

# 软件项目估算\*

王礼强

(东南大学工业发展与培训中心 南京 210096)

**摘要** 软件项目估算是软件项目管理的一个重要组成部分. 本文所介绍的估算法是作者在国外软件公司工作期间的工作总结和归纳.

**关键词** 方法学, 软件项目估算, 软件项目管理, COCOMO 模型, 软件工程.

## 1 估算内容

软件项目就是在一定环境下开发一套能完成一定功能的软件. 软件项目管理就是对软件项目进行估算、组织、计划和跟踪. 软件项目估算包含 3 个方面: 分析和估计项目难度和危险性; 估算项目的总工期和总工作量; 估算项目工期和工作量的分布. 估算在项目管理中的位置如图 1 表示.

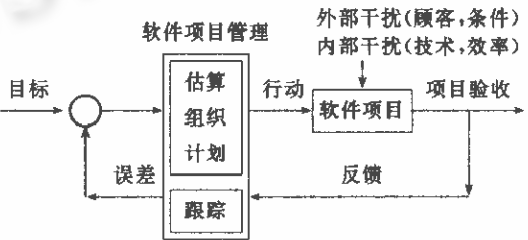


图1

## 2 估算程序

### 2.1 估算难度和危险性

软件项目估算的难度主要来源于技术世界. 首先要从技术世界来分析项目实现的可能性, 一定要明确技术难点; 其次要认真分析工作条件、工作环境、人员素质和资金储备; 最后还要考虑项目负责人与顾客的关系和自身管理能力. 同时, 要承认 $\pm 10\%$ 的估算差.

### 2.2 明确项目类别

软件项目类别就是指所需开发软件的类型. 通常分为 3 种类型:

- (1) 逻辑型: 如控制器软件、数理模型软件以及语言翻译器软件.
- (2) 综合型: 如过程控制软件.
- (3) 超大型: 如网络软件.

### 2.3 估算软件语句数

通常采用分割经验法来估算软件语句数. 首先把软件分割成不同单元, 再根据以往经验

\* 本文研究得到国家教委回国博士基金资助. 作者王礼强, 1964 年生, 博士, 主要研究领域为电机, 实时控制, 项目管理, 系统工程.

本文通讯联系人: 王礼强, 南京 210096, 东南大学工业发展与培训中心

本文 1995-10-30 收到修改稿

估算每个单元语句数. 累计各个单元语句数, 从而获得整个软件语句数. 不过, 科学的估算法还是 PEARSON 法, 也就是: (1) 估计最有可能的语句数  $Q$ ; (2) 估计可能最少语句数  $O$ , 几率小于 1%; (3) 估计可能最多语句数  $P$ , 几率小于 1%.

$$l = \frac{O + P + 4Q}{6} \tag{1}$$

图 2 是 PEARSON 几率分部图.

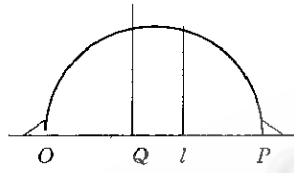


图2

假如只要通过修改以往软件就能完成开发任务的话, 那么应首先估算修改语句数, 再根据下列公式求得相应开发语句数.

$$l = lm * (0.4\% * cm + 0.3\% * pm + 0.3\% * im) \tag{1a}$$

其中  $l$  为等价需要开发的语句数,  $lm$  为需要修改的语句数,  $cm$  为需要修改的语句数中设计部分所占百分比,  $pm$  为需要修改的语句数中编程部分所占比例,  $im$  为需要修改的语句数中组装集成部分所占比例.

### 2.4 估算工作量及工期

本文采用的估算模型叫 COCOMO 模型, 它通过 4 个常量和 2 个公式来描述软件语句数和项目工作量及工期之间的关系.

$$w = A * L^B \tag{2}$$

$$t = C * (w)^D \tag{3}$$

其中  $A, B, C, D$  是常量, 取决于项目类别和开发条件. 公式(2)表达了工作量( $w$ )和语句数( $l$ )之间关系, 公式(3)表达了工期( $t$ )和工作量( $w$ )之间关系.

由公式(2)和(3)可以得到公式(4), 直接表达项目工期( $t$ )和软件语句数( $l$ )之间关系.

$$t = C * (A * l^B)^D \tag{4}$$

一般来说, 对于逻辑型项目  $A = 2.4, B = 1.05, C = 2.5, D = 0.38$ ; 即

$$w = 2.4 * l^{1.05} \tag{2a}$$

$$t = 2.5 * (w)^{0.38} \tag{3a}$$

对于综合型  $A = 3.0, B = 1.12, C = 2.5, D = 0.35$ ; 即

$$w = 3.0 * l^{1.12} \tag{2b}$$

$$t = 2.5 * (w)^{0.35} \tag{3b}$$

对于超大型  $A = 3.6, B = 1.20, C = 2.5, D = 0.32$ , 即

$$w = 3.6 * l^{1.20} \tag{2c}$$

$$t = 2.5 * (w)^{0.32} \tag{3c}$$

其中  $l$  的单位为“千条指令”, 也就是 1 000 条执行语句(不含说明语句),  $w$  的单位为“人月”, 也就是 1 个人在 1 个月中的工作量,  $t$  的单位是月.

