



Shanghai Jiao Tong University

软件过程与管理

Module: Software Estimation Technologies

软件估算技术

上海交通大学软件工程学院

问题

- ◆ 我是市场人员，如何报价，估算开发时限？
- ◆ 我是项目经理，如何科学地估算进度和人月，作出现实的计划，而不是单凭经验和直觉？
- ◆ 我是客户，如何评估开发商的报价？
- ◆ 世界著名的软件企业是如何估算的？
- ◆ 有没有开发进度的极限？
- ◆ CMM关于软件估算有何要求？

Software Process and Management

2

沈备军

本节内容

- ◆ 软件估算概述
- ◆ 软件规模估算
- ◆ 经验模型
- ◆ 大致估计

@ 9.3,10.3. 教材

@时间管理, 成本管理. PMBOK

Software Process and Management

3

沈备军

为什么要进行估算

- ◆ 进行项目计划
- ◆ 跟踪项目
- ◆ 向客户报价
- ◆ 过程和能力改进

Software Process and Management

4

沈备军

估算内容

- ◆ 规模
- ◆ 工作量
- ◆ 进度
- ◆ 关键计算机资源
- ◆ 成本

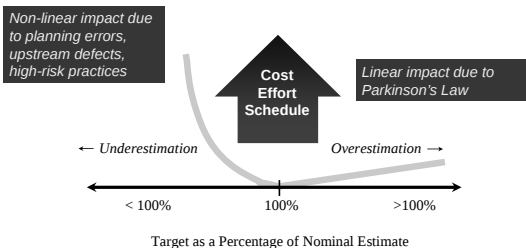
Software Process and Management

5

沈备军

估算精确度带来的影响

对项目进度和费用等的损害程度



Software Process and Management

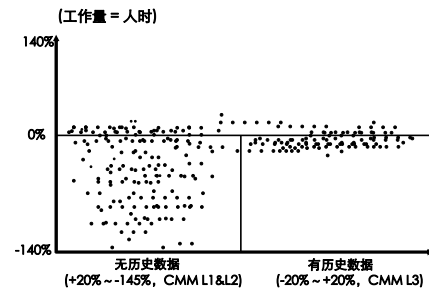
6

沈备军

估算准确性的影响因素

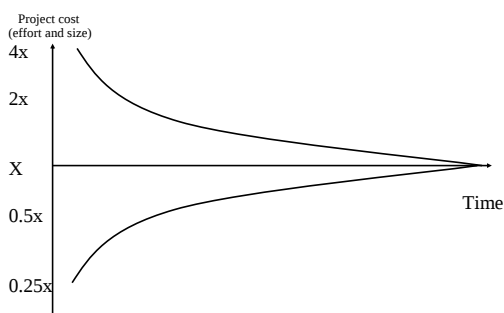
- ◆ 项目复杂性
- ◆ 项目规模
- ◆ 结构不确定性，即需求被确定的程度，功能被分解的容易程度等
- ◆ 历史信息的可用性

历史数据对估算准确度的影响



Ref: Scott Griffin, CIO, Boeing Corp., keynote talk at SEPG'2000.
(Based on 120 projects in Boeing Information Systems)

估算精确性的收敛图



估算收敛举例

- ◆ 初始产品概念阶段 25-400人月 vs. 100人月
- ◆ 已批准的产品概念阶段 50-200人月 vs. 100人月
- ◆ 需求说明书阶段 90-200人月 vs. 135人月
- ◆ 产品设计说明书阶段 145-180人月 vs. 145人月
- ◆ 结束阶段 170人月

注意：不要采用单点估算

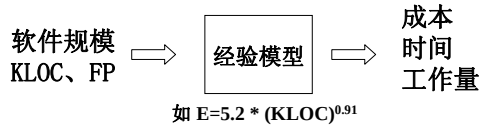
常用的估算方法

- ◆ 工程方法
 - 类似项目直接比较估算工作量和进度
 - 使用生产率数据，根据规模估算工作量
 - 使用经验模型 $d = f(V_i)$ ，根据规模估算工作量和进度
- ◆ 非工程方法
 - 专家的判断
 - Parkinson 法则
 - 从价格出发

