

工作量估算方法

麦哲思科技（北京）有限公司

- 任甲林
 - 高级咨询顾问，主要从事提升软件研发能力的培训与咨询
 - 联系方式：
 - Mobile: 15863181188
 - E-mail: renjialin@measures.net.cn
 - MSN: dylan_ren@msn.com
 - Blog:
 - <http://dylan1971.blog.ccidnet.com/>
 - 工程经验：
 - 93年从事软件开发, 参与了50多个项目的开发
 - 曾为北京汉王、上海鹏开、深圳富士康、深圳鹏开、四川长虹、大连华信、成都虹微、昆山中创、北京信城通、南京诚迈、南京润和、珠海矩力等多家公司咨询

- 工作量估算的一般原理
- Pert Sizing估算方法
- 宽带DELPHI方法估算方法
- 策划扑克法
- 七种场景下的工作量估算方法

工作量估算的一般原理

软件工作量估算困难的原因

- 软件开发是复杂的和不可见的
- 软件开发是人力密集型工作的，因而不能以机械的观点来看待
- 传统的工程项目经常会以相近的项目做参考，不同的只是客户和地点，而绝大部分软件项目是独一无二的
- 新技术的不断出现和应用
- 缺少项目经验数据，许多组织无法提供原有项目数据，而即使提供了这些项目数据，也未必非常有用

- 帕金森定律 (Parkinson's Law)
 - “**工作总是用完所有可利用的时间** (Work expands to fill the time available)”，这意味着容易达到的目标将使员工工作上变得松懈
- 布鲁克斯定律 (Brooks' Law)
 - 实现一个项目需要的工作量不是与分配到项目的员工数成比例地增长。当项目组的规模增长时，投入管理、协调和沟通的工作量也在增长。
 - 极端情况下，Brooks定律会出现这样的情况：“**投入更多的人到一项延迟的工作上，可以导致该项工作更加延迟**”。
- Barry Bohem曾经提出“可以将软件开发进度压缩25%，但是不能再多了”

过于乐观的进度计划

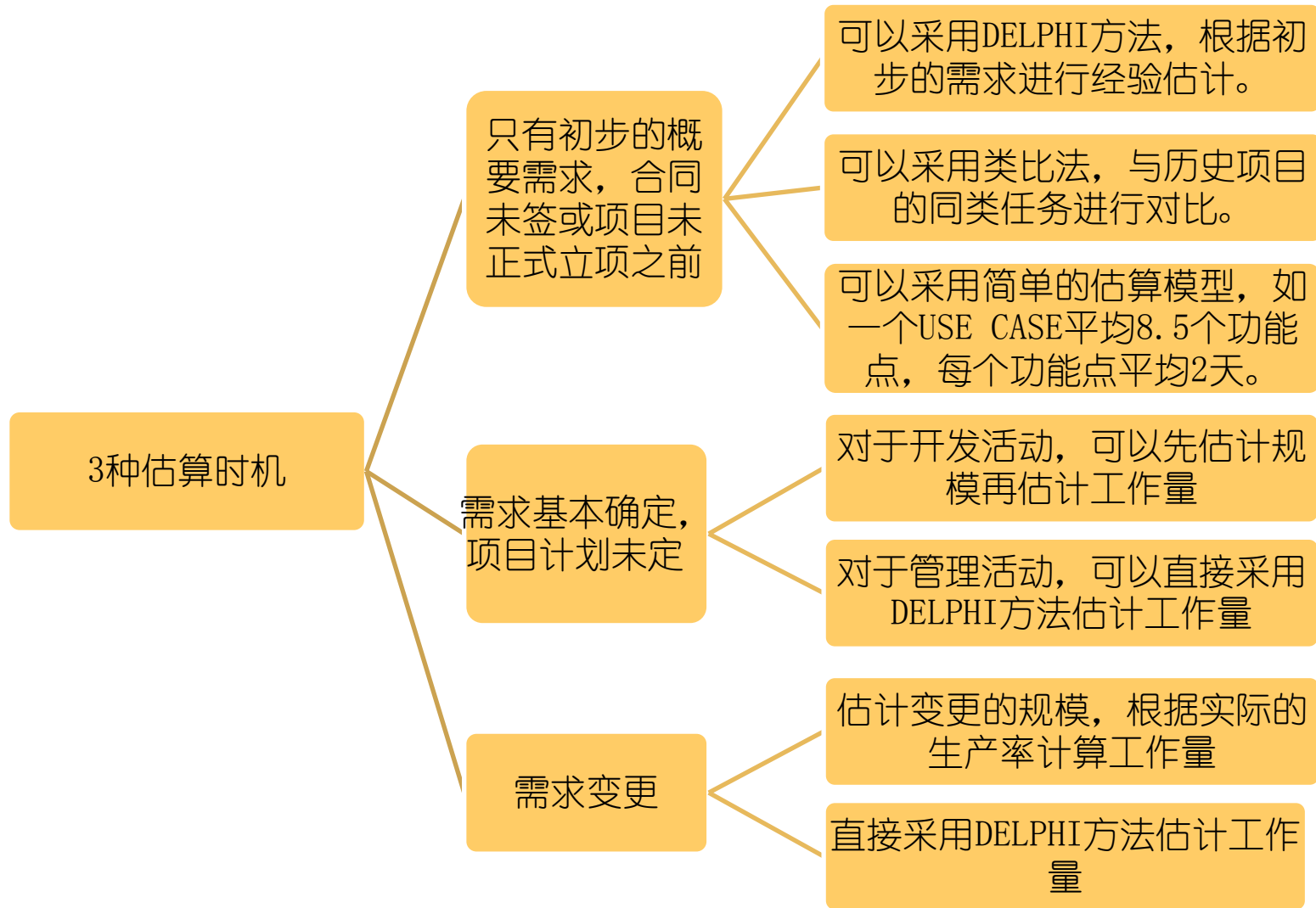
- Microsoft Word for Windows 1.0开发
- 包含249,000行代码,投入660人月,前后历时5年,实际花费时间为预期时间的5倍

