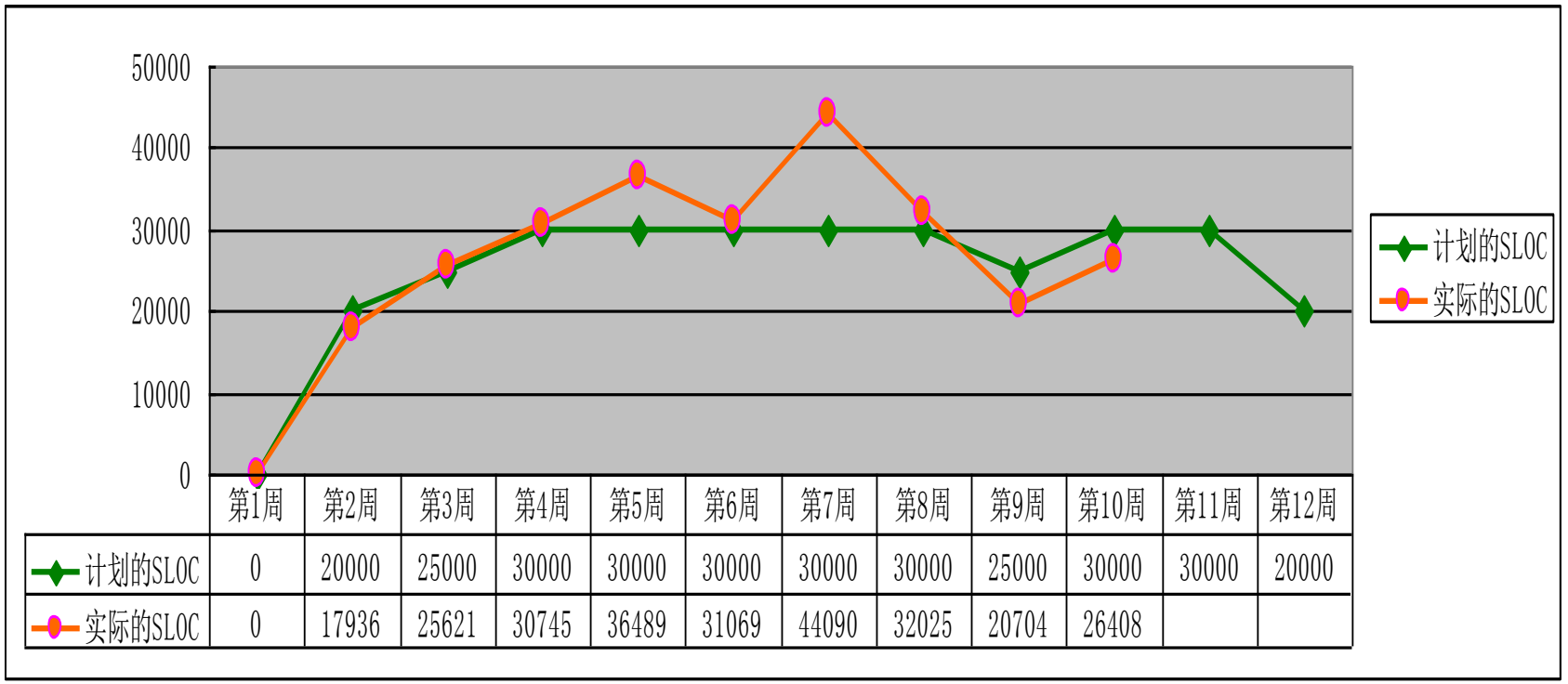
Measures

- 通过该图可以看出：
  - CI的变化次数随项目的进展越来越少
  - 项目的CI的合计变化率为50%



纵向对比：物理规模的计划与实际对比

Measures

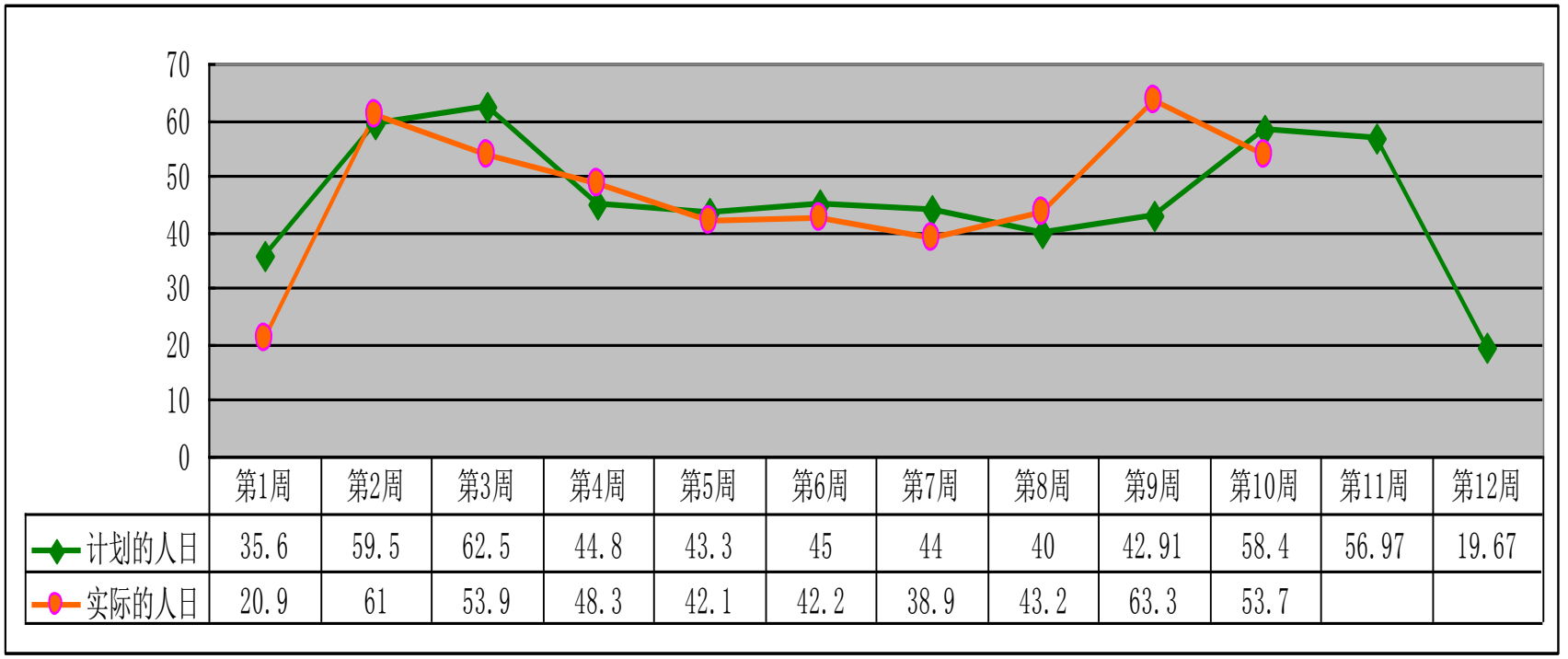


	第1周	第2周	第3周	第4周	第5周	第6周	第7周	第8周	第9周	第10周	第11周	第12周
计划的SLOC	0	20000	25000	30000	30000	30000	30000	30000	25000	30000	30000	20000
实际的SLOC	0	17936	25621	30745	36489	31069	44090	32025	20704	26408		

- **1:(第1周)环境准备阶段，所以没有制作代码。**
- **2:(第5周)本周中自动生成的代码比较多，所以代码行数有大幅增加。**

纵向对比：工作量的计划与实际对比分析

Measures



	第1周	第2周	第3周	第4周	第5周	第6周	第7周	第8周	第9周	第10周	第11周	第12周
计划的人日	35.6	59.5	62.5	44.8	43.3	45	44	40	42.91	58.4	56.97	19.67
实际的人日	20.9	61	53.9	48.3	42.1	42.2	38.9	43.2	63.3	53.7		

- **1：（第1周）主要是环境准备工作，不影响后续开发任务。**
- **2：（第9周）周六、日加班导致实际工作时间增多。**

Measures

如何编写数据分析报告

写分析报告

Measures

- 结论在前，数据在后
- 要便于输出打印
- 信息量要大一点
- 要包括横向分析、纵向分析

写分析报告-局部分析

Measures

- 评审速率与效率的分析
  - 目前只有两个项目结束了，测试的效率及速率数据提交上来了，而且这2个项目分属不同的类型，不具有可比性。
  - 这两个项目的文档评审效率都比较低，一个是0.9，一个是2，相对于低的评审效率，他们的评审速率又不尽相同，名片识别的速率也比较低1.6，就是说每小时看了不到2页文档，但发现的问题是0.9，不到1个，效率不够；多媒体是文档评审的速率太高了，每小时看13页，这个速度比较高，很难保证看文档仔细，自然很难发现问题，效率就低了。代码评审也是这个问题。以后其他的项目要注意看文档仔细，评审时可对文档分部分给不同的人，保证大家都仔细看，提高效率。