使用生产率数据

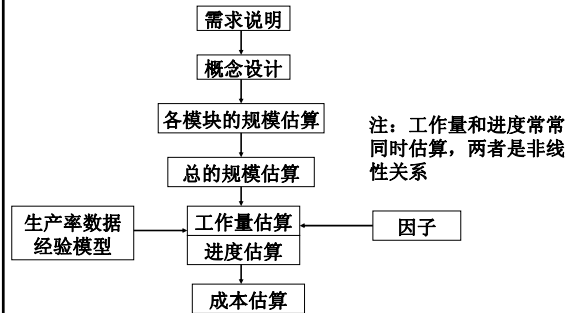
- ◆ 工作量估算值 = 项目规模 / 生产率
 - 举例：10000行 / (100行/人天) = 100人天
- ◆ 生产率不能简单地比较
 - 规模的估算方法会不同
 - 编程语言
 - 应用领域
 - 过程
 - 资源
 - 工程师
 - 质量

使用经验模型



- 使用经验模型是目前最佳的估算方法
- 经验模型的经验数据是从有限的一些项目样本中得到的
 - IBM 经验模型
 - Putnam 经验模型
 - COCOMO 经验模型

常用的估算步骤



软件项目成本

- 直接材料成本
 - 项目外购的直接用于项目并将最终交付给用户的硬件、网络、第三方软件和服务等
- 直接人力资源成本
 - 工作量（软件项目的主要成本）
- 项目的实施费用成本
 - 差旅费、交通费、通信费、出差补贴等
- 其他直接成本
 - 设备和场地的租借、项目组专用设备的折旧费、合同的税费、项目的销售和广告费等
- 间接成本（分摊成本）
 - 办公场地的租金、公司管理费用、公司整体运作的市场和广告费、水电费、办公通信费等

项目经理应该关注的是项目的可控成本

估算举例（需求说明）

- 开发一个软件工程网站：
 - 栏目资源：需求、设计、测试、过程和工具
 - 讨论组
 - 网上咨询
 - 资料上传
 - 搜索引擎
 - 会员管理
 - 采用微软平台

估算举例（概念设计）

- 语言：ASP + HTML
- 数据库：SQL Server
- 模块：
 - 需求栏目 设计栏目
 - 测试栏目 工具栏目
 - 讨论组 网上咨询
 - 搜索引擎 会员管理

估算举例（规模估算）

基于类似的两个项目，做出以下估算

产品	估算单位	规模
需求规范说明书	SRS 页	10
概要设计文档	HLD 页	15
需求栏目	DLD 页	10
	LOC	150
设计栏目	DLD 页	11
	LOC	120
网上咨询	DLD 页	14
	LOC	250
搜索引擎	DLD 页	10
	LOC	450
...		

合计

Plan 页	10
SRS 页	10
HLD 页	15
DLD 页	120
LOC	2000
Test 页	40

软件估算技术

沈备军

估算举例（工作量估算）

任务	规模	生产率	工作量	校正因子
编写Plan	10页	1页/ph	10 PH	生产率校正 新手: 50% 普通: 100% 高手: 150%
评审Plan	10页	2页/ph	5 PH	
编写SRS	10页	0.2页/ph	50 PH	
评审SRS	10页	1页/ph	10 PH	
编写HLD	15页	1页/ph	15 PH	
评审HLD	15页	2页/ph	7.5 PH	
编写DLD	120页	1页/ph	120 PH	
评审DLD	120页	2页/ph	60 PH	
编程	2000行	20行/PH	100 PH	
代码评审	2000行	20行/PH	100 PH	
Test	40	5行/PH	8 PH	
合计			485.5 PH	

Software Process and Management

19

沈备军

估算举例（进度估算）

二个工程师负责做此项目

周数	日期	工作小时数	累积工作小时数	获得值 Earned Value
1	9/1	80	80	16
2	9/8	80	160	32
3	9/15	100	260	53
4	9/22	80	340	70
5	9/29	80	420	86
6	10/5	65.5	485.5	100

Software Process and Management

20

沈备军

估算技巧

- ◆ 分解估算（过程分解和模块分解）
 - 和的误差大于误差的和
- ◆ 给出估计一个范围（如最好情况下）或一个可信赖程度（如90%的把握）。
 - 期望值 = (乐观值 + 4 * 一般值 + 悲观值) / 6
- ◆ 避免无准备的估算
 - 留出估算的时间
- ◆ 使用以前项目的数据

