应用COCOMO Ⅱ模型估算软件开发工作量

任永昌，赵 颖

(辽宁工业大学 计算机科学与工程学院，辽宁 锦州 121001)

摘 要：准确的工作量估算是保证软件开发正常进行的必要手段，COCOMO Ⅱ模型是估算的重要方法。文中运用功能点分析法估算软件规模，其步骤是估算初始功能点数、估算调整后的功能点数、将功能点转化为源代码行；设计出标称进度工作量、调整进度工作量、标称进度、调整进度等公式对工作量进行估算。

关键词：COCOMOII模型；软件开发工作量；功能点分析法

中图分类号：TP311 **文献标识码**：B **文章编号**：1673-0569(2007)03-0268-04

1 引言

软件开发工作量估算是成本估算的基础。Barry W. Boehm 教授1981年在其著作《Software Engineering Economics》中提出了用于软件成本估算的早期构造性成本模型（COCOMO，Constructive Cost Model），在80年代早期被广泛使用。90年代后，软件项目管理和开发技术与工具发生了很大变化，早期COCOMO模型已不再适应新的软件成本估算和过程管理的需要，1994年Boehm重新研究和调整原有模型，根据未来软件市场的发展趋势，发表了COCOMO Ⅱ模型。Barry W. Boehm又通过对大量软件开发项目进行测算，推出了COCOMO Ⅱ. 2000。本文是COCOMO Ⅱ. 2000在软件工作量估算中的具体应用。

2 规模估算

规模估算是成本估算的基础。COCOMO Ⅱ中规模表示为源代码千行数（KSLOC）。常用的方法有工作分解结构、类比评估技术、Parkson法则、专家判定技术、功能点分析法等。其中功能点分析法是基于数学理论、适用于项目的各个阶段，是COCOMO Ⅱ提倡的一种方法。

2.1 初始功能点数

运用计划评审技术（PERT，Program Evaluation and Review Technique）估算出表1所示每类功能点的数量后，再对功能点的复杂性（用加权因子表示）做出判断，分成简单、一般、复杂三种情况[6]，初始功能点数（IFP，Initialization Function Potions）是通过表1计算出来的。

2.2 调整后的功能点数

初始功能点数是通过建立一个标准来确定某个特定的测量参数（简单、一般、复杂）进行估算，加权因子的确定带有一定的主观性。初始功能点数与技术复杂因子（TCF，Technical Complexity Factor）相乘得到调整后的功能点数作为软件规模估算的功能点数。

收稿日期：2007-07-09。

基金项目：国家"863"计划资助项目(No:2005AA501560)，国家外专局资助项目(No:WJ200321001)

作者简介：任永昌（1969-），男，副教授，博士，硕士生导师，主要从事计算机软件的开发与设计、软件开发成本估算方法的研究。

表1　初始功能点数估算表

功能点类别	数量	加权因子			初始功能点数（IFP）（=数量×加权因子）
		简单	一般	复杂	
外部输入		3	4	6	
外部输出		4	5	7	
外部查询		3	4	6	
外部接口文件		7	10	15	
内部逻辑文件		5	7	10	
总计数值					

技术复杂因子 TCF 共由 14 个因素组成，如表 2 所示。每个因素按照其对系统的重要程度分为六个级别，如表 3 所示。

表2　技术复杂因子的组成

序号	名称	序号	名称
1	可靠的备份和恢复	8	联机更新主文件
2	数据通信	9	复杂的输入输出
3	分布式处理	10	复杂的内部处理
4	系统的重要性	11	代码的可重用性
5	稳定实用的操作环境	12	数据的转换与安装
6	联机数据处理	13	完善的功能和性能
7	多重屏幕和多重操作	14	易于修改和维护

表3　权重表（Fi 的取值）

0	1	2	3	4	5
没有影响	偶有影响	轻微影响	一般影响	较大影响	严重影响

TCF 可用公式 1 计算出来：

$$TCF = 0.65 + 0.01 \times \left(\sum_{i=1}^{14} F_i\right) \tag{1}$$

调整后的功能点数 FP 可由公式 2 计算出来。

$$FP = IFP \times TCF \tag{2}$$

2.3　功能点转换为源代码行数

功能点和源代码行是从两个不同的角度来度量软件规模，它们之间存在着较强的相关性。对于具体的软件开发部门，可根据该部门历史数据经过统计处理获得功能点数和源代码行之间的关系。表 4 给出了在不同编程语言环境下每个功能点对应的源代码行数的参考值。

表4　不同编程语言下 FP 与 LOC 间换算关系

编程语言	LOC/FP	编程语言	LOC/FP
Java	53	Unix Shell Scripts	107
Visual C++	34	Lisp	64
Visual Basic	29	4GL	20
PowerBulider	16	Prolog	64
Ada95	49	Pascal	91

数据来源：《Software Cost Estimation With COCOMO Ⅱ》。

3　工作量估算

工作量用人月（PM Person Months）表示，一个人在一个月内从事软件开发的时间数。每人月的

人时数（PH/PM，Person－Hours per Person－Month），是一个可调整的系数。根据不同情况，工作量可用标称进度（NS，Nominal－Schedule）表示，也可用调整进度（AS，Adjusted－Schedule）。标称进度公式不包括要求的开发进度（SCED，Required Development Schedule）成本驱动因子（见表6）。SCED反映项目面临的进度压力。