写分析报告-总体分析

Measures

- 从以上八种度量指标结果来看，该项目运作大致平稳。
- 1. 项目经理针对9，10月份人力超出的部分，采取了及时的矫正措施，项目成员也相应提高自己的工作效率，控制投入，尽量在工作时间完成任务，使得本月人力较少了19.57%。人力投入减少，成本也会相应降低，因此本月成本节省了18.45%。
- 2. 进度方面，通过项目经理修改项目时程等措施，目前项目总体进度还落后3%，其中测试阶段的进度落后的最多，主要是操作手册、维护手册的编写与评审还没有完成。在建制与移转阶段还落后5%，主要是整合与验证测试、系统正式上线及上线辅导的任务还没有完全解决。总体来说，各阶段的主要任务紧本已经完成，通过项目成员的相互合作及项目经理采取有效矫正措施，前几个月落后的进度已渐渐赶上。造成需求分析阶段、设计阶段、程序制作阶段工作任务延期严重的原因是这几个阶段的里程碑评审没有完成，由于项目成员对相关任务的知识技能掌握不够，也是影响这些阶段工作任务未能结案的原因。

写分析报告-总体分析

Measures

- 3.项目度量方面，《度量与分析计划书》中计划了8个度量项目，其中项目规模的度量时机是在里程碑审查前，由于项目在本月进行了一次里程碑审查，因此本月对项目规模进行了度量，从项目度量分析表中可以看到，提案阶段的度量完成率是86%，原因是PPQA从需求分析阶段才投入项目，导致项目的质量状况在提案阶段没有度量数据。其它阶段都按照计划完成度量 。
- 4.需求变更和列管文件变更方面，本月针对一项需求发生了变更，在建制与移转阶段共发生了10次变更，需求稳定度为93.46%，项目总体共发生了14次变更，总体需求稳定度为90.85% ，与上月相比本月对于需求变更的次数减少很多，说明虽然需求分析阶段的工作不够细致和深入，但随着项目的继续进行，项目成员在处理项目流程时也逐渐成熟，对于分析问题和解决问题的能力也有所提高。在列管文件变更分析表中可以看到到本月为止共对19项文件与基准进行了变更，其中变更次数最多的是项目计划书和需求规格书然后是设计说明书，对于项目来说，最重要的就是项目计划书和需求规格书，随着项目的进行 ，项目经理要根据项目的实际运作情况不断的修订项目计划书，需求也不是一尘不变的，系统分析人员也会根据客户提出各种的需求而不断地修改需求规格书.

写分析报告-总体分析

Measures

- 5.质量方面，RM在本月的总体缺陷率降低为0，是需求规格书和需求追朔表在上月出现的缺陷已经得到解决。另外， MA的总体缺陷率也降低为0 ，主要是本月针对度量与分析报告进行了检查， 零缺陷。 CM、PMC在本月的总体质量状况也出现好转，其中架构管理作业的缺陷率由上月的11.54%低到3.85% ；项目监控记录的缺陷率也由上月的17.65%降低到5.88% 。通过QA的问题发现形式，可以综合全面的看出项目在运作时产生的缺陷，从质量分析表中看出，在建制语移转阶段发现的缺陷最多，在所有发现的缺陷中，属于需求分析阶段的缺陷最多，这同需求变更分析表反映的问题相同，由于需求分析做得不够细致，导致在移转阶段时出现很多变更。在列管文件变更分析表中也可看到《需求规格书》变更的次数也最多。

Measures

性能基线的建立方法

如何建立性能基线?

性能基线

Measures

- 定义：
  - 基线是基于历史数据建立的，刻画了选中的过程的性能
- 理解与实施要点
  - 基线可以针对一个子过程或过程元素、一个生命周期的阶段过程、一组相关的过程而建立
  - 针对不同的部门、产品线、业务领域等可以建立不同的基线
  - 如果对组织标准过程有几类裁剪，则可能需要针对不同类型的裁剪建立不同的基线
  - 基于历史的度量数据建立基线，如果样本点比较多（k>35)则基线的可信度比较高
  - 如果样本点不够多(<35)，则可以建立尝试的基线(k>20)，SEI认可的建立尝试控制限的样本数量>=6
  - 建立基线时要注意数据分组/分层的现象

性能基线的描述方式

Measures

- 构成基线的要素：
  - 分布的类型、上限、下限、中值

ID	基线名称	类型	单位	下限	中心线	上限	分布类型
1	需求评审效率	all	个/人时	0	0.763	1.928	非正态
2	需求评审缺陷密度	all	个/页	0	0.713	1.807	正态
3	设计评审效率	all	个/人时	0	0.892	2.6	正态
4	设计评审缺陷密度	all	个/页	0	0.2451	0.5818	正态
5	测试用例评审效率	all	个/人时	0	1.416	3.169	正态
6	测试用例评审缺陷密度	all	个/页	0	1.028	1.412	正态
7	用户手册评审效率	all	个/人时	0	1.006	1.887	正态
8	用户手册评审缺陷密度	all	个/页	0	0.2147	0.4282	正态
17	系统总体测试的效率	嵌入式	个/人时	0	0.9172	0.4823	正态
18	系统总体测试的缺陷密度	嵌入式	个/KLOC	0	0.952	2.830	正态
19	概念阶段进度偏差率	all	%	-5%	2%	9%	正态
20	计划阶段进度偏差率	all	%	-7%	3%	12%	正态
21	开发验证阶段进度偏差率	all	%	-10%	3%	17%	正态
22	转移验证阶段进度偏差率	all	%	-9%	5%	19%	正态
23	整体进度偏差率	all	%	-9%	5%	19%	正态
24	周工作量偏差率	all	%	-9.40%	0.09%	9.59%	正态

性能基线案例

Measures

PPB 元素	*上限*	*均值*	*下限*	计量单位
需求定义	50.8	35	31.6	Hours/Complex Requirement
	29.2	21	15.8	Hours/Nominal Requirement
	13.4	8.6	5.2	Hours/Simple Requirement
设计	81.4	49.8	43.6	Hours/Complex Requirement
	44.4	31.7	21.2	Hours/Nominal Requirement
	19.6	13.3	5.5	Hours/Simple Requirement
实现	13.4	8.6	5.4	Hours/Interface
	35.4	17.7	10.3	Hours/Design Page
	6.54	4.31	3.21	Hours/Object
集成	301.5	153.5	23.5	Hours/Subsystem
	32.5	16.8	7.8	Hours/Component
系统测试	19.52	12.4	8.3	Hours/Test Scenario

建立基线的流程

- 1 确定建立基线的过程及其属性
- 2 收集历史数据
- 3 验证数据的准确性、完整性，初步观察数据的特点
- 3 对数据分类
  - 项目类型
  - 规模
  - 生命周期
  - 技术平台
  - 人员背景
  - ……
- 4 进行方差分析，判断不同类的项目是否需要建立不同的基线
- 5 选择基线建立的方法
- 6 采用XMR图
  - 对项目的数据按时间排序
  - 画出控制图，剔除异常点
  - 计算平均值
  - 计算控制线
  - 判断离散系数的大小
  - 确定性能基线
- 7 采用箱线图
  - 画出箱线图， 剔除异常点
  - 计算中值
  - 计算控制线
  - 确定性能基线
- 8 描述并发布性能基线

应该建立哪些性能基线？

Measures

- 性能基线是为管理而服务的，性能基线的需求来源于：
  - 客户
  - 高层经理
  - 项目经理
  - EPG
  - QA、CM、测试、开发人员等
- 常见的性能基线包括：
  - 质量：缺陷密度、测试投入、评审投入
  - 进度：进度偏差率
  - 成本：工作量偏差率、工作量分布
  - 需求：规模变更率
  - 过程性能：估计偏差率、评审效率、测试效率、开发效率
  - ……

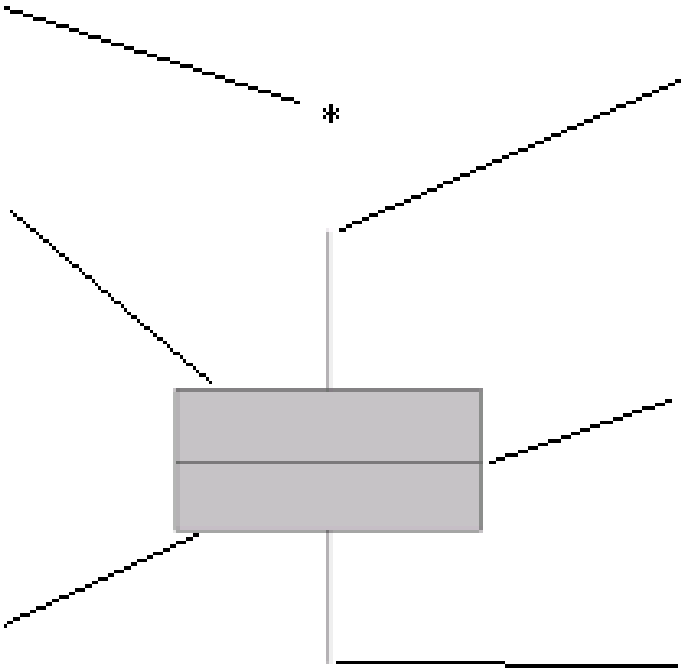
基线的建立方法-箱线图

Measures

异常值 – 异常大或异常小的观测值。超过须线的值即为异常值。

默认情况下，方框顶部为上四分位数 (Q3) – 75% 的数据值小于等于此值。

默认情况下，方框底部为下四分位数 (Q1) – 25% 的数据值小于等于此值。



默认情况下，上须线延伸至上限范围内的最高数据值。

上限 = Q3 + 1.5 (Q3 – Q1)