Software Process and Management

21

沈备军

估算技巧（续）

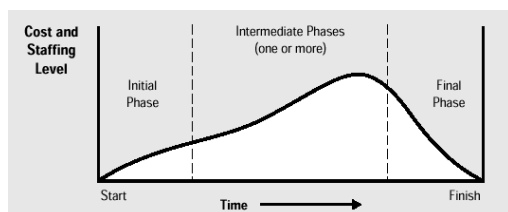
- ◆ 估算师和开发人员共同估算
- ◆ 走查估算
- ◆ 分类法估算
- ◆ 不要忽略普通任务
- ◆ 使用软件估算工具
- ◆ 使用几种不同估算方法，并比较它们的结果
- ◆ 项目进行中进行改变估算规则

Software Process and Management

22

沈备军

过程分解技术



人力和费用分布图

Software Process and Management

23

沈备军

软件开发工作量分布

- ◆ 计划 2 — 3 %
- ◆ 分析 10 — 25 %
- ◆ 设计 20 — 25 %
- ◆ 编码 15 — 20 %
- ◆ 测试 30 — 40 %

Software Process and Management

24

沈备军

讨论：项目估算

- ◆ 假定有一个6个月的进度计划，你计划4周达到第一个里程碑，而实际花了5周，当你错过进度日期时，如何修正进度：
 - 在后续进度中弥补损失的一周
 - 把这一周加到整个进度中
 - 把整个进度乘以拖延的数量，本例应该是乘以25%

本节内容

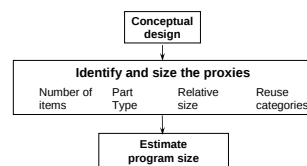
- ◆ 软件估算概述
- ✧◆ 软件规模估算
- ◆ 经验模型
- ◆ 大致估计

规模估算

- ◆ LOC 代码行
- ◆ FP 功能点
- ◆ 需求的数量
- ◆ 文档的页数

LOC 估算—PROBE方法

- ◆ PROBE (PROxy Based Estimating)



Example: Building Costs

- ◆ Problem
 - The builder needs to know the floor area (in sq. ft.) to estimate the cost of construction.
 - Clients normally cannot describe their needs in square feet.
 - They usually can describe the type and number of rooms they want.
- ◆ Solution
 - Use rooms as a proxy for size.
 - Use historical (typical) data to translate from rooms to square feet.

Example: Customer Requirements

- ◆ Bedrooms: 1 large, 2 medium, and 1 small
- ◆ Bathrooms: 1 large and 2 medium
- ◆ Kitchen: 1 medium
- ◆ Living room: 1 large
- ◆ Dining room: 1 medium
- ◆ Family room: 1 large
- ◆ Utility: 1 medium

Historical Building Data

	Small	Medium	Large
Bedrooms	90	140	200
Bathrooms	25	60	120
Kitchens	100	130	160
Living rooms	150	250	400
Dining rooms	100	140	200
Family rooms	150	240	340
Utility	25	50	80

Software Process and Management

32

沈备军

Proxy Calculation

Bedroom	1 large	=	1 x 200	=	200
Bedroom	2 medium	=	2 x 140	=	280
Bedroom	1 small	=	1 x 90	=	90
Bathroom	1 large	=	1 x 120	=	120
Bathroom	2 medium	=	2 x 60	=	120
Kitchen	1 medium	=	1 x 130	=	130
Living room	1 large	=	1 x 400	=	400
Dining room	1 medium	=	1 x 140	=	140
Family room	1 large	=	1 x 340	=	340
Utility	1 medium	=	1 x 50	=	50
Total (sq. ft.) =					1870

Software Process and Management

32

沈备军

估算整个规模

- ◆ PROBE uses historical data, linear regression, and the prediction interval to produce estimates of known accuracy.
- ◆ E is the estimated proxy size.

