

软件项目中工作量估算方法的研究

齐卓亚

(中国长江三峡集团公司 信息中心 宜昌 443002)

摘要 软件项目工作量估算是软件项目成本估算的基础。合理的软件成本估算有助于业主方在以下方面做出理性的决策:在软件项目投资、招标、软件项目外包以及对软件项目开发过程中的监督管理等方面。COCOMO 模型是目前应用于软件项目成本估算方法中最广泛、最受欢迎的模型之一。本文首先研究并给出了软件项目工作量估算的常用方法,详细介绍了 COCOMO 模型和 COCOMOII 模型估算方法,并指出了模型修正的意义所在;然后介绍了工作量估算中软件项目规模度量的常用方法,重点介绍了基于功能点分析的软件规模度量方法;最后,通过研究分析前人提出的软件项目工作量估算“三步法”,针对算法的问题和不足,提出了完善的建议和改进的方法,并据此提出了新的软件项目工作量估算流程模型;同时,阐述了本文提出算法的合理性和有效性。

关键词 软件工作量 软件规模 估算

中图分类号 TP311.52

文献标识码 A

文章编号 5888

Research on the Method of Estimating the Workload in the Software Project

Qi Zhuoya

(Information Center, China Three Gorges Corporation Yichang 443002)

Abstract Software project workload estimation is the basis of software project cost estimation. Reasonable software cost estimation will help the owner side to make rational decisions in the following aspects: investment in software project, bidding, software project outsourcing, and the supervision and management in the process of software project development, etc. The COCOMO model is one of the broadest and most popular models in software project cost estimation. At first, this paper studies the commonly used method and software project effort estimation is presented, and COCOMOII model COCOMO model estimation method was introduced in detail, and points out the significance of model correction; This paper introduces the general method of measuring the scale of software project in the workload estimation, and introduces the software scale measure method based on function point analysis. Finally, through the research analysis software project put forward by the predecessors' effort estimation "three-step", in view of the problems and the insufficiency algorithm, puts forward some suggestions of perfecting and improving method, and accordingly puts forward a new software project effort estimation process model; At the same time, the rationality and validity of the proposed algorithm are expounded.

Keywords Software engineering Software workload Software size Estimation

一、引言

软件项目的工作量通常以人日或人月为单位,表示的是一个人在一天或者一个月内从事软件开发的时间数。软件项目工作量估算与软件项目的开发内容、开发团队、开发工具以及所使用的开发技术等因素有密切的关系。

软件项目工作量估算是制定软件开发计划的依据和前提。另外,软件项目工作量估算还是软件项目成本估算的基础。合理的软件成本估算有助于业主方在软件项目的投资、招标、外包以及对项目开发过程中的监督管理等方面做出理性的决策。

在估算软件项目工作量时,我们一般都先要对软件规

模进行度量,这也在一定程度上印证了于洪玉^[1]等人的观点:软件规模估算是软件开发工作量估算的前提和基础。软件规模的度量有两种基本的策略:第一个是度量问题的大小,如功能点;第二个是度量解决方案的大小,如源代码行数、模块数^[2,3]。文献^[4]列出了目前 7 种常见的软件规模度量的方法,给出了每一种方法的原理和适用情况,并对每一种方法的优缺点进行了分析。文献^[5]则和大多数文献一样,首先将软件规模估算的常用方法分为三类,然后再对每一类方法中方法分别讨论。其实,目前业界比较流行的软件规模度量方法只有两种:代码行估算法(Lines Of Code, LOC)和功能点分析估算法(Function Point Analysis,

FPA),前者是站在开发者的角度进行代码行的估算和规模的度量,这样的估算结果不易与用户(不懂技术)进行沟通,并且代码行的估算与估算人员的项目经验、技术水平、所采用的开发工具和开发方法、所使用的开发平台和开发语言等因素密切相关;而后者从交付给用户的功能角度出发,与具体的实现无关,相对比较客观。FPA是目前度量软件规模所有方法中的最受欢迎、也是使用最广泛的方法^[6-9]。文献^[10]给出了功能点分析估算法的具体方法和实现步骤。

软件项目工作量的估算精度受多种因素的影响,像软件项目的开发内容、软件项目的规模、开发团队的协作水平、团队的稳定性、所使用的开发工具、开发技术以及开发人员的技术水平等因素有密切的关系。多种不可控因素的影响给软件工作量估算增加了难度,而对于众多的工作量估算方法而言,不同的估算方法很难得到完全一样的估算结果。到目前为止,没有哪一种方法被证明是适用于所有的软件项目的,也没有哪一种方法被证明在任何情况下的估算都是最精确的^[11]。