中位数 – 数据的中间位置值。一半观测值小于等于此值。

默认情况下，下须线延伸至下限范围内的最低值。

下限 = Q1– 1.5 (Q3 – Q1)

基线的建立方法-箱线图

Measures

- 箱线图有3个值决定
  - 中位数 Q2
  - 第一分位数 Q1
  - 第三分位数 Q3
- 盒子的形状由Q2,Q1,Q3的位置决定
- 内距 (IQR) =Q3-Q1
- 内围:
  - Q1-1.5IQR
  - Q3+1.5IQR
- 外围:
  - Q1-3IQR
  - Q3+3IQR

Q1与Q3的计算方法

Measures

- EXCEL中
  - k= trunk ((quart/4)*(n-1)))+1 ,quart为0到4之间的一个整数，即第quart四分位数。N为这组数中数值的个数。
  - F=(quart/4)*(n-1)- trunk ((quart/4)*(n-1)。
  - 在数组中找到第k，k+1个整数，按下列公式计算：
  - Output = a[k]+(f*(a[k+1]-a[k]))
  - a[k] = 第k小\的数值；
  - a[k+1] = 第k+1小\的数值；
- MINITAB中
  - 计算下四分位数：
    - k= trunk (n/4),n为这组数中数值的个数；
    - f=n/4- trunk (n/4)
    - Output = a[k]+(f+0.25)*(a[k+1]-a[k]))
    - a[k] = 第k小\的数值；
    - a[k+1] = 第k+1小\的数值；
  - 计算上四分位数：
    - k=ceiling(3*n/4)+1,n为这组数中数值的个数。
    - f=3*n/4-int(3*n/4)；
    - 在数组中找到第k，k+1个整数，按下列公式计算：
    - Output = a[k]-(f+0.25)*(a[k]-a[k-1]))

算法不同，结果有所不同，都是一种近似！

Measures

案例一：缺陷密度基线

案例一：采用箱线图建立基线

Measures

编号	缺陷密度(个/KLOC)
1	1.37
2	1.57
3	0.7
4	0.47
5	0.89
6	0.67
7	0.21
8	0.67
9	0.89
10	0.25
11	0.63
12	0.6
13	0.13
14	0.47
15	2.38
16	0.33
17	1.11
18	0

- 在EXCEL中计算结果如下：
  - Q1=0.37
  - Q2=0.65
  - Q3=0.89
  - IQR=0.52
- 于是建立基线如下：
  - 下限：0 （负数无意义，故取值为0）
  - 中值：0.65
  - 上限：1.67

Measures

案例二：工作量分布基线

性能基线建立的案例-原始数据

Measures

序号	需求	设计	开发	测试	交付	管理活动	总工作量
1	155	163	308	118	256	175	1175
2	107.5	126.2	345.19	101.69	9	175	864.58
3	8.25	14	182.7	123.7	10.7	117.15	456.5
4	153	208	366	266	7	330	1330
5	4.125	48.9375	43.0625	122.125	287.875	20.9375	527.0625
6	1.31	2.44	4.5	7.25	10.63	17.25	43.38
7	75	45	191	81	48	281	721
8	460.24	376.56	857.72	146.44	209.2	209.2	2259.36
9	200	218.75	412.5	300	99.75	287.5	1518.5
10	312	80	118	92	263	26	891
11	1.31	2.44	4.5	7.25	10.63	17.25	43.38
12	41	26	12	29	5	16	129
13	20	30	150	30	245	70	545
14	102.5	236.63	526.23	166.44	5.38	100.38	1137.56
15	38.13	158.87	265.19	242.13	143.01	38.2	885.53
16	184.13	198.6	510.88	360.06	73	199.83	1526.5
17	61.25	154.38	314.38	328	44.75	64.5	967.26
18	25.25	168	255.06	120.31	74.58	125.88	769.08
19	7	380	281	222	0	0	890
20	196	4	619	310	20	51	1200
21	51	5	523.5	228.5	0	55.5	863.5
22	168.5	196.2	294.2	238	0	204	1100.9

对原始数据变换为比例

Measures

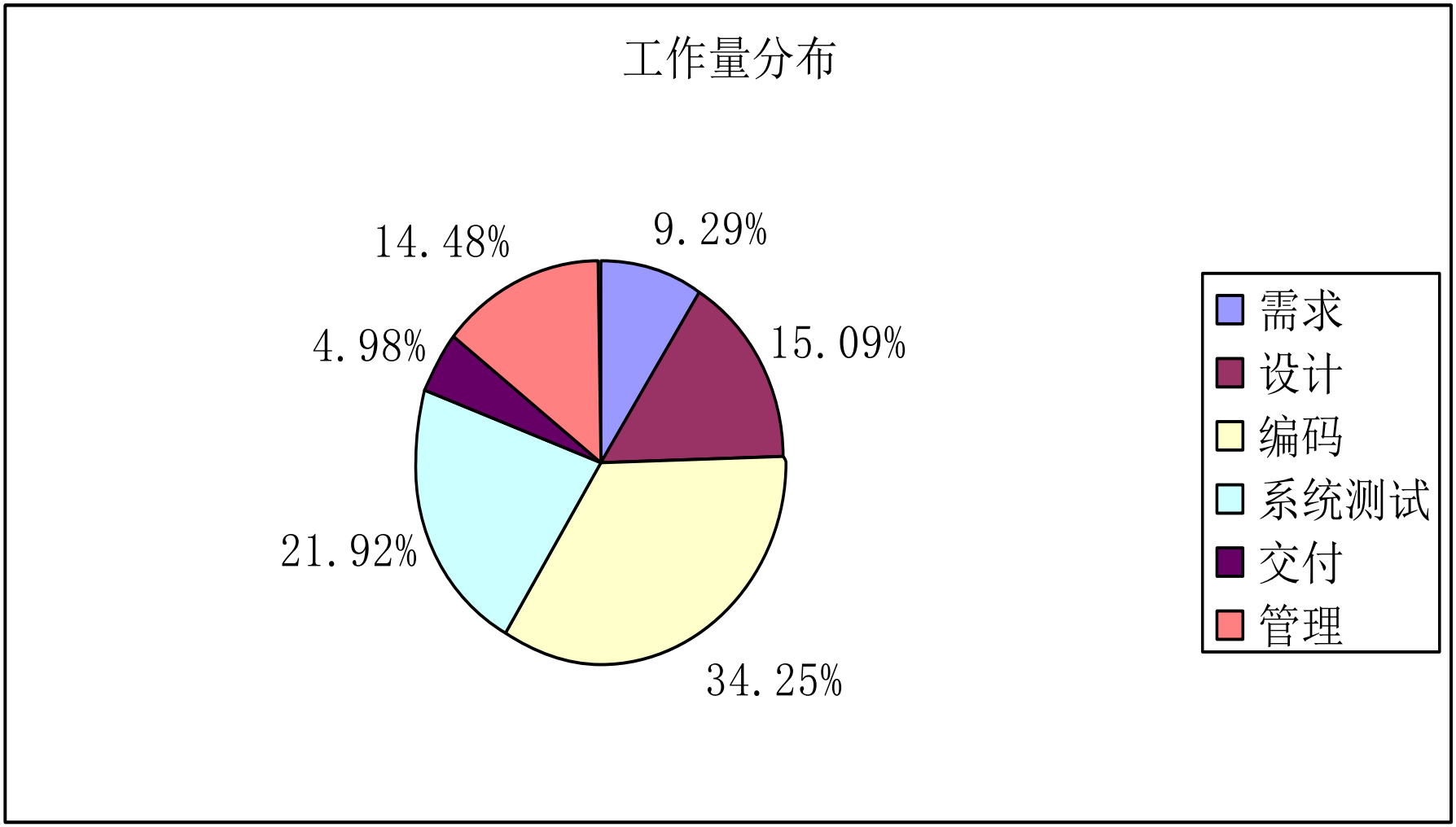
序号	需求%	设计 %	开发 %	测试 %	交付 %	管理活动%
1	13.19%	13.87%	26.21%	10.04%	21.79%	14.89%
2	12.43%	14.60%	39.93%	11.76%	1.04%	20.24%
3	1.81%	3.07%	40.02%	27.10%	2.34%	25.66%
4	11.50%	15.64%	27.52%	20.00%	0.53%	24.81%
5	0.78%	9.28%	8.17%	23.17%	54.62%	3.97%
6	3.02%	5.62%	10.37%	16.71%	24.50%	39.76%
7	10.40%	6.24%	26.49%	11.23%	6.66%	38.97%
8	20.37%	16.67%	37.96%	6.48%	9.26%	9.26%
9	13.17%	14.41%	27.16%	19.76%	6.57%	18.93%
10	35.02%	8.98%	13.24%	10.33%	29.52%	2.92%
11	3.02%	5.62%	10.37%	16.71%	24.50%	39.76%
12	31.78%	20.16%	9.30%	22.48%	3.88%	12.40%
13	3.67%	5.50%	27.52%	5.50%	44.95%	12.84%
14	9.01%	20.80%	46.26%	14.63%	0.47%	8.82%
15	4.31%	17.94%	29.95%	27.34%	16.15%	4.31%
16	12.06%	13.01%	33.47%	23.59%	4.78%	13.09%
17	6.33%	15.96%	32.50%	33.91%	4.63%	6.67%
18	3.28%	21.84%	33.16%	15.64%	9.70%	16.37%
19	0.79%	42.70%	31.57%	24.94%	0.00%	0.00%
20	16.33%	0.33%	51.58%	25.83%	1.67%	4.25%
21	5.91%	0.58%	60.63%	26.46%	0.00%	6.43%
22	15.31%	17.82%	26.72%	21.62%	0.00%	18.53%