Software Process and Management

33

沈备军

Proxy示例

- ◆ Classes, functions, and procedures
- ◆ Product elements
 - database elements
 - screens, reports, scripts, files
 - book chapters

Software Process and Management

34

沈备军

Example C++ Class Size Ranges

Type	LOC per item				
	VS	S	M	L	VL
Calculation	2.34	5.13	11.25	24.66	54.04
Data	2.60	4.79	8.84	16.31	30.09
I/O	9.01	12.06	16.15	21.62	28.93
Logic	7.55	10.98	15.98	23.25	33.83
Set-up	3.88	5.04	6.56	8.53	11.09
Text	3.75	8.00	17.07	36.41	77.66

Software Process and Management

35

沈备军

FP计算

信息域参数	计数	加 权 因 数			加权计数		
		简单	中间	复杂			
用户输入数		×	3	4	6	=	
用户输出数		×	4	5	7	=	
用户查询数		×	3	4	6	=	
文 件 数		×	7	10	15	=	
外部接口数		×	5	7	10	=	
总 计 数							

Software Process and Management

36

沈备军

计算FP

$$FP = \text{总计数} \times (0.65 + 0.01 \times \text{SUM}(Fi))$$

- ◆ 总计数是所有加权计数项的和
- ◆ Fi ($i = 1..14$) 是复杂性校正值,
 Fi 的取值0..5:

0 没有影响	1 偶然的
2 适中的	3 普通的
4 重要的	5 极重要的

计算复杂性校正值Fi

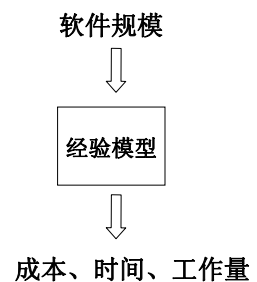
1. 系统是否需要可靠的备份和恢复?
2. 是否需要数据通信?
3. 是否有分布处理的功能?
4. 是否性能成为关键?
5. 系统是否运行在既存的高度实用化的操作环境中?
6. 系统是否需要联机数据项?
7. 联机数据项是否需要建立多重窗口显示和操作, 以处理输入处理?
8. 主文件是否联机更新?
9. 输入、输出、文件、查询是否复杂?
10. 内部处理过程是否复杂?
11. 程序代码是否可复用?
12. 设计中是否包括了转移和安装?
13. 系统是否设计成可以重复安装在不同机构中?
14. 系统是否设计成易修改和易使用?

本节内容

- ◆ 软件估算概述
- ◆ 软件规模估算
- ◆ 经验模型
- ◆ 大致估计

经验模型

- ◆ IBM估算模型
- ◆ Putnam SLIM
- ◆ COCOMO
- ◆ ESTIMACS
- ◆ FAST
- ◆ MAINSTAY
- ◆ PRICE
- ◆ SPQR 等



IBM经验模型

$$E = 5.2 \times L^{0.91}$$

$$D = 4.1 \times L^{0.36} = 14.47 \times E^{0.35}$$

$$S = 0.54 \times E^{0.6}$$

$$DOC = 49 \times L^{1.01}$$

- ◆ L 是源代码行数 (KLOC), E 是工作量 (PM), D 是项目持续时间(月), S 是人员需要量 (人), DOC 是文档数量 (页)。

COCOMO 经验模型

- ◆ CONstructive COst Model
- ◆ 结构型成本估算模型是一种精确、易于使用的成本估算方法。
- ◆ 1981年基于63个项目的分析基础上提出第一个版本 COCOMO'81
- ◆ 最新版本COCOMO II

COCOMO'81

- ◆ COCOMO'81按其详细程度分成三级：
 - 基本COCOMO模型
 - 中间COCOMO模型
 - 详细COCOMO模型

COCOMO 模型的术语

- ◆ DSI（源指令条数）定义为代码的源程序行数。若一行有两个语句，则算做一条指令。它包括作业控制语句和格式语句，但不包括注释语句。
 $KDSI = 1000DSI$ 。
 MM（度量单位为人月）表示开发工作量。
 TDEV（度量单位为月）表示开发进度。
- ◆ 软件开发项目的分类
 - 组织型
 - 半独立型
 - 嵌入型

基本COCOMO模型

- ◆ 基本COCOMO模型是静态单变量模型，用源代码行数(LOC) 为自变量的经验函数计算软件开发工作量

总体类型	工 作 量	进 度
组织型	MM = 2.4 (KDSI) ^{1.05}	TDEV = 2.5 (MM) ^{0.38}
半独立型	MM = 3.0 (KDSI) ^{1.12}	TDEV = 2.5 (MM) ^{0.35}
嵌入型	MM = 3.6 (KDSI) ^{1.20}	TDEV = 2.5 (MM) ^{0.32}