当然, 我们知道任何模型都有它们的假设条件和局限性. COCOMO 模型也不例外, 其

假设条件是:软件开发使用行之有效的软件工程方法学;软件项目包括软件开发整个过程:从项目说明到项目验收;工作量单位是“人月”,相当于 152 工作小时。

在实际使用中,根据具体情况,可对以上计算的工作量进行修正,乘以修正系数。下面请看几个影响因素。

#### · 产品

(1) 可靠性要求修正系数  $k_1$ 。

通常  $k_1=1$ ,但如果运行故障会导致经济损失和人身安全,那么就要对软件进行重复测试和严格检查,这时  $k_1=1.29$ 。

(2) 数据库大小修正系数  $k_2$ 。

通常  $k_2=1$ ,但如果数据库语句数 100 倍地大于程序语句数时,这时  $k_2=1.08$ 。

(3) 软件复杂程序修正系数  $k_3$ 。

通常  $k_3=1$ ,但如果软件中经常采用循环机构子程序以及通信中断,这时  $k_3=1.45$ 。

#### · 计算机

(4) 运算速度制约修正系数  $k_4$ 。

通常  $k_4=1$ ,但如果运行速度太慢明显影响了开发人员的效率,这时  $k_4=1.22$ 。

(5) 内存制约修正系数  $k_5$ 。

通常  $k_5=1$ ,但如果运行程序明显受到机器内存限制时,那么  $k_5=1.19$ 。

(6) 执行机器的稳定性修正系数  $k_6$ 。

通常  $k_6=1$ ,但如果运行机器同时又作服务器时,那么  $k_6=1.16$ 。

(7) 开发计算机的反应速度修正系数  $k_7$ 。

通常  $k_7=1$ ,但如果编辑器的回答时间大于 2s,或者 100 条指令的编译时间大于 10s 时,  
 $k_7=1.11$ 。

#### · 人员

(8) 对涉及领域的经验修正系数  $k_8$ 。

通常  $k_8=1$ ,但如果涉及经验不足 3 年时, $k_8=1.19$ 。

(9) 对最终运行机器的熟悉程度修正系数  $k_9$ 。

通常  $k_9=1$ ,但如果涉及经验不足 1 年时, $k_9=1.11$ 。

(10) 对所用语言的熟悉程度修正系数  $k_{10}$ 。

通常  $k_{10}=1$ ,但如果涉及经验不足 1 年时, $k_{10}=1.07$ 。

#### · 管理

(11) 整个队伍的积极性修正系数  $k_{11}$ 。

通常  $k_{11}=1$ ,但如果队伍缺乏交流和失去实现目标信心时, $k_{11}=2.06$ 。

(12) 现代方法的使用修正系数  $k_{12}$ 。

通常  $k_{12}=1$ ,但如果未用或错用方法时, $k_{12}=1.17$ 。

(13) 各种工具的使用修正系数  $k_{13}$ 。

通常  $k_{13}=1$ ,但如果缺乏下列工具之一时, $k_{13}=1.16$ 。如配置管理、项目管理、设计语言、交叉编译器。

应用以上修正系数,可得如下公式:

$$w' = w \cdot (k_1 \cdot k_2 \cdots k_{12} \cdot k_{13}) \tag{5}$$

其中  $w$  是用公式(2)计算的工作量,  $w'$  是乘以修正系数后的工作量.

### 2.5 估算工作量和工期的分布

软件项目开发过程包含 7 个阶段: 项目说明、总体设计、单元设计、编译调试、单元测试、软件集成、项目验收. 各阶段之间关系可用如图 3 的 V 形图来描述.

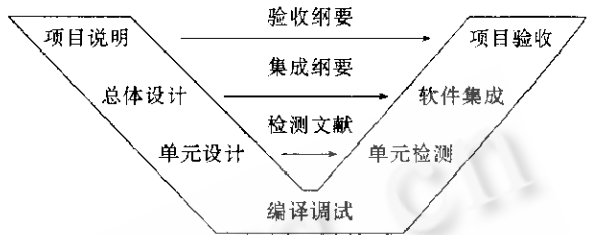


图3

项目说明是理解顾客需求,明确软件功能、开发环境和外界接口;总体设计是确定软件结构,明确软件单元;单元设计是确定子程序和有关常数、变量和通信协议;编译调试是指编程、翻译和子程序调试;单元检测是检查和测试软件单元模型;软件集成是把 2 个和 2 个以上的软件单元进行组装;项目验收是根据项目说明验收集成后的软件. 通过统计和实践,发现工作量和工期的分布可用下列表格表示.