性能基线建立的案例-控制图法

Measures

	需求	设计	编码	系统测试	交付	管理
3sigma上限	31.55%	29.29%	65.63%	39.18%	12.76%	32.66%
2sigma上限	23.81%	23.86%	53.60%	32.41%	9.94%	25.93%
1sigma上限	16.07%	18.43%	41.58%	25.64%	7.12%	19.20%
平均值	8.00%	13.00%	29.55%	18.88%	4.29%	12.47%
1sigma下限	0.60%	7.57%	17.53%	12.11%	1.47%	5.74%
2sigma下限	0.00%	2.15%	5.50%	5.34%	0.00%	0.00%
3sigam下限	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
sigma	7.85%	5.43%	12.03%	6.77%	2.82%	6.73%
离散系数	98.13%	41.77%	40.70%	35.84%	65.81%	53.97%
对平均值进行规格化	9.28%	15.08%	34.28%	21.91%	4.98%	14.47%

性能基线建立案例-工作量分布



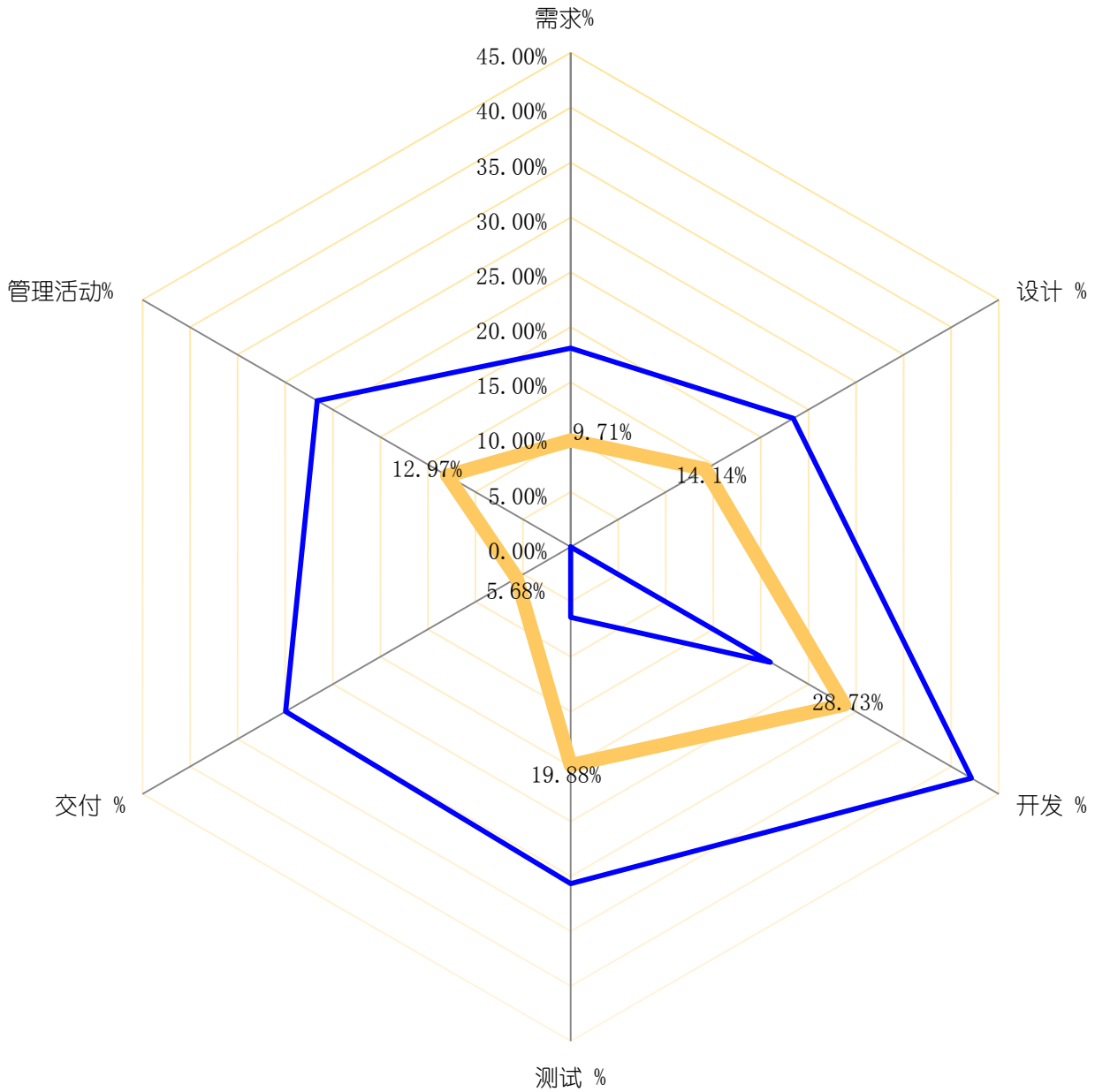
根据箱线图法建立的基线

Measures

	需求%	设计 %	开发 %	测试 %	交付 %	管理活动%
Q1	3.38%	5.78%	26.28%	12.48%	1.20%	6.49%
Q2	9.71%	14.14%	28.73%	19.88%	5.68%	12.97%
Q3	13.19%	17.53%	36.84%	24.60%	20.38%	19.91%
IQR	9.81%	11.75%	10.56%	12.13%	19.18%	13.43%
上限	18.09%	23.41%	42.12%	30.67%	29.97%	26.63%
下限	0.00%	0.00%	21.00%	6.42%	0.00%	0.00%

性能基线建立案例-工作量分布

Measures



基线建立练习

Measures

- 有11个项目的数据测试用例密度数据，计量单位 用例个数/KLOC
- 请建立测试用例密度的性能基线，并按照“基线的描述”这一小节的格式描述建立的基线

项目名称	测试用例密度
青**项目	1.78
C**目	1.37
青**统	1.18
计**目	6.72
招**目	5.18
蛇**统	3.01
出**统	1.32
空**目	0.82
港**目	0.87
国**目	1.16
数**品	0.38

Measures

度量与分析技术应用实例

如何跟踪项目进展？
如何分析与使用工作量度量数据？
如何分析生产率？
如何使用度量数据管理测试过程？
如何进行测试预测？
如何设定量化管理目标？

Measures

如何跟踪项目进展？

EV值跟踪

计划与实际的偏离跟踪

工期的跟踪

Measures

- 偏离的确定
  - 任务的工期偏离
  - 项目的工期偏离
    - 关键路径的识别
- 绝对偏离与相对偏离
  - 绝对偏离=计划完成日期-实际完成日期-节假日
  - 相对偏离=（计划完成日期-实际完成日期-节假日）/计划工期
- 比较基准的选择
  - 最初版本
  - 最新版本

挣值管理

Measures

- 是一种综合了范围、进度计划、资源和项目绩效测量的方法，它对计划完成的工作、实际挣得的收益、实际花费的成本进行比较，以确定成本与进度完成量是否按计划进行。

3个关键基本变量

Measures

- 计划价值（PV，planned value）
  - 又叫计划工作预算费用（budgeted cost of work scheduled,BCWS），描述了在项目进度表中任何一个给定点上，应该要完成多少工作。它把计划要完成的工作通过量化的方式表示出来，并成为实际项目进度测量的基准（也被称为绩效测量基准，或叫PMB）。
- 实际费用（AC，actual cost）
  - 又被称为已完成工作实际费用（actual cost of work performed,ACWP),它反应了到目前为止（或在某个指定的时间段内）在已完成工作上所消耗的资源量。
- 挣值（EV，earned value）
  - 是指一项活动或一组活动的已经完成工作的计划价值。通俗来说，就是所做工作值多少钱。
  - 当某项任务完成时，其计划价值就转变成了挣值。