中间COCOMO模型

- ◆ 中间COCOMO模型在用LOC为自变量的函数计算软件开发工作量（称为名义工作量）的基础上，用涉及产品、硬件、人员、项目等方面的15种影响因素调整工作量估算。

$$MM = r \cdot (KDSI)^c \cdot \prod_{i=1}^{15} f_i$$

中间COCOMO模型的名义工作量公式

总体类型	工 作 量	进 度
组织型	MM = 3.2 (KDSI) ^{1.05}	TDEV = 2.5 (MM) ^{0.38}
半独立型	MM = 3.0 (KDSI) ^{1.12}	TDEV = 2.5 (MM) ^{0.35}
嵌入型	MM = 2.8 (KDSI) ^{1.20}	TDEV = 2.5 (MM) ^{0.32}

工作量因素 f_i		非常低	低	正常	高	非常高	超高
产 品 因 素	软件可靠性	0.75	0.88	1.00	1.15	1.40	
	数据库规模		0.94	1.00	1.08	1.16	
	产品复杂性	0.70	0.85	1.00	1.15	1.30	1.65
计 算 机 因 素	执行时间限制			1.00	1.11	1.30	1.66
	存储限制			1.00	1.06	1.21	1.56
	虚拟机易变性		0.87	1.00	1.15	1.30	
	环境周转时间		0.87	1.00	1.07	1.15	
人 员 的 因 素	分析员能力		1.46	1.00	0.86	0.71	
	应用领域实际经验	1.29	1.13	1.00	0.91	0.82	
	程序员能力	1.42	1.17	1.00	0.86	0.70	
	虚拟机使用经验	1.21	1.10	1.00	0.90		
	程序语言使用经验	1.41	1.07	1.00	0.95		
项 目 因 素	现代程序设计技术	1.24	1.10	1.00	0.91	0.82	
	软件工具的使用	1.24	1.10	1.00	0.91	0.83	
	开发进度限制	1.23	1.08	1.00	1.04	1.10	

这里所谓虚拟机是指为完成某一个软件任务所使用硬、软件的结合。

软件估算技术

沈备军

中间COCOMO模型举例

- ◆ 一个规模为10KDSI的商用微机远程通信的嵌入型软件，使用中间COCOMO模型进行成本估算。
- ◆ 程序名义工作量
 $MM = 2.8 (10)^{1.20} = 44.38 \text{ (MM)}$
- ◆ 程序实际工作量
 $MM = 44.38 \times$
 $= 44.38 \times 1.17 = 51.5 \text{ (MM)}$

Software Process and Management

49

沈备军

影响工作量因素 f_i	情 况	取 值
1 软件可靠性	只用于局部地区，恢复问题不严重	1.00 (正常)
2 数据库规模	20000字节	0.94 (低)
3 产品复杂性	用于远程通信处理	1.30 (很高)
4 时间限制	使用70%的CPU时间	1.10 (高)
5 存储限制	64K中使用45K	1.06 (高)
6 机器	使用商用微处理机	1.00 (额定值)
7 周转时间	平均2小时	1.00 (额定值)
8 分析员能力	优秀人才	0.86 (高)
9 工作经验	远程通信工作3年	1.10 (低)
10 程序员能力	优秀人才	0.86 (高)
11 工作经验	微型机工作6个月	1.00 (正常)
12 语言使用经验	12个月	1.00 (正常)
13 使用现代程序设计技术	1年以上	0.91 (高)
14 使用软件工具	基本的微型机软件	1.10 (低)
15 工期	9个月	1.00 (正常)

Software Process and Management

50

沈备军

中间COCOMO模型举例

- ◆ 开发所用时间
 $TDEV = 2.5 (51.5)^{0.32} = 8.9 \text{ (月)}$
- ◆ 如果分析员与程序员的工资都按每月6,000美元计算，则该项目的开发人员的工资总额为
 $51.5 \times 6,000 = 309,000 \text{ (美元)}$
- ◆ 做为对比，现在用IBM模型计算：
 $PM = 5.2 (10)^{0.91} = 42.27 \text{ (人月)}$
 $D = 4.1 (10)^{0.38} = 9.84 \text{ (月)}$
 $S = 0.54 (42.27)^{0.60} = 5.1 \text{ (人)}$

