

基于功能点 COCOMO 估算模型的应用

欧阳钟辉,李莉

[摘要] 使用 COCOMO 模型进行软件成本估算,可以将经济分析的一般概念和技术与软件工程紧密地联系起来;可以帮助项目经理测定各个软件开发阶段所需要的时间和工作量;可以进行软件的成本效益分析、盈亏分界分析及用户作出是否自主开发的决策。

[关键词] COCOMO 模型;软件工程;成本效益分析;盈亏分界分析

[作者简介] 欧阳钟辉,泉州师范学院陈守仁工商信息学院院长,电子商务研究所所长,厦门大学软件学院、泉州师范学院教授,博士,福建 厦门 361005;李莉,泉州师范学院硕士研究生,福建 泉州 362000

[中图分类号] G202 **[文献标识码]** A **[文章编号]** 1672 - 2728(2009)01 - 0056 - 04

软件成本估算将经济分析的一般概念和技术与软件工程紧密地联系起来,能够较好地进行软件的成本效益分析、盈亏分界分析及在面对自主开发还是购买产品的难题时作出决策。它为软件分析员在系统设计阶段进行实际的软硬件分析时提供可靠的依据,帮助项目经理确定各个软件阶段及活动需要的时间和 workload。

基于上述原因,国内软件行业也开始重视软件的思想方法并逐步地将这些方法和技术运用到实践中去。因此,研究和开发适合我国软件行业特点的软件成本估算技术是势在必行的。

一、软件成本估算的定义及过程

软件成本估算是对将要开发的软件项目所需的工作量和工作进度做出预测。从实践中看,软件成本估算是从软件项目中一些较为容易把握的属性出发,比如软件的功能点、源代码行数等,推导开发软件项目所需的工作量和工作进度。

软件成本估算并非资金意义上的估算,它仅对将要开发的软件项目的工作量和工作进度做出估算^{[1](P7) [2](P8) [3]}。

在对一个软件项目进行成本估算时一般要经过以下 3 个阶段:

(一)规模估算阶段。有两种基本策略:一是估算问题大小,如功能点;二是估算解决方案的大小,如源代码行数、模块数。

(二)在工作量和工作进度估算阶段,人们根据有关软件大小的信息来估算软件开发成本。一般来说主要有专家判定、类比法、成本估算模型等 3 种估算方法。

(三)估算反馈阶段。这里的反馈有两层含义:一层是针对成本估算方法本身而言的;另外一层含义则是指估算实践当中的阶段性结果反馈。

下面介绍的成本估算模型 - COCOMO 模型及其估算方法着重研究成本驱动因子,并根据国内软件行业情况,对这些因子进行了一一筛选,使其本地化。

二、COCOMO 模型的研究与改进

(一) COCOMO 模型简介。1981 年,Barry Boehm 在其经典著作《软件工程经济学》中,介绍了一种软件成本估算模型的层次体系,称为 CO-COMO (Constructive Cost Model, 构造型成本模型)^{[4](P7)}。

在利用 COCOMO 模型进行软件成本估算过程中,首先采用功能点或代码行度量法估算出软件项目的功能点数,再通过调整后的功能点数进行代码行转换,使用千代码行数作为描述项目规模的单位。应用五个比例因子,通过相关计算,可以将规模转化为工作量,并通过 17 个成本驱动因子对工作量进行调整。最后,采用进度计算公式,计算出开发该项目所需要的进度以及人数。

[项目基金] 国家科技部“十一五 国家科技支撑计划“制造业信息化工程”,2006DAF01A23;国家外专局:现代化港口大物流信息化创新示范平台,RJZ20063500037;福建省科技厅:“纺织服装企业信息化公共服务平台”(EIPSP),2006H0106

本文将基于功能点的规模估算作为一个单元单独介绍,现在先对工作量估算方法进行介绍和改进。

(二)工作量估算方法。以人月(Person - Month,简称PM)表示的工作量值PMNS,用公式2-1进行估算。

$$PM_{NS} = A \times Size^e \times \prod_{i=1}^M EM_i \quad 2-1$$

其中,Size表示估算的规模,单位是源代码千行数(KSLOC)。EM_i为工作量乘数,EM_i的个数n的值对于后体系结构模型是17,对早期设计模型是7。指数E的计算如式2-2所示:

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j \quad 2-2$$

参数A,B,EM_i,SF_j的值是根据数据库中历史项目的实际参数和工作量的值进行校准而获得的。

A可较准,一般取2.94。

B可较准,一般取0.91。

在COCOMO中,用工作量估算公式中的指数E体现不同规模的软件项目所具有的相对规模经济性和不经济性的最终影响合力。

若E < 1.0,则表明项目总体能够体现出规模经济性。如果项目的规模加倍,则工作量不会翻倍。项目的生产率也随着产品规模的增加而提高。

若E = 1.0,则表明项目的规模经济性和不经济性是平衡的。这种线性模型通常用于小项目的成本估算。

若E > 1.0,则项目就表现出总体的规模不经济性。这通常是由于人员交流开销的增大和大型系统集成开销的增长。

SF代表五个指数比例因子。具体见表1。

表1 五个比例因子

比例因子	说明
先例性(PREC)	表示以前是否开发过类似项目
开发灵活性(FLEX)	表示软件性能与已经建立的需求和外部接口规范的一致程度
体系结构/风险化解(RESL)	通过风险管理衡量项目的风险及建立体系结构的工作量
团队凝聚力(TEAM)	衡量项目相关人员的管理状况
过程成熟度(PMAT)	衡量项目过程的规范程度,主要围绕SEI的CMM而进行

(三)软件开发成本驱动因子的本地化分析和改进。为了提高候选因子的覆盖面,同时尽量避免重复,有必要对候选因子进行筛选,筛选因子的时候,有三个基本原则:一是一般重要性原则;二是独立性原则;三是简单性原则。

1.对规模经济性比例因子进行分析和改进。在COCOMO II模型的标称工作量估算公式中,规模经济型比例因子总共有五个,详见表1。对这五个比例因子分别作出筛选,筛选的结果如表2所示。

表2 规模经济型比例因子筛选结果

因子	筛选结果	理由
PREC	保留	项目的固有属性
FLEX	保留	项目的固有属性
REST	取极低级别值	企业所承担的项目多为中小型项目,这些项目仅需要在成熟的体系结构中选择一个适合的进行应用。此因子影响不显著。
TEAM	取较高级别值	国内中小企业项目人员地理位置集中,人员较少,项目成员之间相互熟悉,所有交流通畅,协调容易。
PMAT	取极低级别值	国内中小企业基本都未通过CMM(Capability Maturity Model,能力成熟度模型)认证,并且企业内部也基本没有严格的项目流程管理,项目开发还处于作坊式的开发模式。

综上所述,规模经济性指数 E可改进为如下:

$$E = B + 0.01 \times (\text{PREC} + \text{FLEX} + \text{RESL} + \text{TERM} + \text{RMA T})$$

$$= B + 0.01 \times (\text{PREC} + \text{FLEX} + 5.65 + 2.19 + 6.24)$$

$$= B + 0.1408 + 0.01 \times (\text{PREC} + \text{FLEX})$$

其中 B 的值可校准, COCOMO 中标称值一般取 0.91。

2 改进工作量乘数。在 COCOMO 模型中,成本驱动因子体现了项目的特征以及组织的差异,对成本驱动因子的改进主要有以下几点:

(1) 合并冗余成本驱动因子。中小型项目需求不是很复杂,配备人员较少,情况相对简单,对于这样的项目, COCOMO 的许多成本驱动因子就是冗余的,必须对冗余因子进行合并。

(2) 保留其标称值的成本驱动因子。有些成本驱动因子不适用中小型项目,在中小型项目中的作用无需考虑,取其标称值即可。

(3) 无需考虑其作用,但要弱化其标称值的成本驱动因子。由于 COCOMO 原模型的有些成本驱动因子可以无需考虑其作用,但其标称值不符合中国中小企业的实际情况,所以对这些成本驱动因子保留一个非标称的新值。

(4) 保留并增强其作用的因子。人员连续性因子 (PCON),描述了项目组成员的年/人员周转率。本文将这两个因子的等级量化,其量化描述。如表 3 所示。

表 3 人员连续性 (PCON) 因子

因子等级	极低	很低	低	标称	高
量化描述	80%	48%	24%	12%	6%
工作量乘数	1.70	1.29	1.12	1.00	0.90

CPLX:产品复杂性,该成本驱动因子描述产品本身的实现复杂程度,是不受企业规模、国情差别环境影响的,因此保留 COCOMO 的等级划分。如表 4 所示。

表 4 产品复杂性 (CPLX) 因子

因子等级	很低	低	标称	高	很高
工作量乘数	0.73	0.87	1.00	1.17	1.34

(5) 增加的因子。对于一些 COCOMO 模型并未考虑,但对国内中小企业却有较大影响的问题,可以增加新的参数以反映其对估算结果的影响效果。

三、基于功能点的规模估算

(一) 功能点分析法是由 20 世纪 70 年代末期 BM 公司的工程师 Allan Albrecht 首先提出来的,功能点估算的过程共有 7 个步骤^[5]。

通过综合 14 个 GSC 的影响程度 (degree of influence, DI),根据以下的公式计算功能点调整因子 VAF^[6]:

$$VAF = TDI \times 0.01 + 0.65$$

其中 TDI 等于 14 项 GSC 的 DI 总和。

计算开发型项目功能点计数用下式计算开发型项目的功能点计数:

$$FPC = (UFP + CFP) \times VAF$$

其中; FPC 是开发型项目功能点计数;