3个关键基本量的计算案例

Measures

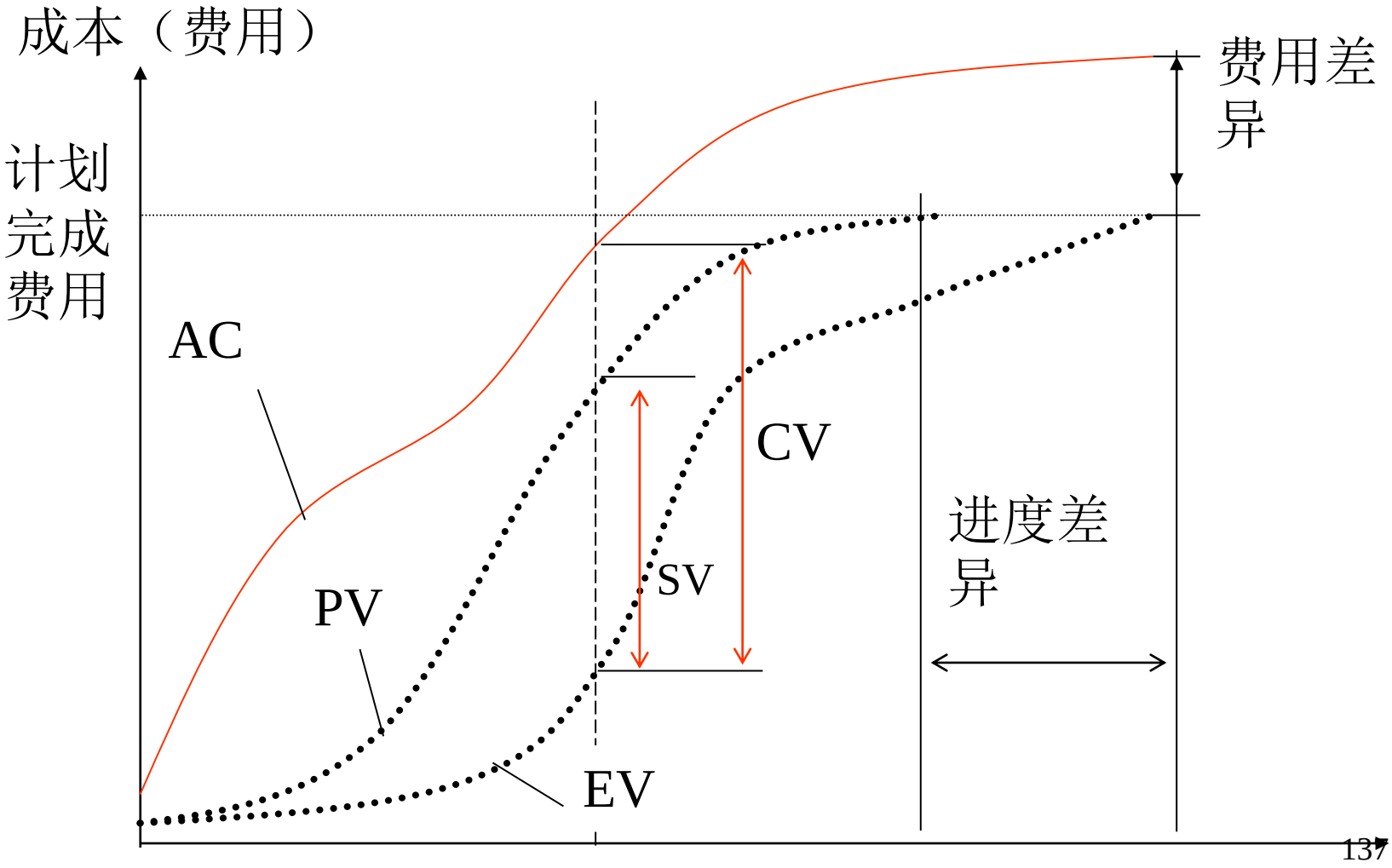
任务	计划工作量	实际工作量	任务完成否	挣值
（3月17日——3月21日）个人保后监管代收款出账审批流程编码以及单元测试完成	5	6	1	5
（3月17日——3月21日）个人保后监管要件押金出账审批流程编码以及单元测试完成	5	5	1	5
（3月17日——3月21日）反担保物借用流程编码及单元测试完成	5	4	1	5
（3月17日——3月21日）项目解保流程编码及单元测试完成	5	4	0	0
（3月17日——3月21日）档案管理编码和单元测试	5	4	0	0
小计	25	23		135 15

2个差异分析变量

Measures

- 项目进度差异（SV，schedule variance）
  - -等于挣值与预算成本的差异，SV=EV-PV
  - -变化：进度差异率 SV%=(EV-PV)/PV
  - -例如：EV=15,PV=25,SV=-10,说明进度滞后
- 项目成本差异（CV，cost variance）
  - -等于挣值与实际成本的差异，CV=EV-AC
  - -变化：成本差异率 CV%=(EV-AC)/EV
  - -例如：EV=15,AC=23,CV=15-23=-8,说明成本超支

挣值评价曲线图



基于时间的进度偏差和进度绩效指标

Measures

- SVt=计划完成时间-实际完成时间=PT-AT
- SPIt=PT/AT
- 举例：
  - 某项目计划工期为12个月，实际工期为18个月，计划工作量150人月，实际工作量为200人月，则当项目完工时：
    - 基于工作的绩效度量指标
      - SV=EV-PV=0
      - SPI=EV/PV=1
      - AC=200
      - PV=150
      - CPI=EV/AC=0.75
    - 基于时间的绩效度量指标
      - PT=12
      - AT=18
      - SVt=12-18=-6
      - SPIt=12/18=0.66

2个指标变量

Measures

- ①成本绩效指标（CPI，cost performance index）：等于挣值与实际成本之比，它衡量的是正在进行的项目的成本效率。计算公式为：

CPI=EV／AC

-CPI<1.0　　费用超支

-CPI>1.0　　费用节余

- ②计划完工指标（SPI，schedule performance index）：等于挣值与预算成本之比，它衡量的是正在进行的项目的完工程度。计算公式为：

SPI=EV／PV

-SPI>1.0　　表示进度超前

-SPI<1.0　　表示进度落后

2个预测变量

Measures

- 完工尚需估算（ETC，estimate to complete）
  - 指完成项目预计还需要的成本。预测ETC的大多数技术是根据迄今为止的实际绩效对原始估算进行一些调整。
- 完工估算（EAC，estimate at completion）
  - 是指规定的工作范围完成时项目的预计总成本。预测EAC通常使用的技术是按照项目迄今的实际绩效调整原始的成本估算，通常表示为：EAC=迄今实际成本+ETC
- 实证研究说明：当项目进行到15%-20%时，累积的成本绩效指标就已经基本稳定下来，累积CPI的变动幅度就不超过10%。因此可以用来预测整个项目完成时的总成本。

常用的EAC计算方法

Measures

假设	预测公式
未来的费用绩效将会与过去的费用绩效保持一致	EAC=AC+[(BAC-EV)/CPI]=BAC/CPI BAC=原始的项目估算
未来的费用绩效将会与最近3个测试周期的总费用绩效保持一致	EAC=AC+[(BAC-EV)/((EVi+EVj+EVk)/(ACi+ACj+ACk))]
未来的费用绩效将会收到过去进度绩效的影响	EAC=AC+[(BAC-EV)/(CPI*SPI)]
未来的费用绩效将会受到过去进度绩效和费用绩效2个指标的影响，且这2个指标对未来绩效的影响程度成一定的比例	EAC=AC+[(BAC-EV)/(0.8CPI+0.2SPI)]

计算实例

Measures

指标	值
原始估算（BAC）	40
目前挣值（EV）	15
目前实际投入（AC）	23
计划投入（PV）	25
进度偏差（SV）	-10
成本偏差（CV）	-8
进度业绩指标（SPI）	0.60
成本业绩指标（CPI）	0.65
预计完工估算（EAC）	61.33
预计完工尚需估算（ETC）	38.33

挣值管理基本绩效数据解释

Measures

绩效测量数据		进度		
		SV>0 & SPI>1.0	SV=0 & SPI=1.0	SV<0 & SPI<1.0
费用	CV>0 & CPI>1.0	进度提前 费用结余	符合进度计划 费用结余	进度滞后 费用结余
	CV=0 & CPI=1.0	进度提前 符合预算计划	符合进度计划 符合预算计划	进度滞后 符合预算计划
	CV<0 & CPI<1.0	进度提前 费用超支	符合进度计划 费用超支	进度滞后 费用超支

挣值管理和基本的项目管理问题

Measures

项目管理问题	挣值管理绩效测量数据
我们在时间方面的情况如何？	进度分析和预测
我们的进度是提前了还是滞后？	进度偏差(SV)
我们的时间利用率如何？	进度绩效指数（SPI）
我们将会在什么时候完成工作？	完工时间估算（EACt）
我们在费用方面的情况如何？	费用分析和预测
我们费用是超支了还是尚有结余？	费用偏差(CV)
我们的资源利用率如何？	费用绩效指数（CPI）
我们该以怎样的利用率使用剩余的资源？	完工尚需绩效指数（TCPI）
完成全部的项目工作需要消耗多少资源？	完成时估算（EAC）
我们的总费用将会超支还是会有结余	完工偏差（VAC）
剩余的工作预计还需要多少费用？	完成尚需估算（ETC）

案例分析:ZC-ZKZ项目

Measures

- 项目代号：jr0240
- 项目名称：中科智项目
- 项目经理：许慧
- 项目的计划工期：　　月
- 项目开始日期:
- 项目计划结束日期:
- 项目的计划总工作量:
- 项目的阶段划分：里程碑3
- 项目的当前阶段:
- 本阶段的计划结束日期:
- 当前日期：2008年4月11日

现状

Measures

指标	值
原始估算（BAC）	135
目前挣值（EV）	65
目前实际投入（AC）	159.5
计划投入（PV）	135
进度偏差（SV）	-70
成本偏差（CV）	-94.5
进度业绩指标（SPI）	0.48
成本业绩指标（CPI）	0.41
预计完工估算（EAC）	331.27
预计完工尚需估算（ETC）	171.77

进度分析和与预测

Measures

- 进度偏差：我们的进度是提前了还是滞后了？
  - -SV=-70　　进度滞后
  - -SV%=SV/PV=-51.85%　　进度落后计划的51.85%
  - -SVt=PT-AT=？
- 进度绩效指标：我们的时间利用率如何？
  - -SPI=EV/PV=0.48　时间利用率比较低，项目工作的执行效率只有48%
- 完成时间估算：我们将会在什么时间完成工作？
  - -EACt= (BAC/SPI)/(BAC/原计划完工的月数) = (135/0.48) /(135/原计划完工的月数) =？
  - -关键路径分析