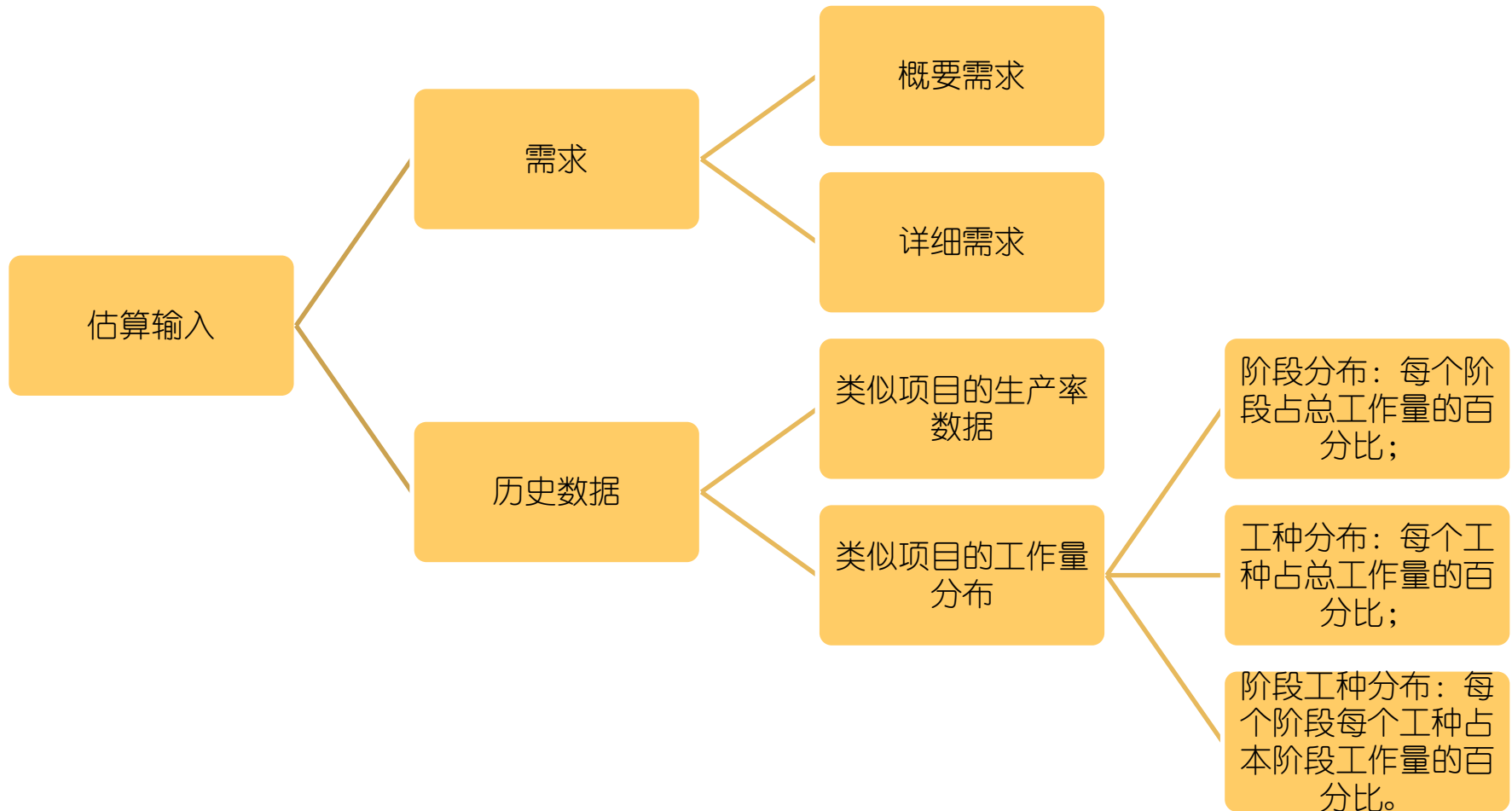
提交报告日期	预期交付日期	预期到完成交付还需要的开发时间	实际到完成交付所需花费的开发时间	估算偏差率
84年9月	85年9月	365天	1887天	81%
85年6月	86年7月	395天	1614天	76%
86年1月	86年11月	304天	1400天	78%
86年6月	87年6月	334天	1235天	73%
87年1月	87年12月	334天	1035天	68%
.....
89年6月	89年9月	92天	153天	40%
89年7月	89年10月	92天	123天	25%
89年8月	89年11月	92天	92天	0%
89年11月	89年11月	0	0	0%

过于乐观的进度计划

- 导致WinWord1.0开发延迟的几个主要因素：
 - 项目初期制定的开发目标是不可实现的。盖茨下达的指示是用最快的速度开发最好的字处理软件，争取在12月内完成。实现这两个目标中的任何一个都是困难的，同时达到则是不可能的。
 - 过紧的进度计划降低了计划的精确度。
 - 开发过程中频繁换人。5年中共换了4个组长，其中有2人因进度压力离职，1人是出于健康的原因而离职。
 - 迫于进度压力，开发人员匆忙写出一些低质量的和不完整的代码，然后宣称已实现某些性能。这造成了WinWord不得不将用于提高软件稳定性的时间由预计的3个月增加到12个月。
- 由于该项目中，创新比速度更重要，因而试图缩短开发周期，反而使周期变长

3种工作量估算的典型时机





(1) 有文档产出的活动

- 文档规模敏感的活动，即规模是影响其工作量的主要因素。先估计文档规模，再估计活动的工作量。软件开发活动一般是规模敏感的活动，但是也有的软件开发是复杂度敏感或其他特征敏感的，比如算法类的软件。
- 文档规模不敏感的活动，即工作量与规模关系不大。采用经验法直接估计工作量，如编写管理类文档的活动可能就和规模关系不大。