标称进度工作量PM估算公式为：

$$PM_{NS} = A \times (Size)^{E} \times \prod_{i=1}^{16} EM_i \tag{3}$$

调整进度工作量PM估算公式为：

$$PM_{AS} = PM_{NS} \times SCED \tag{4}$$

式（3）中A＝2．94，是常数。Size为规模估算的源代码千行数（KSLOC），指数E体现了五个比例因子（SF，Scale Factors）的作用，说明了不同规模的软件项目具有的相对规模经济和不经济性。当E的值大于1时，所需工作量的增加速度大于软件规模的增加速度，体现出规模不经济性；E值小于1时表示规模经济性。指数E的计算公式为：

$$E = B + 0.01 \times \sum_{j=1}^{5} SF_j \tag{5}$$

式（5）中B＝0．91，是常数。每个比例因子SF都有一个等级变动范围，从“很低”到“极高”，每个等级都有一个权重，见表5。

表5 比例因子SFj值

比例因子	很低	低	标称	高	很高	极高
先例性	6．20	4．96	3．72	2．48	1．24	0．00
开发灵活性	5．07	4．05	3．04	2．03	1．01	0．00
体系结构/风险化解	7．07	5．65	4．24	2．83	1．41	0．00
团队凝聚力	5．48	4．38	3．29	2．19	1．10	0．00
过程成熟度	7．80	6．24	4．68	3．12	1．56	0．00

数据来源：对[4]中数据进行综合整理而来。

式（3）中EM（Effort Multiplier）是工作量乘数，表示成本驱动因子对开发工作量的影响程度。如果作为乘数的成本驱动因子等级导致更多的软件维护工作量，则相应的EM高于1．0。相反，如果等级减少开发工作量，则相应的EM小于1．0。

成本驱动因子分为四大类，每个大类又分为若干小类。成本驱动因子见表6，每个成本驱动因子的工作量乘数见表7。

表6 成本驱动因子

驱动因子	说明	驱动因子	说明
产品	REPL 软件可靠性	人员	ACAP 分析员能力
	DATA 数据库规模		PCAP 程序员能力
	CPLX 产品复杂性		PCON 人员连续性
	RUSE 可复用开发		APEX 应用经验
	DOCU 匹配生命周期的文档编制		PLEX 平台经验
			LTEX 语言和工具经验
平台	TIME 执行时间约束	项目	TOOL 软件工具的使用
	STOR 主存储约束		SITE 多点开发
	PVOL 平台易变性		SCED 要求的开发进度

表7　工作量乘数 EM

序号	小类	很低	低	标称	高	很高	极高
1	RELY	0. 82	0. 92	1. 00	1. 10	1. 26	
2	DATA		0. 90	1. 00	1. 14	1. 28	
3	CPLX	0. 73	0. 87	1. 00	1. 17	1. 34	1. 74
4	RUSE		0. 95	1. 00	1. 07	1. 15	1. 24
5	DOCU	0. 81	0. 91	1. 00	1. 11	1. 23	
6	TIME			1. 00	1. 11	1. 29	1. 63
7	STOR			1. 00	1. 05	1. 17	1. 46
8	PVOL		0. 87	1. 00	1. 15	1. 30	
9	ACAP	1. 42	1. 19	1. 00	0. 85	0. 71	
10	PCAP	1. 34	1. 15	1. 00	0. 88	0. 76	
11	PCON	1. 29	1. 12	1. 00	0. 90	0. 81	
12	APEX	1. 22	1. 10	1. 00	0. 88	0. 81	
13	PLEX	1. 19	1. 09	1. 00	0. 91	0. 85	
14	LTEX	1. 20	1. 09	1. 00	0. 91	0. 84	
15	TOOL	1. 17	1. 09	1. 00	0. 90	0. 78	
16	SITE	1. 22	1. 09	1. 00	0. 93	0. 86	0. 80
17	SCED	1. 43	1. 14	1. 00	1. 00	1. 00	

数据来源：《Software Cost Estimation With COCOMO Ⅱ》

4　结束语

Boehm 教授在 COCOMO Ⅱ中引入了贝叶斯校准和 COCOMO Ⅱ建模方法学，同时推出了针对特定机构裁剪的 COCOMO Ⅱ模型，并把功能点分析方法用于规模估算，以提高估算精度，取得了较好的效果。随着应用领域、开发方法等方面的变化，对 COCOMO Ⅱ提出了新的挑战。COCOMO Ⅱ模型需要不断演化，吸收其它估算方法的优点，增强模型的灵活性、可扩展性和可操作性，才能表现出强大的生命力，才能使软件工作量估算更接近实际值。

参考文献：

[1]高英慧，任永昌，叶景楼. COCOMO 模型在软件维护成本及进度估算中的应用[J]. 渤海大学学报：自然科学版，2007(1)：87－89.
[2]K Pillai. A Model for Software Development Effort and Cost Estimation[J]. IEEE Transactions on Software Engineering. 1997, Vol. 23(8).
[3]张利萍，李宏光. 灰色神经网络预测算法在 DMF 回收过程中的应用[J]. 微计算机信息，2005(1)：183－185.
[4]Barry W. Boehm. Software Cost Estimation With COCOMO Ⅱ [M]. Pearson Education. 2005.
[5]杨松，杨文莲. 基于关键字和链接的搜索引擎优化等略[J]. 渤海大学学报：自然科学版：2006(3)：269－271.
[6]李帜，等. 功能点分析方法与实践[M]. 北京：清华大学出版社，2005.

Application of COCOMO Ⅱ in software development workload estimate

REN Yong-chang , ZHAO Ying

(1. Computer Science & Engineering College, Liaoning Technology University, Jinzhou 121001, China)

Abstract: Accurate workload estimate is an essential method of guaranteeing software development, COCOMOII is an important way for cost estimate. Function point analysis is usedto estimate software scale, inc luding initialization function potions estimate, adjustment function points estimate, transforming function point into KSLOC. The design ot the formulae ot the progress of nominal progress workload, the adjusted schedule, the nominal schedule, the estimate formula of adjustment progress and so on is used to estimate the schedule.

Key words: COCOMOII; software development workload estimate; function point analysis