成本分析和预测

Measures

- 成本偏差：我们费用是超支了还是尚有结余？
  - -CV=EV-AC=-94.5　费用超支
  - -CV%=CV/EV=-145.38%　费用超支145.38%
- 成本绩效指标：我们的资源利用率如何？
  - -CPI=EV/AC=0.41　费用超支，投入1元钱只完成了价值0.41元的任务
- 完工尚需绩效指标：我们该以怎样的利用率使用剩余的资源？
  - -TCPI=(BAC-EV)/(BAC-AC)=-2.86　已经无法在预算内完工了
- 完成时估算：完成全部的项目工作需要消耗多少费用？
  - -EAC=BAC/CPI=331.27
- 完工偏差：我们的总成本将会超支还是会有结余？
  - -VAC=BAC-EAC=-196.27　成本超支
  - -VAC%=VAC/BAC=-145.38% 成本超支145.38%
- 完成尚需估算：剩余的工作预计还需要多少费用？
  - -ETC=(BAC-EV)/CPI=171.77

挣值管理实证研究中的核心结论

Measures

- 假设
  - 项目的完成量已经超过15%
    - 项目完工时总的超支额将不少于目前水平
    - 项目完工时总的超支比重将高于目前的超支比重
- 结论
  - 一旦发生成本超支，就不可能恢复正常
- 依据
  - 自1977年开始，美国国防部700多个大型项目的数据
- 分析：
  - 相比近期的工作，远期的工作更为粗糙，因此如果连近期的计划都无法实现，远期计划就更不可能实现了

挣值管理的不足

Measures

- 挣值管理的一个基本假设是可以比较准确地预计未来的情况（未来的情况变化很少），而且实际进度可以准确地度量，从而可以建立项目基准计划。但是对于某些不确定性程度较高的项目当中与现实相差很远，如技术不确定性较高的研究和开发项目，这一假设的基础非常不牢固。
- 挣值管理本身作为一个绩效度量技术，能够度量当前的绩效、预测未来的发展趋势，但进度和成本的困难可能是一些重要问题的表现，而不是问题的根源，挣值管理并没有产生解决问题的方法。而其他的项目管理工具，如网络计划技术则可以识别关键路径，并产生一定的管理建议。
- 大量的创新性、具有高度复杂性的项目出现，挣值管理在这些类型的项目中应用具有一定的困难。对于具有决策灵活性的阶段性项目而言，挣值管理的使用可能是无效的。大型的开发项目可能会有阶段性，存在决策点，在决策点根据前一阶段的成本和收益状况，决定下一阶段活动是否继续。这一类型的项目通常无法进行项目的整体计划，因此也难以使用挣值管理。

Measures

如何分析工作量数据

工作量的跟踪

Measures

- 每项任务的计划工作量与实际工作量
- 项目组总体的计划工作量与实际工作量
- 每类任务的计划工作量与实际工作量
  - -评审的工作量投入
  - -QA的工作量投入
  - -CM的工作量投入
  - ……
- 每周跟踪，分析差异

工作量偏差分析

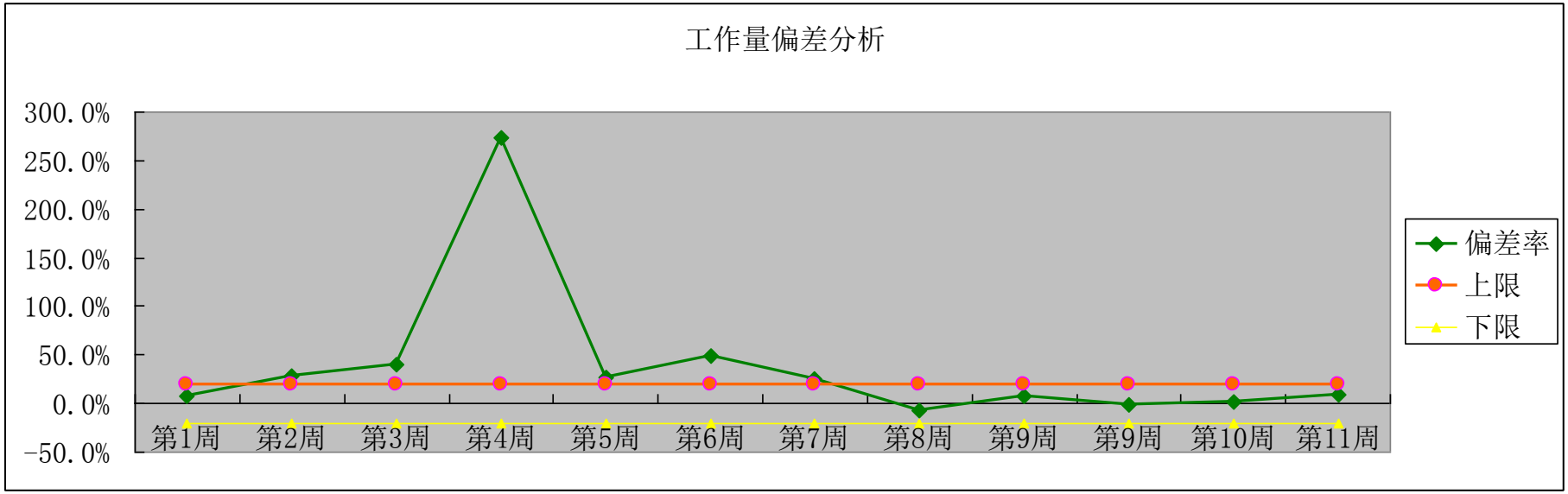
Measures

时间段	合计	第1周	第2周	第3周	第4周	第5周	第6周	第7周	第8周	第9周
计划的人日	170.94	18	24	20	9	35	20	16	14.18	14.76
实际的人日	235.32	19.5	30.7	27.9	33.6	44.5	29.87	20.1	13.25	15.9
偏差率	37.7%	8.3%	27.9%	39.5%	273.3%	27.1%	49.4%	25.6%	-6.6%	7.7%
偏差上限	20.0%	20.0%	20.0%	20.0%	20.0%	20.0%	20.0%	20.0%	20.0%	20.0%
偏差下限	-20.0%	-20.0%	-20.0%	-20.0%	-20.0%	-20.0%	-20.0%	-20.0%	-20.0%	-20.0%

工作量偏差分析

Measures

- 第二周的比率已经超过阈值，通过调整PP进行了调整。
- 第三周的比率超过阈值，由于原估计的工作量与实际的工作量有很大的偏差，同客户沟通中。
- 第四周的比率超过阈值，通过与客户的沟通，对PP进行第二次调整。
- 第五周的比率超过阈值，工作量超负荷，与客户进行沟通。
- 第六周中客户将第一个迭代和第二个迭代的测试终了日调整到5月27日。
- 第七周结束的时候完成了第一个迭代和第二个迭代的测试工作。



工作量分布的分析

Measures

- 按阶段的工作量分布
- 按工种的工作量分布
- 特定活动的工作量分布
  - 返工工作量
- 计划内外的工作量分布

建立工作量分布模型的步骤

Measures

- 收集原始数据
- 验证原始数据
- 整理原始数据
  - 比如：将交付与维护合并，将项目管理类合并
- 进行稳定性分析
- 确定上下控制限

22个项目的原始数据

序号	需求	设计	开发	测试	交付	管理活动	总工作量
1	155	163	308	118	256	175	1175
2	107.5	126.2	345.19	101.69	9	175	864.58
3	8.25	14	182.7	123.7	10.7	117.15	456.5
4	153	208	366	266	7	330	1330
5	4.125	48.9375	43.0625	122.125	287.875	20.9375	527.0625
6	1.31	2.44	4.5	7.25	10.63	17.25	43.38
7	75	45	191	81	48	281	721
8	460.24	376.56	857.72	146.44	209.2	209.2	2259.36
9	200	218.75	412.5	300	99.75	287.5	1518.5
10	312	80	118	92	263	26	891
11	1.31	2.44	4.5	7.25	10.63	17.25	43.38
12	41	26	12	29	5	16	129
13	20	30	150	30	245	70	545
14	102.5	236.63	526.23	166.44	5.38	100.38	1137.56
15	38.13	158.87	265.19	242.13	143.01	38.2	885.53
16	184.13	198.6	510.88	360.06	73	199.83	1526.5
17	61.25	154.38	314.38	328	44.75	64.5	967.26
18	25.25	168	255.06	120.31	74.58	125.88	769.08
19	7	380	281	222	0	0	890
20	196	4	619	310	20	51	1200
21	51	5	523.5	228.5	0	55.5	863.5
22	168.5	196.2	294.2	238	0	204	1100.9

Measures

如何分析生产率

生产率的含义

Measures

- 生产率=规模/工作量
- 软件规模的度量：
  - 物理规模：代码行
    - 对于代码行，要注意明确定义如下的规则：
      - 是否包含注释行？
      - 是否包含空行？
      - 是否包含机器自动生成的代码？
      - 是物理行还是逻辑行？
  - 逻辑规模：功能点
- 文档规模的度量：
  - 页数
- 测试用例规模的度量
  - 个数
- 工作量的度量：
  - 2种维度的度量：
    - 按阶段：
      - 全生命周期的工作量
      - 编码阶段的工作量
  - 按是否含管理：
    - 含管理活动的工作量
    - 不含管理活动的工作量