(2) 有代码产出的活动

- 先估计代码的规模，再根据生产率、复用率、难度系数等估计工作量。

(3) 无工作产品产出的活动

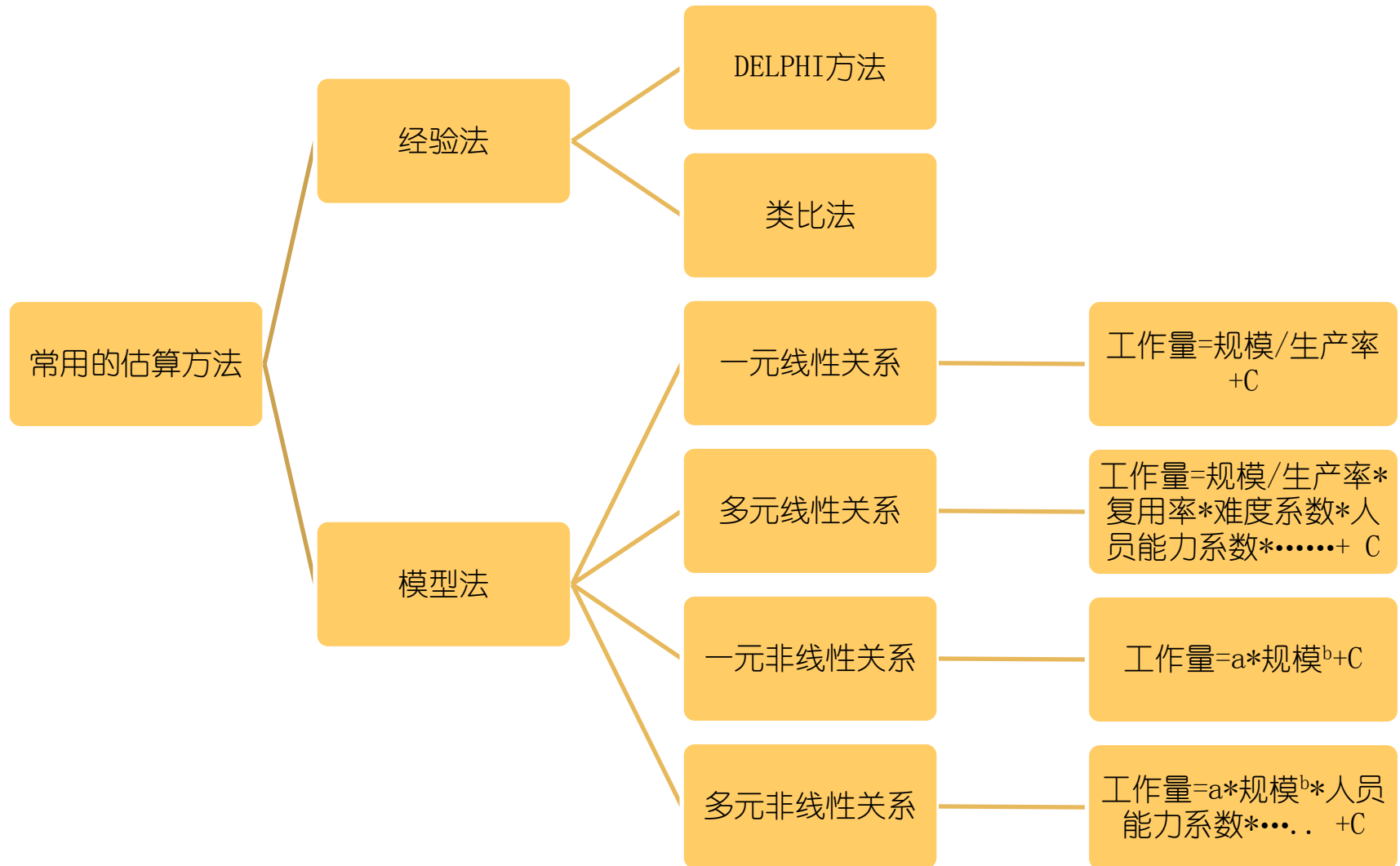
- 采用经验法，直接估计工作量

- 4种颗粒度
 - 整个项目的总工作量
 - 某类任务的工作量
 - 某个阶段的工作量
 - 某个任务的工作量
- 不同颗粒度之间的换算关系：
 - 项目的总工作量=所有任务类型的工作量之和；
 - $\quad\quad\quad$ =所有任务的工作量之和；
 - $\quad\quad\quad$ =所有阶段的工作量之和；
 - 某个阶段的总工作量=该阶段的所有任务的工作量之和；
 - $\quad\quad\quad$ =该阶段的所有任务类型的工作量之和；
 - 某种任务类型的工作量=所有阶段的该类型的任务的工作量之和；

这些换算关系可以用来校验从不同的角度进行估算时的可接受性

。

常用的工作量估算方法

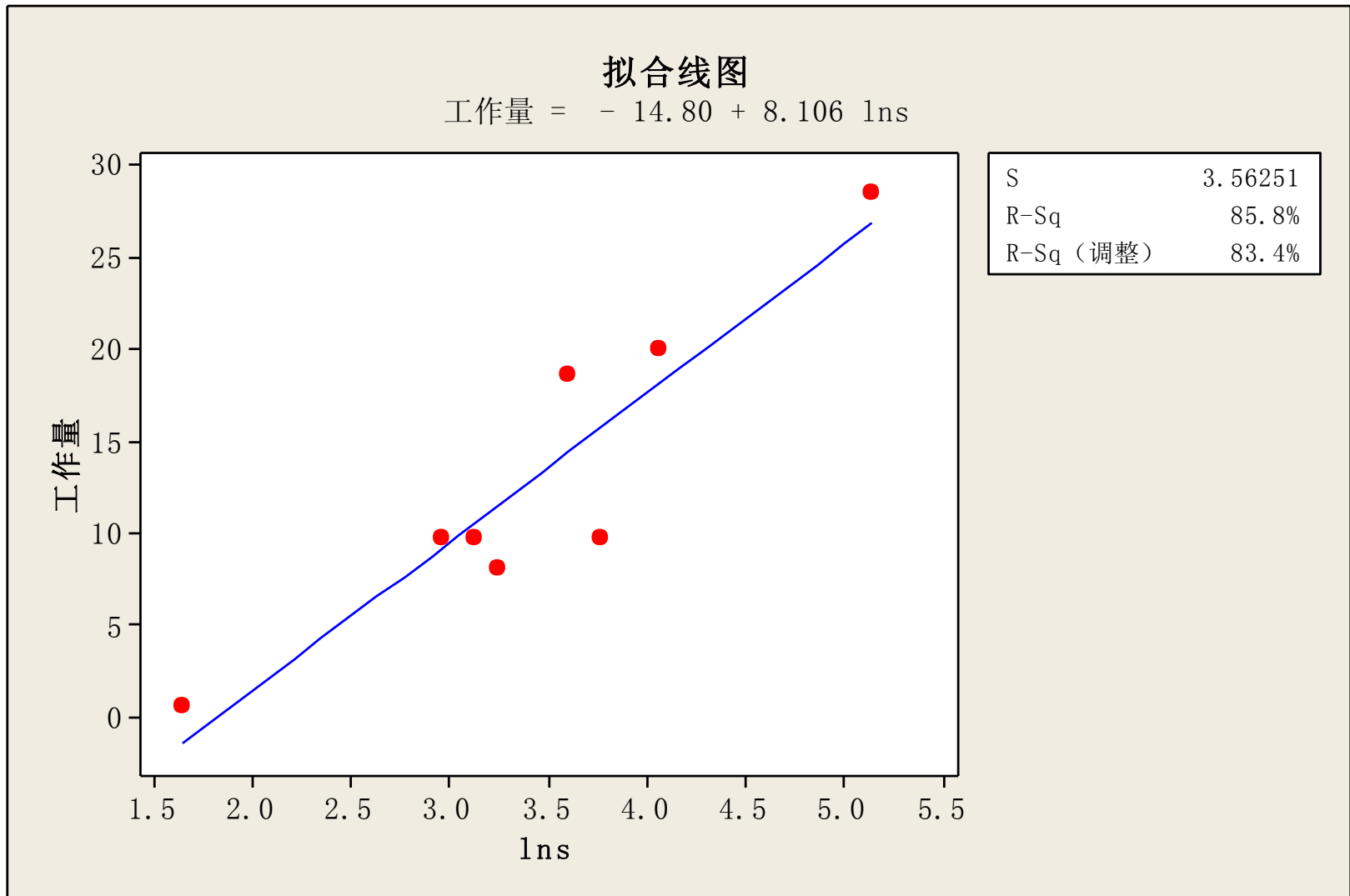


某公司JAVA语言的规模与工作量的回归分析

项目编号	规模	工作量
2007001D-AS-AML-AD	171.42	28.6
2007005D-AS-BSD	43	9.75
2007007D-AS-AMLCP	25.711	8.05
2007008D-IDEV4-NTTR-COM	58	20
2006_007_itf	19.22	9.727
2007_001_Projectpump 3	5.2	0.52
2006_007_itf	22.88	9.727
2006_002_fuhgetsu	36.74	18.6