Software Process and Management

51

沈备军

详细COCOMO模型

- ◆ 详细COCOMO模型包括中间COCOMO模型的所有特性，但用上述各种影响因素调整工作量估算时，还要考虑对软件工程过程中每一步骤（分析、设计等）的影响。
- ◆ 例如，关于软件可靠性（RELY）要求的工作量因素分级表（子系统层），如表所示。

Software Process and Management

52

沈备军

软件可靠性工作量因素分级表(子系统层)

阶段 RELY级别	需求和 产品设计	详细 设计	编程及 单元测试	集成及 测试	综合
非常低	0.80	0.80	0.80	0.60	0.75
低	0.90	0.90	0.90	0.80	0.88
正常	1.00	1.00	1.00	1.00	1.00
高	1.10	1.10	1.10	1.30	1.15
非常高	1.30	1.30	1.30	1.70	1.40

Software Process and Management

53

沈备军

COCOMO II

- ◆ 考虑到需求的不稳定性
- ◆ 有三种模型：
 - 应用组成模型
 - 早期设计模型
 - 结构后期模型

Software Process and Management

54

沈备军

COCOMO II 17个影响因子 Fi

- ◆ 产品因素
 - 软件可靠性
 - 数据库规模
 - 产品复杂性
 - 文档
 - 产品复用性
- ◆ 计算机因素
 - 执行时间限制
 - 存储限制
 - 平台稳定性
- ◆ 人员因素
 - 分析员能力
 - 应用领域经验
 - 程序员能力
 - 平台经验
 - 语言使用经验
 - 人员持续性
- ◆ 项目因素
 - 使用软件工具
 - 多地点开发
 - 工期

COCOMO II 工具演示

正确使用模型

- ◆ 了解各种模型的原理和优缺点，根据本企业的具体情况来使用模型
- ◆ 采用您自己的历史数据调整模型，减少偏差
- ◆ 使用多种估算方法
- ◆ 可借助工具

本节内容

- ◆ 软件估算概述
- ◆ 软件规模估算
- ◆ 经验模型
- ◆ 大致估计

大致估算

- ◆ When
 - 当没有历史数据时
 - 当需要简单的估算方法时
- ◆ Why
 - 比直觉更精确
- ◆ How
 - 可能的最短进度
 - 有效进度
 - 普通进度

可能的最短进度 (1)

假定:

- ◆ 最拔尖的开发者的 (前10%)
- ◆ 理想的管理
- ◆ 先进的工具支持, 理想的办公环境
- ◆ 最具时效的开发方法和开发工具
- ◆ 尽可能地压缩进度, 直至不可能再进一步压缩

可能的最短进度 (2)

系统规模 LOC	系统软件		商业软件		封装商品软件	
	进度(月)	工作量(人月)	进度(月)	工作量(人月)	进度(月)	工作量(人月)
10,000	6	25	3.5	5	4.2	8
15,000	7	40	4.1	8	4.9	13
20,000	8	57	4.6	11	5.6	19
25,000	9	74	5.1	15	6	24
30,000	9	110	5.5	22	7	37
35,000	10	130	5.8	26	7	44
40,000	11	170	6	34	7	57
45,000	11	195	6	39	8	66
50,000	11	230	7	46	8	79
60,000	12	285	7	57	9	98

可能的最短进度 (3)

系统规模 LOC	系统软件		商业软件		封装商品软件	
	进度(月)	工作量(人月)	进度(月)	工作量(人月)	进度(月)	工作量(人月)
70,000	13	350	8	71	9	120
80,000	14	410	8	83	10	140
90,000	14	480	9	96	10	170
100,000	15	540	9	110	11	190
120,000	16	680	10	140	11	240
140,000	17	820	10	160	12	280
160,000	18	960	10	190	13	335
180,000	19	1,100	11	220	13	390
200,000	20	1250	11	250	14	440
250,000	22	1650	13	330	15	580
300,000	24	2100	14	420	16	725
400,000	27	2900	15	590	19	1,000
500,000	30	3900	17	780	20	1,400