案例：设计生产率分析

Measures

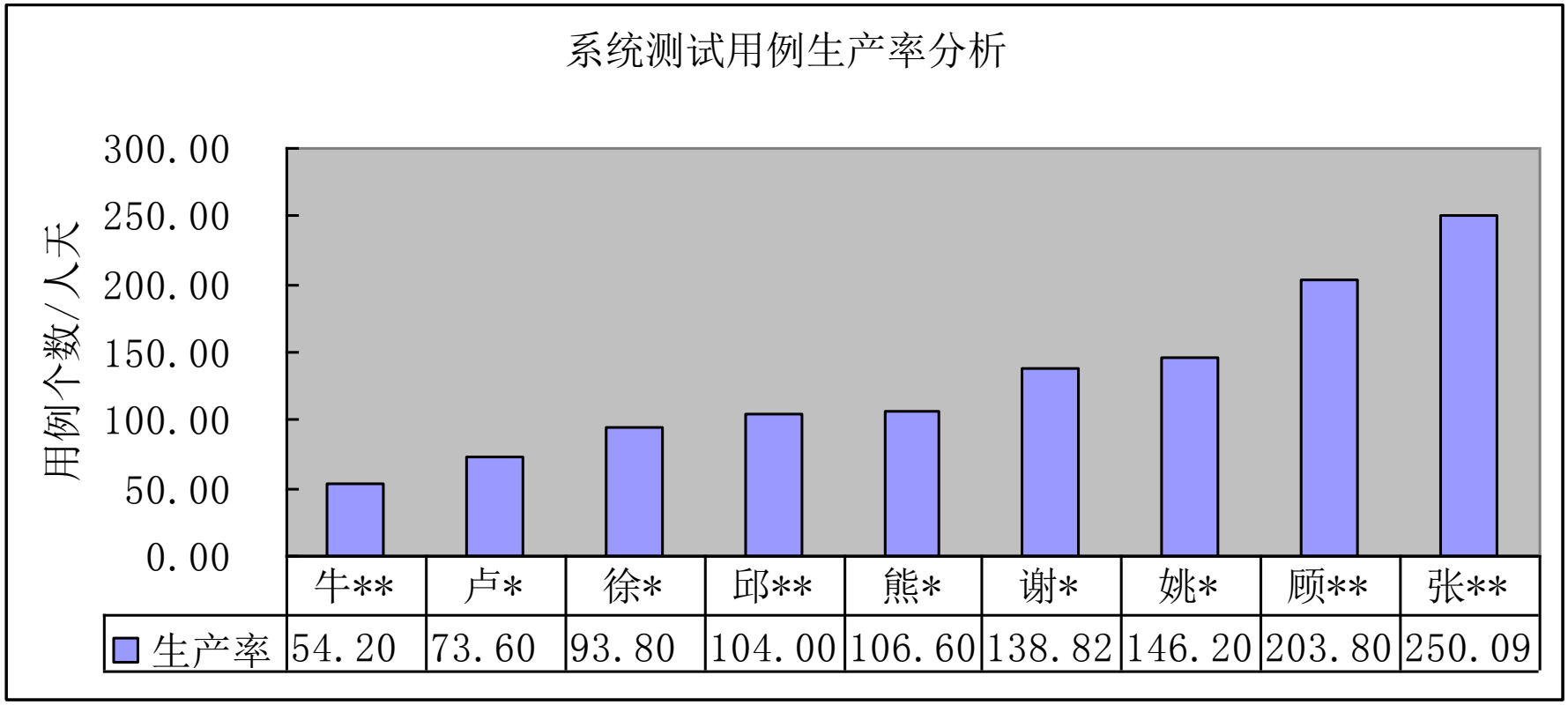
序号	设计者	实际工作量(人天)	实际规模(页)	生产率	原因
1	吴**	17	10	0.59	本项目没有可参考数据，在中定义了6page/人天的质量标准，但本次不作考核。
2	耿**	17	28	1.65	
3	强**	25	132	5.28	
4	王*	20	155	7.75	
5	喻*	26	208	8.00	
6	张**	17	143	8.41	
7	舒**	19	166	8.74	
8	黄**	22	198	9.00	
9	孙**	22	206	9.36	
10	乔*	2	31	15.50	

案例：个人编码生产率分析

Measures

序号	程序员	预计工作量(人天)	预计规模(行)	实际工作量	实际规模	实际生产率	偏差值	原因
1	陈***	19.5	4485	22	1613	73.3	-68.1%	承担了共通的维护
2	陈**	11	2530	26	2786	107.2	-53.4%	上项目的时间较短
3	沈*	11	2530	27	4849	179.6	-21.9%	承担了Leader的Review和其他事务
4	陈*	24	5520	31	9734	314.0	36.5%	开发人员UT1不足
5	孙**	18	4140	21	7016	334.1	45.3%	代码Review和调试不足
6	储*	31	7130	33	11667	353.5	53.7%	画面比较简单
7	马*	12	2760	15	5315	354.3	54.1%	工作不认真，bug较多
8	蔡**	25.5	5865	34	13932	409.8	78.2%	承担的页面比较相似
9	舒**	4.5	1035	6	2531	421.8	83.4%	页面比较简单
10	李*	26	5980	34	14425	424.3	84.5%	业务、功能简单。UT不充分，工作不认真产生2个NG
11	黄**	27	6210	27	12760	472.6	105.5%	画面功能简单,多为字段的显示
12	孙*	24.5	5635	29	14138	487.5	112.0%	加班严重

案例:系统测试用例生产率分析



Measures

如何使用度量数据管理评审过程

同行评审中的度量数据

Measures

- 基本度量元
  - 评审规模
  - 评审工作量
  - 评审时间
  - 参与的人数
  - 缺陷个数
- 派生度量元
  - 评审速率
  - 评审效率
  - 缺陷密度
- 度量对象
  - 个人
  - 总体

评审度量数据的分析-基本度量元

Measures

- 评审规模
  - 每次评审的规模上限
    - 需求、设计 20页
    - 测试用例 20页
    - 代码 1000行
- 评审时间
  - 222法则
    - 个人评审准备时间不要超过2小时
    - 预评审会议时间在半小时左右，最长不要超过2小时
    - 记录会议时间不要超过2小时
- 评审员数量
  - 3-7人

评审度量数据分析-派生度量元

Measures

- 派生度量元
  - 评审速率
  - 评审效率
  - 缺陷密度

工组产品类型	每小时数
需求	250行（5页）
概要设计	200行（4页）
详细设计	150行（3页）
源码，测试用例	150行（无注释） （4页）
测试计划	200行（4页）

Inforguard2次评审的度量数据

Measures

	评审工作	单位	规模	评审时间(人时)	一级缺陷	二级缺陷	三级缺陷	建议类缺陷	缺陷总数	缺陷密度	评审速率
2007年9月	《InforGuard V4产品需求规格说明书》	页	*63*	*41.0*	*0*	*2*	*24*	*35*	*61*	*0.97*	*1.5*
	《InforGuard V4概要设计说明书》	页	*29*	*19.0*	*0*	*1*	*16*	*7*	*24*	*0.83*	*1.5*
2008年3月	《InforGuard V4产品需求规格说明书》	页	*19*	*12.0*	*0*	*5*	*10*		*15*	*0.79*	*1.6*
	《InforGuard V4概要设计说明书》	页	*25*	*20.0*	*2*	*3*	*15*		*20*	*0.80*	*1.3*
	《InforGuard V4.0详细设计说明书》	页	*120*	*16.0*	*0*	*2*	*30*		*32*	*0.27*	*7.5*

Inforguard2次评审数据与目标比较

Measures

	度量项	单位	质量目标						偏差情况分析
			下限	上限	目标	实际值	缺陷数	规模	
2007年9月	《InforGuard V4产品需求规格说明书》	个/页	0.95	1.15	1.00	0.97	61	63.0	度量项进行内部评审时缺陷数统计不足
	《InforGuard V4概要设计说明书》	个/页	0.68	0.82	0.80	0.83	24	29.0	
2008年3月	《InforGuard V4产品需求规格说明书》	个/页	2.00	4.00	1.05	0.79	15	19.0	
	《InforGuard V4概要设计说明书》	个/页	0.95	1.15	0.75	0.72	18	25.0	
	《InforGuard V4.0详细设计说明书》	个/页	0.68	0.82	11.50	0.27	32	120.0	

案例：代码走查的度量数据分析

Measures

	代码行	问题个数	人数	准备工时	会前问题数	会议时间	会议问题个数	总工时	会期	发现效率	缺陷密度
多媒体8.3	591	11	7	4.7	9	4.7	2	9.4	0.7	1.17	18.61
马蹄莲主模块	2230	47	7	11.2	32	8.2	15	19.4	1.2	2.42	21.08
马蹄莲设置模块	1523	35	7	8.3	34	7	1	15.3	1	2.29	22.98
马蹄莲第3次设置	1963	18	8	6.3	15	8	3	14.3	1	1.26	9.17

分析同行评审数据

Measures

如果发现的缺陷少于正常情况	
可能的原因	考虑采取的措施
工作产品非常简单	将类似工作产品的审查转变为其他方法
也许没有仔细评审	检查评审速度, 可能需要换人评审
评审专家没有接受足够的审查培训, 或对评审的材料没有足够多的经验	安排或执行培训 安排其他队伍进行重新评审
工作产品有非常好的质量	通过评审速度, 作者, 评审专家的经验来确认这个事实, 看该质量是否可以在项目的其他部分重新 修改随后活动的缺陷预测;看是否有通用的过程改进的经验