$$\text{工作量} = -14.8 + 8.11 \ln \text{规模}$$

拟合结果的观察



工作量估算成功的6个要点

(1) 有经验的人参与估算

- 要对估计的内容有类似或相同的实践经验
- 要经过了估计的训练

(2) 任务定义要明确，分解的颗粒度要小

- 任务的交付物是什么
- 对任务完成的质量要求是什么？
- 任务的输入是什么？
- 究竟是做什么？

(3) 确保任务没有遗漏

- 如果估计的内容遗漏了，显然整体的规模就会有偏差，所以穷举所有的任务是最基本的工作。

(4) 要借鉴历史数据

- 是否采用相同的技术平台？
- 是否是同类型的软件，比如都是嵌入式软件或者都是MIS软件？
- 是否项目的规模相近？
- 是否采用相同的生命周期模型？
- 是否人员的专业技能相近？

(5) 采用多种方法互相验证

- 对多种方法的估计结果进行对比，通过对比发现差异比较大，然后再仔细分析差异原因，提高估计的合理性。
- 采用不同的估算方法得出的估算值，偏差范围在5%以内，即可以接受估算结果

(6) 在项目进展过程中要持续估算，逐渐优化

某项目估算偏差率分析

OX项目 (MIS软件, JAVA) 2次估计偏差率分析

	项目	计量单位	SRS确定前估计	SRS确定后估计	差异
A	编码与单元测试工作量	人天	30	38	26.67%
B	其他活动工作量	人天	145	282	94.48%
A+B	工作量总计	人天	175	320	82.86%
A/(A+B)	编码工作量比例 =A/(A+B)		17.1%	11.9%	-30.73%
C	估计的LOC	LOC	3898	5016	28.68%
C/A	编码阶段生产率	LOC/人天	130	132	1.59%
C/(A+B)	全生命周期生产率	LOC/人天	22	16	-29.63%
数据分析	2次估计的工作量偏差主要在于非编码工作量的偏差，达94.5%。2次估计的规模偏差达28.7%，而编码的生产率基本保持一样，全生命周期的生产率下降了30%				17

Pert Sizing估算方法

- Pert Sizing是一种加权平均法，可以用于估计软件项目的规模、工作量和成本等
- 在估计每一项任务时，首先按最佳的、可能的、悲观的三种情况给出估计值，记作：a、m、b
- 然后用以下公式计算期望值，期望值就是最终的估计值：

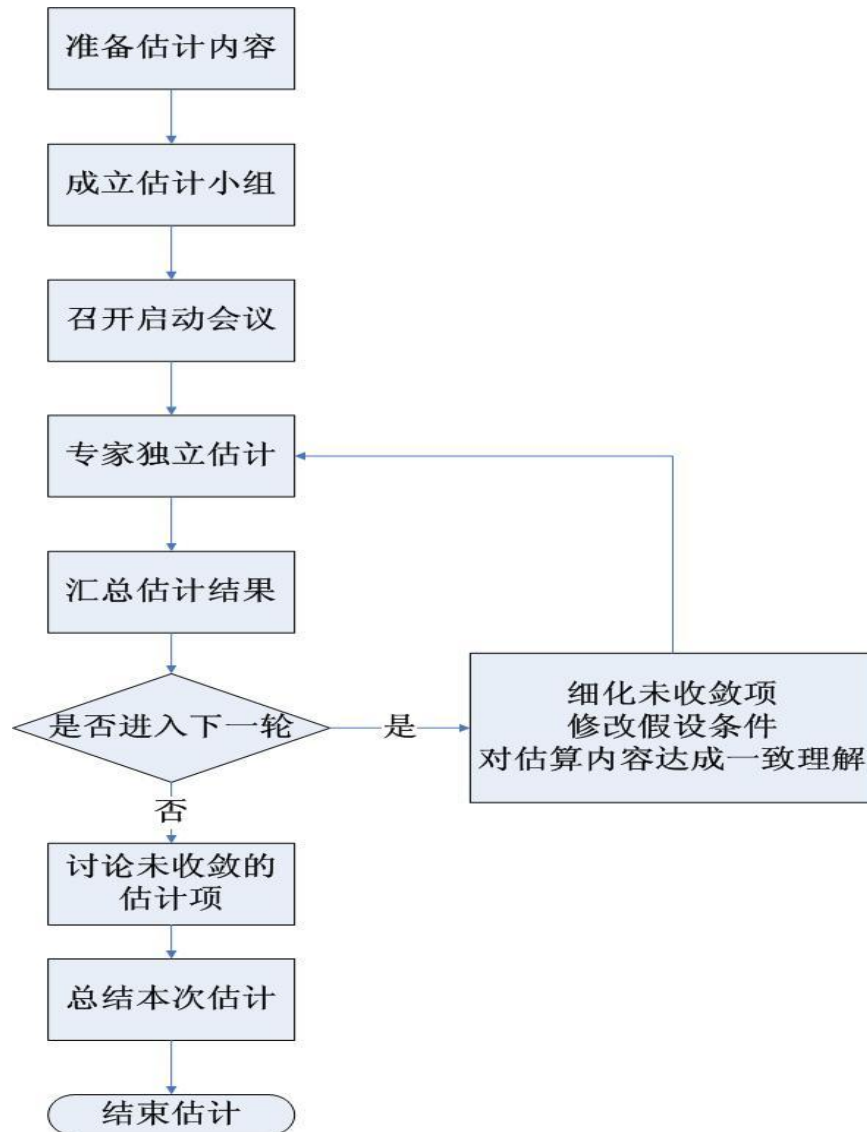
$$\text{期望值} = (a + 4 * m + b) / 6$$

- 在该公式中，包含了如下的2个原理：
 - **原理一**：对公式1进行变换，引入一个中间结果 c ， a 、 m 、 b 的平均值，则可以看出，也就是说， E 是 a 、 m 、 b 三数的平均数与 m 又求了一次平均。
 - **原理二**：从上面的公式也可以看出，在计算 E 的公式中， m 的权重是4，而 a 和 b 的权重都是1，也就是说，该方法强调了中间值的重要性，而最大最小值的重要性并不高。