表1 各阶段工作量分布表

		项目语句数	工作量分布				
			2k	8k	32k	128k	512k
项目类别	逻辑型	项目说明(P <sub>w1</sub> )	6	6	6	6	
		总体设计(P <sub>w2</sub> )	14	14	14	14	
		单元处理(P <sub>w3</sub> )	62	59	56	53	
		设计	24	23	22	21	
		编译	28	26	24	22	
	测试	10	10	10	10		
	组装集成(P <sub>w4</sub> )	14	17	20	23		
	项目验收(P <sub>w5</sub> )	4	4	4	4		
	综合型	项目说明(P <sub>w1</sub> )	7	7	7	7	7
		总体设计(P <sub>w2</sub> )	15	15	15	15	15
		单元处理(P <sub>w3</sub> )	57	55	53	52	49
		设计	23	23	21	21	20
		编译	25	23	23	21	20
	测试	9	9	9	10	9	
	组装集成(P <sub>w4</sub> )	16	18	20	22	24	
项目验收(P <sub>w5</sub> )	5	5	5	5	5		
超大型	项目说明(P <sub>w1</sub> )	8	8	8	8	8	
	总体设计(P <sub>w2</sub> )	16	16	16	16	16	
	单元处理(P <sub>w3</sub> )	50	48	46	44	42	
	设计	20	19	18	17	16	
	编译	22	21	20	19	18	
测试	8	8	8	8	8		
组装集成(P <sub>w4</sub> )	20	22	24	26	28		
项目验收(P <sub>w5</sub> )	6	6	6	6	6		

表2 各阶段工期分布表

		项目语句数	工期分布				
			2k	8k	32k	128k	512k
项目类别	逻辑型	项目说明(P <sub>t1</sub> )	9	9	10	11	
		总体设计(P <sub>t2</sub> )	17	16	16	16	
		单元处理(P <sub>t3</sub> )	53	51	47	43	
		设计	20	20	18	16	
		编译	26	24	22	20	
	测试	7	7	7	7		
	组装集成(P <sub>t4</sub> )	16	19	22	25		
	项目验收(P <sub>t5</sub> )	5	5	5	5		
	综合型	项目说明(P <sub>t1</sub> )	13	15	16	17	18
		总体设计(P <sub>t2</sub> )	19	20	21	21	22
		单元处理(P <sub>t3</sub> )	46	41	37	34	30
		设计	17	15	13	12	10
		编译	22	19	17	15	13
	测试	7	7	7	7	7	
	组装集成(P <sub>t4</sub> )	16	18	20	22	24	
项目验收(P <sub>t5</sub> )	6	6	6	6	6		
超大型	项目说明(P <sub>t1</sub> )	18	20	23	25	27	
	总体设计(P <sub>t2</sub> )	23	24	25	25	26	
	单元处理(P <sub>t3</sub> )	36	31	27	23	20	
	设计	13	11	9	7	5	
	编译	16	13	11	9	8	
测试	7	7	7	7	7		
组装集成(P <sub>t4</sub> )	16	18	18	20	20		
项目验收(P <sub>t5</sub> )	7	7	7	7	7		

### 2.6 估算软件开发成本

假设各阶段单位工作量价格表如表 3. 那么,在不超工期情况下,根据项目类型、项目语句数和估算的工作量,结合表 1 和表 3,可得:

$$s = w \cdot (P_{w1} \cdot S_{w1} + P_{w2} \cdot S_{w2} + \cdots + P_{w5} \cdot S_{w5})$$

其中  $s$  指预算成本,  $w$  指估算的工作量.

表 3

阶段	单元价格
项目说明	$S_{w1}$
总体设计	$S_{w1}$
单元处理	$S_{w1}$
组装集成	$S_{w1}$
项目验收	$S_{w1}$

当然,在超工期情况下,就要加上由于超期而引起的超工作量花费以及对顾客的赔偿,即

$$s' = s + (t - T_0)P_t + S_{t-T_0} \quad (5a)$$

其中  $S_{t-T_0}$  指由于超期而引起的超工作量花费,  $T_0$  指正常预测工期,  $P_t$  指单位延期赔偿。

### 参考文献

- 1 Barry W Boehm. Software engineering economics. New Jersey: Prentice-Hall, 1981.
- 2 MICROSOFT. Project for windows. Version 3.0, 1992.
- 3 Halsall Fred. Data communications. Computer Networks and OSI. 1985.
- 4 CEGELEC. Gestion de Projet. 1993.

## SOFTWARE PROJECT ESTIMATION

Wang Liqiang

(Center for Integrated Engineering Southeast University Nanjing 210096)

**Abstract** The software project estimation is an important part of the software project management. This paper is justly a resume of author's working experience in an European company.

**Key words** Methodology, software project estimation, software project management, COCOMO model, software engineering.