分析同行评审数据

Measures

如果发现的缺陷多于正常情况	
可能的原因	考虑采取的措施
产品质量很低	检查作者是否需要培训 重做产品 重新考虑分配将来的任务
产品非常复杂	确保在后续阶段有良好的评审或测试 为系统测试增加估计 将产品分成更小的组件
有太多的小错误(以及很少的大错误)	确定小错误的原因;通过适当增强检查表和使作者意识到常见错误原因,使错误在以后得到改正 评审者也许对工作产品没有充分的理解. 召开一个简短会议或者用不同评审专家另外评审
评审时所使用的参考文档不够详细和清楚	使参考文档得到评审并获批准
评审过的模块是项目中的第一批模块	分析缺陷,更新评审检查表并通知开发者安排培训

同行评审数据分析练习

Measures

- 针对以下7种同行评审的场景分析，挖掘数据背后的问题。
  - 场景1：某次需求审查，个人评审阶段发现的缺陷为10个，会议上发现的缺陷为20个
  - 场景2 ：某次设计审查30页文档，平均个人评审花费的时间为1小时。
  - 场景3：某次代码走查，花费了1个小时，评审了1000行代码
  - 场景4：审查20页的需求文档，有5个专家参与，其中2个专家A花费了1小时进行了个人评审，其他3位专家没有进行个人评审。
  - 场景5：某次代码审查，专家A的个人评审速率为：1000行/小时，其他专家的个人评审效率约为300行/小时。
  - 场景6：某次需求审查的效率为1.8个/人时，组织级建立的基线为 0个/人时---1.6个/人时
  - 场景7：某次需求评审持续进行了1天的时间。

Measures

推荐的测试度量元

测试度量元-1/5

Measures

序号	优先级	度量对象	度量元	计量单位	采集周期	采集/计算方法	分析方法	作用
1	1	用户发现的各类型的缺陷	缺陷个数	个	交付阶段 维护阶段	直接统计	80-20分析：对缺陷类型按缺陷个数排序，找出客户发现的最多的20%的缺陷类型	分析客户的关注点是什么？为什么客户能发现这些类型的缺陷，为什么我们没有测试出来？定义改进措施
2	1	软件模块	缺陷密度	个/KLOC	系统测试阶段	缺陷个数/代码规模	80-20分析：对所有模块的缺陷密度进行排序比较，找出缺陷密度最大的20%模块	找出质量最差的模块，采取改进措施
3	1	遗留的缺陷	缺陷个数	个	系统测试阶段	上阶段遗留的缺陷个数+本阶段发现的缺陷个数-本阶段解决的缺陷个数	和阶段出口准则对比	里程碑评审决策的依据

测试度量元-2/5

Measures

序号	优先级	度量对象	度量元	计量单位	采集周期	采集/计算方法	分析方法	作用
4	1	各级别严重程度的缺陷	缺陷个数	个	系统测试阶段	直接统计	和项目目标对比	判断是否达到测试结束与产品发布准则
5	1	回归测试活动	缺陷个数	个	系统测试阶段	直接统计	趋势线分析：横坐标为某次回归测试，纵坐标为缺陷个数，建立拟合曲线，判断收敛性	针对每次回归测试活动，进行收敛分析，作为发布决策的依据
6	2	代码走查	代码走查的效率	个/小时	编码阶段	代码走查发现的缺陷个数/代码走查的工作量	和项目目标对比	比较评审与测试的效率，以确定二者投入工作量的比例

测试度量元-3/5

Measures

序号	优先级	度量对象	度量元	计量单位	采集周期	采集/计算方法	分析方法	作用
7	2	单元测试	测试效率	个/小时	编码阶段	单元测试发现的缺陷个数/单元测试的工作量	和项目目标对比	建立单元测试的性能基线，预测单元测试投入的工作量
8	2	系统测试	测试效率	个/小时	系统测试阶段	缺陷个数/测试工作量	和项目目标对比	建立测试活动的投入产出模型，用以估计为了达到项目的质量目标而需要的测试工作量
9	2	系统测试	系统测试用例相对逻辑规模的密度	个/功能点	系统测试阶段	系统测试用例个数/需求的规模	和项目目标对比	判断系统测试的充分性

测试度量元-4/5

Measures

序号	优先级	度量对象	度量元	计量单位	采集周期	采集/计算方法	分析方法	作用
10	2	系统测试	系统测试用例相对物理规模的密度	个/KLOC	系统测试阶段	系统测试用例个数/代码规模	和项目目标对比	判断系统测试的充分性
11	3	测试发现各类型的缺陷	缺陷个数	个	编码阶段系统测试阶段	直接统计	80-20分析：对缺陷类型按缺陷个数排序，找出最多的20%的缺陷类型	根据类型的分布，查找错误的原因，定义编码或设计的改进措施
12	3	单元测试	缺陷密度	个/KLOC	编码阶段	单元测试发现的缺陷个数/代码规模	和项目目标对比	建立单元测试的性能基线，用以制定项目的质量目标

测试度量元-5/5

Measures

序号	优先级	度量对象	度量元	计量单位	采集周期	采集/计算方法	分析方法	作用
13	3	单元测试	单元测试用例相对物理规模的密度	个/KLOC	编码阶段	单元测试用例个数（或断言的个数）/KLOC	和项目目标对比	判断单元测试的充分性
14	3	集成测试	集成测试用例相对物理规模的密度	个/KLOC	集成阶段	集成测试用例个数/KLOC	和项目目标对比	判断集成测试的充分性
15	3	系统测试	测试用例的有效性	%	系统测试阶段	依据测试用例发现的缺陷个数/所有的缺陷个数	和项目目标对比	评价测试用例的有效性，判断是否需要提高测试用例的设计技术

Measures

如何设定量化管理目标？

如何设定量化管理目标

Measures

- 量化管理目标的来源：
  - 基于客户需求
  - 基于历史数据
  - 基于标杆数据
  - 基于经验数据
- 量化管理目标的分类
  - 进度
  - 质量
  - 工作量与成本
  - 过程性能
  - 客户满意度
  - 规模

管理美国国防部项目的定量目标

Measures

项目	目标	不当等级
缺陷清除效率	>95%	<70%
初始缺陷密度	每个功能点<4	每个功能点>7
超过风险储备的额外开支	0%	>10%
需求渐变（功能点或相当的事物）统计	平均每月<1%	>50%
程序的文档记录总计	每个功能点<3页	每个功能点>6页
人员流动	每年1%-3%	每年>5%

控制限的设置

Measures

- Infosys公司的控制限：
  - 早期（经验确定）：
    - 工作量：35%
    - 工期：15%
  - 改进后（统计结果）：
    - 工作量：20%
    - 工期：10%
- 项目早期的阶段，控制限较宽，项目后期，控制线较窄

Infosys公司的缺陷分布

Measures

过程阶段	占总缺陷的百分比
需求规格评审+概要设计评审+详细设计评审	15%-20%
代码评审+单元测试	50%-70%
集成测试+系统测试	20%-28%
验收测试	5%-10%

Infosys公司评审能力基准

Measures

评审项	准备期间的评审速度（如果不同于小组评审期间的评审速度）	小组评审期间的评审速度	装饰性缺陷/次要缺陷的缺陷密度	紧急缺陷/主要缺陷的缺陷密度
需求		5-7页/小时	0.5-1.5个缺陷/页	0.1-1.3个缺陷/页
概要设计		4-5页/小时	0.5-1.5个缺陷/页	0.1-0.3个缺陷/页
详细设计		3-4页/小时	0.5-1.5个缺陷/页	0.2-0.6个缺陷/页
编码	160-200LOC/小时	110-150LOC/小时	10-60个缺陷/KLOC	10-60个缺陷/KLOC
集成测试计划		5-7页/小时	0.5-1.5个缺陷/页	0.1-0.3个缺陷/页
集成测试用例		3-4页/小时		
系统测试计划		5-7页/小时	0.5-1.5个缺陷/页	0.1-0.3个缺陷/页
系统测试用例		3-4页/小时		
项目管理和配置管理计划	4-6页/小时	2-4页/小时	0.6-1.8个缺陷/页	0.1-0.3个缺陷/页

小结

软件度量的成功与失败教训

Measures

- 失败的教训
  - 目的不明，事后发现度量的内容与管理无关
  - 开发人员拒绝执行，认为会否认其工作业绩
  - 要求广泛收集数据，程序烦琐，不堪重负
  - 管理者感觉到可能发生问题或者没有成功的结果，而放弃支持度量工作
- 成功经验：
  - 收集有用的度量数据
  - 度量的结果真正用于决策
  - 坚持就是胜利

小结

- 需要采集什么数据？
- 如何展示数据?
  - 五种基本图形
  - 如何设计指示器
- 如何采集与校验数据?
- 如何定义度量体系?
- 如何分析数据?
  - 基本数据分析技术
  - 如何编写分析报告
  - 过程稳定性分析技术
  - 性能基线的建立方法
- 度量应用实例
  - 如何分析与使用工作量数据？
  - 如何跟踪项目进展？
  - 如何分析与使用生产率?
  - 如何使用度量数据管理评审过程？
  - 如何设定量化管理目标？
- 小结

Measures

谢　谢！