- 当只有一个人参与估算时，则需要估算人估算三个数值：悲观值、可能值、乐观值，然后套用公式1就可以了。
- 当有2个人参与估算时，则需要2个估算人分别估算三个数值：悲观值、可能值、乐观值，然后分别计算这3个数值的平均值，再将悲观值、可能值、乐观值的平均值代入公式1就可以了。
- 当有3个人参与估算时，有2种方法：
 - 类似第（2）种情况处理，此时总共需要估计9个数；
 - 每个人只估计一个数，取最大最小值作为悲观值与乐观值，取中间的数值作为可能值，代入公式1。
- 当有N个人（ $N > 3$ ）参与估算时，也有2种方法：
 - 类似第（2）种情况处理，此时总共需要估计 $3N$ 个数；
 - 每个人只估计一个数，取最大最小值作为悲观值与乐观值，对中间的 $N-2$ 个数值取平均值作为可能值，代入公式1。
- $N < 6$

宽带DELPHI方法估算方法

宽带DELPHI方法估算流程



活动1：准备估计的内容。

- 准备估计的内容时，有两个要点：
 - （1）完备的识别被估计的内容。
 - （2）尽管Delphi方法既可以对颗粒度比较大的任务进行估计，也可以对颗粒度比较小的任务进行估计，但是还是要细分任务，其目的是为了
- 提高估计的准确度与加快估计值收敛的速度。如果让一个人去估计一幢大楼的使用面积和去估计一个房间的使用面积，估计的准确率显然差别很大。由作者将被估计的内容细分为更小颗粒度的问题，这样更容易把握，估计时更容易快速收敛。

活动2：成立估计小组。

- 估计小组有协调人、作者和3到6名估计专家组成。
- 协调人负责计划和协调软件估计活动，协调人在担任此角色时不能用自己的观点去引导专家，也不能因为自己的认识或偏见而对软件估计的结果进行歪曲。
- 协调人不能是作者，也不必作为专家。
- 专家的数量不宜太多，否则成本比较高。
- 专家必须具备2个条件：
 - （1）有业务与技术经验，熟悉被估计的内容；
 - （2）要有估计的经验，接受过估计方法的培训，并曾经在实践中评价过自己的估计准确率。

活动3：召开启动会议

- 协调人负责召开启动会议，在启动会议上主要进行以下工作：
 - 作者向专家介绍估计的内容、项目的各种假设和限制条件。
 - 专家对被估计的内容达成一致，并确认各种假设和限制条件。
 - 专家对估计结果的度量单位达成一致。
 - 比如估计软件的规模时，是用行还是千行作为计量单位，代码行是否包括注释行、空行、开发平台自动生成的语句等要达成一致。
 - 对估计结束的准则达成一致。
 - 结束的准则包括：
 - 估计结果可接受的判断方法，即估计结果在多大的偏差范围内是认为是可接受的，此时称为估计结果收敛。
 - 在连续几轮无法收敛后，估计应该结束；
 - 对于不能收敛的估计内容如何确定估计结果。
 - 结束准则也可以由协调人和作者在活动2中确定。

活动4：专家独立估计

- 在专家进行独立估计时，需要注意：
 - 专家的估计活动不应受外界压力的影响，协调人或者作者不能给出估计结果的上下限或其他限定。
 - 各专家之间没有讨论和咨询。
 - 各专家采用的估计方法也不受限制。各专家即可以根据自己的经验估计，也可以采用类比的方法进行估计。
 - 如果专家认为被估计的内容中存在不明确的地方，应记录自己所做出的各种假设。

活动5：汇总估计结果

- 收集各专家的估计结果，制作本轮的估计结果表：
- 差异率的计算方法以及接受的准则应该在活动2或者活动3时确定，比如可以采用如下的差异率计算公式及接受准则：
 - (1) 差异率 = $\max(\text{最大值} - \text{平均值}, \text{平均值} - \text{最小值}) / \text{平均值}$ ；
 - (2) 当差异率 < 25% 时接受平均值为最终估计结果。
- 需要注意的是对于颗粒度比较小的任务对差异率比较敏感，所以可能需要区别定义接受准则。
- 注意识别离群点，如果存在离群点，可以考虑删除之

Delphi方法估计结果记录表

轮次

任务ID	任务	最大值	最小值	平均值	差异率	是否接受

活动6：讨论不收敛项

- 将本轮的匿名估计结果公布给各专家
- 讨论不收敛的估计项
 - 在讨论时只对被估计的内容的理解进行讨论，不讨论每个人具体的估计结果，
- 为了达成一致的理解，此时可能需要重新细化估计内容或确定估计假设。

活动7：新一轮的估计

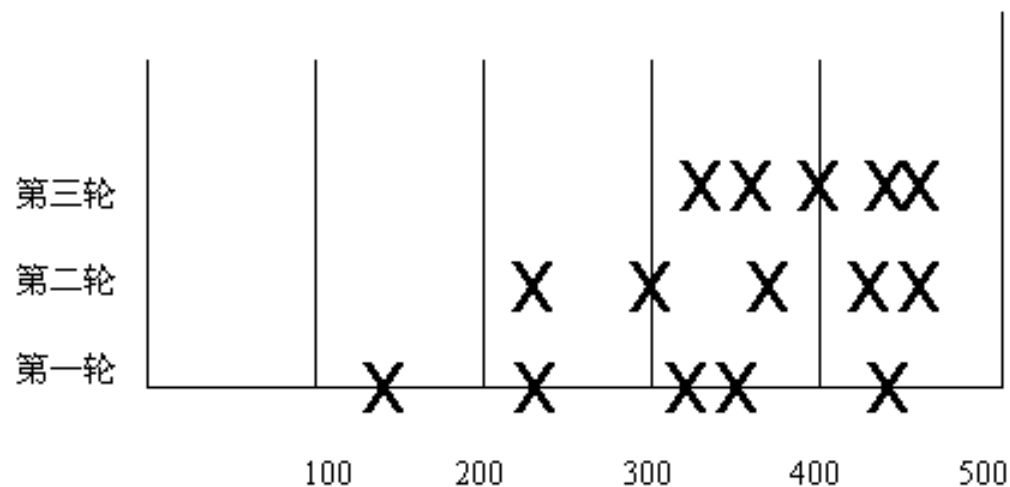
- 对于估计结果未被接受的估计项，执行活动4，如此不断反复，直至满足以下任何一项条件：
 - 完成多轮估计（如4轮），该条件已经在结束准则中确定了。
 - 所有的估计结果收敛于一个可以接受的范围；
 - 所有专家拒绝对各自的估计结果进行修改。

活动8：讨论未收敛的估计项

- 对于结束了多轮估计后，仍然未收敛的估计项，可以采用如下的方法确定最终估计结果：
 - 按照少数服从多数的原则，忽略少数与其他估计结果差异很大的估计结果。
 - 采用“掐头去尾求平均值”的方法。即：排除最大与最小值后，求剩余数值的平均值作为最终估计结果。
 - 由各专家进行讨论，形成一个一致的意见。
- 当然，也不限于采用其他方法确定最终的估计结果。