UFP 是未调整功能点读数;

CFP 是数据转换的功能点计数;

VAF 是值调整因子。

(二) 改进功能点分析法

1. 系统基本特征的调整。针对我国软件行业的特点和现状,对其提出改进的方案,对技术方面的管理特性进行删减,取消估算结果差异很小的系统特性。

(1) 保留数据通信、性能、在线数据输入、最终用户效率、安装难易程度、跨平台性这 6 个系统特性,取消其他 8 个系统特性。

(2) 增加新技术运用难度一个系统特性。由于采用了新技术,因此开发人员需要花费时间去学习并掌握,也就增加了项目开发的时间和精力,同时新技术导致风险的增加,即增加了难度。

2. 开发管理的影响因素的调整。我国软件行业的人员流动性大,开发人员的平均水平不高,这就给软件开发的管理增加了难度。因此,增加开发人员管理难度的调整因素。

3. 对功能点法在权值取值范围不妥当方面的改进。在于中小型软件中,每个基本功能组件通常比较简单,根据笔者的经验,中小型软件的规模一般不超过 500 个功能点。另外,在中小型软件中, LF 中的 DET 数目通常不足 50,而 EIF 中的 DET 数目通常不足 20,在这种情况下, RET 的数目就是决定复杂度的关键。因此,提出以下的复杂度确定规则:

(1) 对 LF,如果其包含 4 个或 4 个以上的 RET,则认为其复杂度为“平均”;否则认为其复杂度为“低”。

(2) 对 EIF,认为其复杂度为“低”。

最后,根据逻辑数据文件的类型(LF或EF)和复杂度,遵照IFPUG方法的规则,确定它们的功能点数。

对逻辑数据文件,经常进行的操作是增、删、改、查列表操作。对中小型软件,增加、修改、删除、查看这4种操作通常要读、写逻辑数据文件中的多数字段;而查看记录列表的操作通常只须读取少量字段。因此,提出如下规则:

(1)对每个LF,认为其至少对应3个复杂度与其相同的EI(对应增加、删除、修改操作),1个复杂度与其相同的EQ(对应查看LF中单个记录的操作),以及1个复杂度为“低”的EQ(对应查看LF中记录列表的操作)。

(2)对每个EF,认为其对应2个复杂度为“低”的EQ(对应查看EF中单个记录的操作,及查看记录列表的操作)。

由于对某些LF,可能会有较多的操作,因此除了应用以上两项规则,还需要根据实际情况进行调整。于是,基于类似的经验,提出如下规则:

(1)对每个LF,除了增加、删除、修改这3种操作外,如果系统中还存在3种或3种以上会对其内容产生影响的操作,则认为其额外对应3个低复杂度的EI。

(2)对每个LF,除了查看LF中单个记录及查看记录列表的操作外,系统中还存在3种或3种以上查看其中的数据的方式,则认为该LF额外对应3个低复杂度的EQ。

(3)对每个LF,判断对其需要几种统计或报表功能,将这个数目作为它对应的EO的数目。

(4)在上述分析过程中,如果同一功能针对多个LF起作用,则该功能只计算1次,但复杂度设为“高”。

最后,根据事务处理功能的类型和复杂度,遵

照IFPUG方法的规则,确定它们的功能点数。

一系列成功案例显示,使用基于功能点的CO-COMO估算模型所得到的估算结果与实际工作量很接近,相对误差平均值都在可接受范围内。而用未改进的估算模型计算出的结果与实际工作量的误差大于20%。经进一步比较,本文提出的软件成本估算法对本地软件项目进行软件成本估算上是有效的。该法简单易用,它为我们提供了一种易于学习、易于使用的软件成本估算方法。

[参考文献]

- [1] 张海藩. 软件工程导论[M]. 北京:清华大学出版社,1998.
- [2] 郑人杰. 软件工程[M]. 北京:清华大学出版社,1999.
- [3] Boehm W. Boehm. Software Engineering Economics[M]. Prentice Hall, 1991.
- [4] Barry W. Boehm. 软件工程经济学[M]. 李师贤,等,译. 北京:机械工业出版社,2004.
- [5] 加莫斯. 赫伦. 功能点分析——成功软件项目的测量实践[M]. 苏薇,盛轶阳,译. 北京:清华大学出版社,2003.
- [6] IFPUG Function Point Counting Practices Manual, Release 4. 1, International Function Point User Group, 1999.
- [7] ISO/IEC20926 - 2003 IFPUG 4. 1 Unadjusted Functional Size Measurement Method - counting Practices Manual[S]2003.
- [8] 欧阳钟辉. 大信息平台CIP[J]. 泉州师范学院学报(自然科学),2006,(4).

[责任编辑:钟山]