Software Process and Management

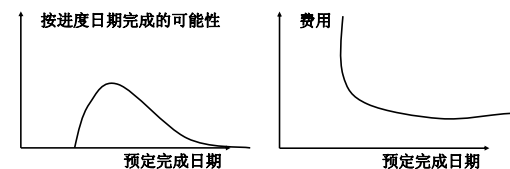
61

沈备军

可能的最短进度 (4)

两个事实:

- ◆ 存在一个可能的最短进度, 而且不可能突破它
- ◆ 当把进度缩短得比普通进度短时, 费用将迅速上涨



Software Process and Management

62

沈备军

有效的进度 (1)

假定:

- ◆ 优秀的开发者 (前25%), 在编程语言和环境上有一年的工作经验
- ◆ 团队对项目有共同的想法, 没有严重的冲突
- ◆ 有效使用编程工具
- ◆ 使用现代编程思想, 主动的风险管理
- ◆ 采用快速开发实践, 使用沟通工具, 优良的工作环境

Software Process and Management

63

沈备军

有效的进度 (2)

系统规模 LOC	系统软件		商业软件		封装商品软件	
	进度(月)	工作量(人月)	进度(月)	工作量(人月)	进度(月)	工作量(人月)
10,000	8	24	4.9	5	5.9	8
15,000	10	38	5.8	8	7	12
20,000	11	54	7	11	8	18
25,000	12	70	7	14	9	23
30,000	13	97	8	20	9	32
35,000	14	120	8	24	10	39
40,000	15	140	9	30	10	49
45,000	16	170	9	34	11	57
50,000	16	190	10	40	11	67
60,000	18	240	10	49	12	83

Software Process and Management

64

沈备军

有效的进度 (3)

系统规模 LOC	系统软件		商业软件		封装商品软件	
	进度(月)	工作量(人月)	进度(月)	工作量(人月)	进度(月)	工作量(人月)
70,000	19	290	11	61	13	100
80,000	20	345	12	71	14	120
90,000	21	400	12	82	15	140
100,000	22	450	13	93	15	160
120,000	23	560	14	115	16	195
140,000	25	670	15	140	17	235
160,000	26	799	15	160	18	280
180,000	28	910	16	190	19	320
200,000	29	1300	17	210	20	360
250,000	32	1300	19	280	22	470
300,000	34	1650	20	345	24	590
400,000	38	2350	22	490	27	830
500,000	42	3100	25	640	29	1100

Software Process and Management

65

沈备军

普通进度 (1)

假定:

- ◆ 中等的开发者, 熟悉编程语言和环境
- ◆ 在解决冲突上有一定经验
- ◆ 一定程度上使用编程工具和现代编程思想
- ◆ 工作环境有些不理想, 但足够了

☺ 对一个一般的项目达到普通进度有50%的把握

Software Process and Management

66

沈备军

软件估算技术

普通进度（2）

系统规模 LOC	系统软件		商业软件		封装商品软件	
	进度(月)	工作量(人月)	进度(月)	工作量(人月)	进度(月)	工作量(人月)
10,000	10	48	6	9	7	15
15,000	12	76	7	15	8	24
20,000	14	110	8	21	9	34
25,000	15	140	9	27	10	44
30,000	16	185	9	37	11	59
35,000	17	220	10	44	12	71
40,000	18	270	10	54	13	88
45,000	19	310	11	61	13	100
50,000	20	360	11	71	14	115
60,000	21	440	12	88	15	145

Software Process and Management

67

沈备军

沈备军

普通进度（3）

系统规模 LOC	系统软件		商业软件		封装商品软件	
	进度(月)	工作量(人月)	进度(月)	工作量(人月)	进度(月)	工作量(人月)
70,000	23	540	13	105	16	175
80,000	24	630	14	125	17	210
90,000	25	730	15	140	17	240
100,000	26	820	15	160	18	270
120,000	28	1000	16	200	20	335
140,000	30	1200	17	240	21	400
160,000	32	1400	18	280	22	470
180,000	34	1600	19	330	23	540
200,000	35	1900	20	370	24	610
250,000	38	2400	22	480	26	800
300,000	41	3000	24	600	29	1000
400,000	47	4200	27	840	32	1400
500,000	51	5500	29	1100	35	1800

Software Process and Management

68

沈备军

案例讨论：项目估算

- ◆ 凭直觉的项目估算
- ◆ 仔细的项目估算

Software Process and Management

69

沈备军