活动9：总结本次估计

- 估计小组对最后汇总的估计结果进行审核，并对结果达成一致。
- 估计小组可以对宽带Delphi方法进行思考，提出改进措施，以使将来的估计活动更加有效。



三轮估计的结果分布图

优点

- 可以产生非常准确的结果
- 利用组织的技能
- 适用任何规模的产品

缺点

- 依赖于少数专家
- 要消耗大量的时间
- 可能存在共同的偏见

- 个人名片系统的工作量估计
- Measures拟开发一套个人名片管理系统，需求如下：
 - (1) 运行在单机上，支持WINDOWS 2003, WINDOWS XP操作系统。
 - (2) 可以录入名片信息：姓名，公司名称，公司地址，邮编，固定电话，手机，职务，职称，网站地址等。
 - (3) 人员可以重名
 - (4) 除上述属性外，可以由操作员自定义增加不超过其他10个属性。
 - (5) 可以对这些人员进行分类。
 - (6) 可以按名片信息中的任何一个数据项进行任意组合条件查询，如查询姓张的、北京的等
 - (7) 可以将数据转成EXCEL导出，也可以导入EXCEL格式的名片信息
 - (8) 数据库系统不限，但是应该占用的硬盘空间尽可能小。
- 其他约束可以自行假设。

策划扑克法

策划扑克法



- 故事点只是一个计量单位的名称而已，你也可以给他命名为其他名字。
- 故事点其实不仅仅是对规模的度量，也包括了对需求复杂度等其他因素的度量。
- 故事点并非业界统一的一个度量单位，不象度量长度的单位：米，大家都知道1米有多长，你说的1米和他说的1米是等长的。
- 故事点仅对本项目具有近似相等的规模，不同的项目所定义的故事点很可能是不等的。

- 包括了所有开发人员：
 - 程序员
 - 测试人员
 - 数据库工程师
 - 分析师
 - 用户交互设计人员等等，
- 在敏捷项目中一般不超过10人。
- 产品负责人参与策划扑克法但是并不作为估算专家。

- ① 每位参与估算的开发人员发放一副估算扑克，扑克上边的数字标为斐波那契序列：1，2，3，5，8，13，20，40。
- ② 选择一个比较小的用户故事，确定其故事点，将该故事作为基准故事。
- ③ 选择一个用户故事。
- ④ 主持人朗读描述，主持人通常是产品负责人或分析师，当然也可以是其他任何人，产品负责人回答估算者提出的任何问题，大家讨论用户故事。
- ⑤ 每个估算者对该用户故事与基准故事进行比较，选择一个代表其估算故事点的牌，在主持人号令出牌前每个人的牌面不能被其他人看到，然后大家同时出牌，每个人都可以看到其他人打出的牌。
- ⑥ 主持人判断估算结果是否比较接近，如果接近则接受估算结果，转向（3）选择下一个故事，直至所有的用户故事都估算完毕，否则转向（7）。
- ⑦ 如果结果差异比较大，请估算值最高及最小的估算者进行解释，大家讨论，时间限定为不超过2分钟。如果大家同意，也可以对该用户故事进行更细的拆分。
- ⑧ 转向（5），一般很少有超过3轮才收敛的现象。

- 在估算完故事点后，可以凭经验估算一个故事点的开发工作量，从而得到所有的用户故事的工作量。
- 也可以进行试验，试着开发一个用户故事，度量花费的工作量，得到开发效率，即在本项目中一个故事点需要花费多少工时，再去估算所有故事的工作量。

练习：策划扑克法

- 30分钟
- 5-10人一组
- 针对家庭财务软件的用户故事采用策划扑克法进行规模估计与工作量估计。
- 假设：
 - 该系统为运行在手机中的系统
 - 采用C++语言开发
- 输出：
 - (1) 每个用户故事的故事点
 - (2) 每个用户故事的工作量估计
 - (3) 该方法的优缺点

七种场景下的工作量估算方法

七种场景下的工作量估算方法

- 合同前的工作量估算
- 基于详细需求的经验估计
- 互补使用经验法与模型法
- 由总体印证基于WBS的估计
- 三维印证基于WBS的估计
- 四维印证基于WBS的估计
- 需求变更的工作量估计

场景一：合同前的工作量估算- 场景描述

- 合同未签，需要给客户报价
- 有客户的概要需求，有类似的项目数据可供参考
- 需要估计整个项目的总工作量，以便于估算总成本，给客户报价

- （1）寻找类似的历史项目，进行项目的类比分析，根据历史项目的工作量凭经验估计本项目的总工作量；
- （2）进行WBS分解，力所能及地将整个项目的任务进行分解；
- （3）参考类似项目的数据，采用经验法估计WBS中每类活动的工作量；
- （4）汇总得到项目的总工作量；
- （5）与第（1）步的结果进行印证分析，根据分析结果，确定估计结果。

项目工作量粗估案例

功能分类	大			中			小			小计
	历史的平均工作量	本系统中的个数	估计工作量	历史的平均工作量	本系统中的个数	估计工作量	历史的平均工作量	本系统中的个数	估计工作量	
单据录入类	3	5	15	2	5	10	1	5	5	30
字典维护类	1	5	5	0.5	5	2.5	0.5	5	2.5	10
参数设置类	3	1	3	2	2	4	1	2	2	9
报表查询类	1	5	5	0.5	10	5	0.5	10	5	15
记账类	10	1	10	7	1	7	3	1	3	20
小计			38			28.5			17.5	84

- 30分钟
- 5人一组
- 将公司做过的产品或项目的软件功能分类、划分为大中小或复杂、一般、简单，针对每类功能的每种复杂度的软件总结其工作量
- 输出：

功能分类	规模分类		
	大	中	小

场景二：基于详细需求的经验估计

- 场景描述：
 - 只有详细需求，没有历史数据
- 估算步骤：
 - (1) WBS分解，将任务分解到一个人或者一个小团队可以执行的颗粒度；WBS分解时要识别出所有的交付物、项目管理活动、工程活动等。
 - (2) 采用经验法估计每个活动的工作量；
 - (3) 汇总得到：每个阶段的工作量、项目的总工作量。
- 其他说明：
 - 在该场景下，只使用了经验法，无法对结果进行印证，难以判断结果的合理性。

- （1）有类似项目的历史数据，但是没有积累历史项目的工作量分布数据
- （2）有编码活动的生产率数据
- （3）有详细需求

场景三：互补使用经验法与模型法-估算步骤

- (1) 产品分解，将系统分为子系统，子系统分解为模块；
- (2) WBS分解，将任务分解到一个人或者一个小团队可以执行的颗粒度；WBS分解时要识别出所有的交付物、项目管理活动、工程活动等。
- (3) 建立WBS分解中的活动与产品元素的映射关系，识别出WBS中哪些活动可以采用模型法估算；
- (4) 估计产品元素的规模，可以采用代码行法或功能点法，并估计每个产品元素的复杂度、复用率等；
- (5) 根据历史的编码阶段的生产率数据和产品元素的规模估计、复杂度、复用率等采用模型法计算每个产品元素的编码工作量；
- (6) 根据历史的类似项目的数据及估算人的经验估计其他活动的工作量，可以采用经验法。
- (7) 汇总得到：每个阶段的工作量、项目的总工作量。
- 其他说明：
 - 在该场景下，混合使用了经验法与模型法，这2种方法互相补充，而不是互相印证。

- 生产率=规模/工作量
- 不同的类型的项目，生产率是不同的。
- 当定义了生产率的明确含义时，要和将来的实际度量的生产率的定义相一致。
- 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
- 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

生产率定义案例

生产率	计量单位	下限	均值	上限
单元测试用例生产率	个/人月	120	240	360
集成测试用例生产率	个/人月	100	200	300
设计文档生产率	页/人月	80	100	120
编码生产率 (JAVA)	KLOC/人月	1.2	2.0	3.0

- （1）有类似项目的历史数据，但是没有积累历史项目的工作量分布数据
- （2）有类似项目的全生命周期的生产率数据（含管理工作量）
- （3）有详细需求

场景四：由总体印证基于WBS的估计 - 估算步骤



- (1) 产品分解，将系统分为子系统，子系统分解为模块；
- (2) 估计产品元素的规模，可以采用代码行法或功能点法；
- (3) 累计出整个产品的总规模，并估计产品总体的复杂度、复用率等；
- (4) 根据类似项目的全生命周期的生产率数据和产品的总规模、复杂度、复用率等采用模型法计算总的开发工作量；
- (5) WBS分解，将任务分解到一个人或者一个小团队可以执行的颗粒度；WBS分解时要识别出所有的交付物、项目管理活动、工程活动等。
- (6) 根据历史的类似项目的数据及估算人的经验估计所有活动的工作量，可以采用经验法。
- (7) 汇总得到：每个阶段的工作量、项目的总工作量。
- (8) 与第(4)步得出的工作量进行比较印证，如果偏差不大，则以第(7)步的结果为准，如果偏差比较大，要仔细分析原因，可能的原因举例如下：
 - 类似项目的生产率数据不适合本项目；
 - WBS分解的颗粒度不够详细；
 - 估算专家的经验不适合本项目；
 - 具体任务的估计不合理；
 - 针对原因，对估算的结果进行调整，使其趋向合理。
- 其他说明：
 - 在该场景下，对于项目的总工作量有2个结果或者多个结果，这些结果可以互相印证，以发现估算过程中的不合理之处，是估计更加合理。

场景五：三维印证基于WBS的估计

- 场景描述：
 - (1) 有类似项目的历史数据
 - (2) 有类似项目的全生命周期的生产率数据（含管理工作量）
 - (3) 有详细需求
 - (4) 实施了CMMI3级，有历史项目的工作量分布数据（阶段分布、工种分布）

场景五：三维印证基于WBS的估计-估算步骤

- (1) 产品分解，将系统分为子系统，子系统分解为模块；
- (2) 估计产品元素的规模，可以采用代码行法或功能点法；
- (3) 累计出整个产品的总规模，并估计产品总体的复杂度、复用率等；
- (4) 根据类似项目的全生命周期的生产率数据和产品的总规模、复杂度、复用率等采用模型法计算总的开发工作量；
- (5) 根据历史项目的工作量分布数据及第（4）步估算的项目总工作量，计算：
 - 每个阶段的工作量
 - 每个工种的工作量
- (6) WBS分解，将任务分解到一个人或者一个小团队可以执行的颗粒度；WBS分解时要识别出所有的交付物、项目管理活动、工程活动等。
- (7) 根据历史的类似项目的数据及估算人的经验估计所有活动的工作量，可以采用经验法。
- (8) 汇总得到：每个阶段的工作量、每个工种的工作量、项目的总工作量。

场景五：三维印证基于WBS的估计-估算步骤

- (9) 与第(4)、(5)步得出的工作量进行比较印证，如果偏差不大，则以第(7)步的结果为准，如果偏差比较大，要仔细分析原因，可能的原因举例如下：
 - 类似项目的生产率数据不适合本项目；
 - WBS分解的颗粒度不够详细；
 - 估算专家的经验不适合本项目；
 - 具体任务的估计不合理；
 - 针对原因，对估算的结果进行调整，使其趋向合理。
- 其他说明：
 - 在该场景下，对于项目的总工作量有2个结果或者多个结果，并且采用2种方法都得到了每个阶段、每个工种的工作量、项目的总工作量，可以从上述的3个维度对这些结果进行互相印证，以发现估算过程中的不合理之处，是估计更加合理。

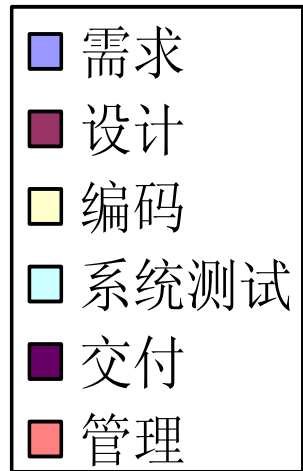
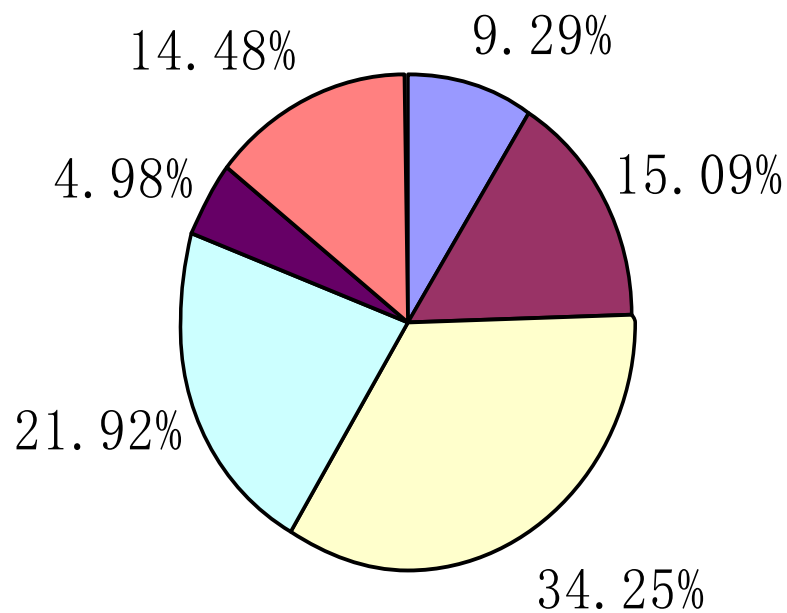
工作量分布模型建立的案例-原始数据

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

工作量分布模型建立的案例-分析后的数据

	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%

工作量分布



工作量分布模型案例

工程类工作量分布：

规模与比例	需求	设计	开发	测试	交付	维护
大型	10.68%	16.82%	41.72%	24.79%	3.19%	2.80%
中型	13.74%	16.47%	38.14%	21.09%	7.46%	3.10%
小型	13.89%	17.57%	28.25%	13.13%	18.82%	8.34%

管理类工作量/工程类工作量

大	10.86%
中	14.95%
小	14.40%

管理类各项工作量比例

项目管理	度量	配置管理	质量保证
65.00%	10.00%	10.00%	15.00%

工作量分布的业内参考数据

规模	需求	架构	构造	系统测试	管理
1KLOC	4%	10%	61%	16%	9%
25KLOC	4%	14%	49%	23%	10%
125KLOC	7%	15%	44%	23%	11%
500KLOC	8%	15%	35%	29%	13%

根据项目类型调整工作量的分配

活动	业务系统、内部 intranet 系统	嵌入式系统、电信、设备驱动程序、系统软件	套装软件、科学系统、工程系统、公众互联网系统
需求	-3%	+20%	-20%
架构	-7%	+10%	-5%
构造	+5%	-10%	+2%
系统测试	-7%	+6%	+9%
管理	+3%	+3%	-15%

- (1) 有类似项目的历史数据
- (2) 有类似项目的编码活动的生产率数据（不含管理工作量）
- (3) 有详细需求
- (4) 实施了CMMI3级，有历史项目的工作量分布数据（阶段分布、工种分布、阶段工种分布）
- (5) 项目采用了瀑布模型

场景六：四维印证基于WBS的估计-估算步骤

- (1) 产品分解，将系统分为子系统，子系统分解为模块；
- (2) 估计产品元素的规模，可以采用代码行法或功能点法，并估计每个产品元素的复杂度、复用率等；
- (3) 根据类似项目的编码活动的生产率数据和产品元素的规模、复杂度、复用率等采用模型法计算每个产品元素的编码工作量；
- (4) 根据历史项目的按工种的工作量分布数据及第(3)步的估算的编码工作量依次计算：
 - i) 根据历史项目的编码的工作量占编码阶段的工作量的百分比与第(3)部计算出的编码工作量计算编码阶段的总工作量；
 - ii) 根据历史项目的编码阶段各工种的工作量分布百分比计算编码阶段每个工种的工作量；
 - iii) 根据历史项目的其他阶段的工作量与编码阶段的工作量比例计算其他阶段的总工作量；
 - iv) 根据历史项目的其他阶段的每个工种的工作量分布百分比及第iii)步的结果计算其他阶段的每个工种的工作量；
- (5) WBS分解，将任务分解到一个人或者一个小团队可以执行的颗粒度；WBS分解时要识别出所有的交付物、项目管理活动、工程活动等。

场景六：四维印证基于WBS的估计-估算步骤

- (6) 根据历史的类似项目的数据及估算人的经验估计所有活动的工作量，可以采用经验法。
- (7) 汇总得到：每个阶段每个工种的工作量、每个阶段的工作量、每个工种的工作量、项目的总工作量。
- (8) 与第(4)步得出的工作量进行比较印证，如果偏差不大，则以第(6)步的结果为准，如果偏差比较大，要仔细分析原因，可能的原因举例如下：
- 类似项目的生产率数据不适合本项目；
 - WBS分解的颗粒度不够详细；
 - 估算专家的经验不适合本项目；
 - 具体任务的估计不合理；针对原因，对估算的结果进行调整，使其趋向合理。
- 其他说明：
 - 在该场景下，对于项目的总工作量有2个结果或者多个结果，并且采用2种方法都得到了每个阶段的工作量、每个工种的工作量、每个阶段每个工种的工作量、项目的总工作量，可以从上述的4个维度对这些结果进行互相印证，以发现估算过程中的不合理之处，是估计更加合理。

场景七：需求变更的工作量估计-场景描述

- 有变更的需求描述
- 项目进行到了编码阶段
- 有本项目的编码的生产率

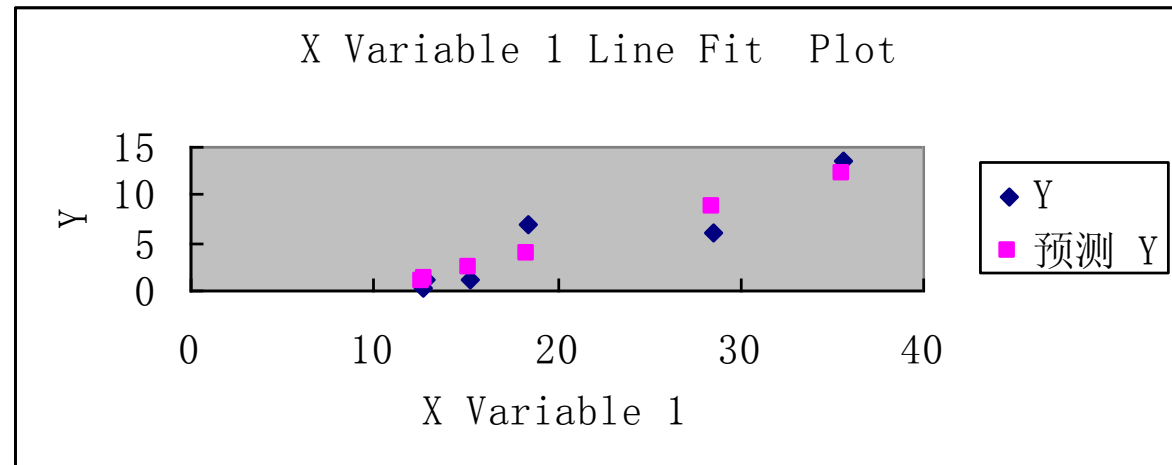
- (1) 进行需求变更的波及范围分析
- (2) 进行本次变更的WBS分解
- (3) 对于变更引起的代码变化进行规模、复杂度等其他属性的估计
- (4) 根据本项目的编码的生产率及估计的规模采用模型法估计工作量
- (5) 对于WBS分解中其他活动进行经验估计
- (6) 汇总所有的工作量得到本次变更的工作量估计

- 在需求变更时需要考虑如下的影响：
 - 技术影响
 - 其他需求
 - 设计
 - 编码
 - 测试用例
 - 交付的文档
 - 管理影响
 - 规模
 - 工作量
 - 工期
 - 成本
 - 质量
 - 计划
 - 风险

需求变更总工作量的预测模型

需求变更工作量 = $0.48803166157189 \times \text{项目计划工作量} - 5.19373148961092$

项目名称	项目工作量 (X1) 人月	需求变更工作量 (Y) 人月
1	35.6	13.42
2	18.48	7.04
3	15.3	1.12
4	28.5	5.93
5	12.88	1.26
6	12.7	0.32



- 讨论：公司的软件估算工作如何做？

Q&A

谢谢!