二、软件项目工作量的估算方法

1、常见工作量估算方法

随着软件项目成本估算的重要性被普遍的认可,作为其前提和基础的软件项目工作量的估算工作变得越来越重要,对软件工作量估算的研究也成为软件工程和项目管理行业的热点。软件项目工作量估算的方法和模型很多,通常分为下面三类:基于既有理论的软件工作量估算方法、基于专家经验的软件工作量估算方法和基于回归模型的工作量估算方法。下面对每一分类方法进行简要的介绍^[12]:

(1)基于既有理论的方法,顾名思义,是将其他领域、行业的估算、预测理论应用于软件项目的工作量估算当中。其中比较典型的有人工神经网络(ANN,Artificial Neural Network)^[13]和贝叶斯网络(Bayesian Network)^[14]等,该类方法的估算精度不高,在实际项目中的应用也不广泛。

(2)基于专家经验的软件工作量估算方法,是比较常用的估算方法,但是,该类方法主要依赖于参与软件工作量估算的专家们的知识和经验,具有很大的主观性和不可重复性。该类方法中,比较常用的有自顶向下法(Top-down approach)、自底向上法(Bottom-up approach)、类比法等。

(3)基于回归模型的工作量估算方法,是目前工作量估算的主流方法,该类方法估算逻辑周密、严谨,过程可重复性强,因此,该类方法被广泛应用于软件工作量的

估算。比较典型的有COCOMO^[15,16](Constructive Cost Model,构造性成本模型)和SLIM^[17](Software Life-cycle Model,软件生命周期模型)等,其中,前者易估算、使用最广泛。

2、COCOMO 估算模型

构造性成本模型COCOMO是Barry Boehm在1981年提出来的,根据其能够估算的颗粒度,可以将该模型分为三个层次:基本COCOMO模型、中级COCOMO模型和高级COCOMO模型。下面对三个层次模型进行简要的介绍^[18],首先给出基本COCOMO模型的计算公式,如公式(1)所示:

$$PM=a*(Size)^b \quad (1)$$

其中,PM是Person Month的缩写,表示软件项目的工作量,单位为人月;Size表示软件项目的规模,单位用千行代码数KLOC表示,这里的代码行数不包括注释;a,b分别是工作量调整因子和规模调整因子,它们取决于项目的类型,因项目类型的不同而有不同的取值,如表1所示:

开发模式	a	b
有机	2.4	1.05
半有机	3.0	1.12
嵌入	3.6	1.2

表1 基本COCOMO模型中a,b的取值

其中,有机表示相对简单的小项目,开发人员对项目有较好的理解和较为丰富的工作经验,并且使用熟悉的环境进行开发;嵌入则表示在十分严格的约束条件下开发,需要解决的问题很新,没有经验可以借鉴;半有机则介于两者之间,其规模和复杂性都属于中等程度。在基本COCOMO模型中,软件项目的工作量是软件项目规模的函数;同时,我们还可以发现,在基本模型中,只考虑了软件规模对软件项目工作量的影响。中级COCOMO模型则在基本COCOMO模型的基础上,考虑了产品、硬件、人员、项目对软件项目工作量估算的影响,因此,中级COCOMO模型适用于能够估算出项目的代码行数,并且对软件项目有较好理解的情况;高级COCOMO模型则是在中级COCOMO模型的基础上,进一步考虑了软件工程中每一个阶段(如:分析、设计、编码、测试)的估算结果,因此,高级COCOMO模型适用于那些对每个阶段都有足够详细信息的软件项目的估算。由于篇幅关系,这里不再对中级COCOMO模型和高级COCOMO模型展开介绍。

随着软件项目管理和相关技术的发展,Barry Boehm于1994年对COCOMO模型进行了很大程度的升级,发表了

COCOMOII 模型,并于2000年推出了COCOMOII模型2000版本。COCOMOII模型使用三个螺旋式的过程模型:应用组装模型、早期设计模型和后体系结构模型。应用组装模型是一种基于对象点的度量模型,通过计算屏幕、报表、第三代语言(3GL)模块等对象点的数量来确定基本的规模;早期设计模型是为了支持在项目开始后的一个阶段而提出的设计模型,这一模型可以使用FPA和SLOC进行规模估算;后体系结构模型则主要服务于项目进入开发阶段,项目进入开发阶段后,需要确定一个具体的生命周期体系结构,从而项目就能够为估算提供更多更加准确的信息。在三个过程模型中,早期设计模型和后体系结构模型使用相同的函数来估算软件项目开发的工作量。

3、COCOMOII模型的估算原理和估算步骤

本节首先给出COCOMOII模型的估算公式和估算原理,然后给出使用该模型进行软件工作量估算的具体步骤。

这里先给出该模型工作量估算的计算公式,如公式(2)所示:

$$PM_{NS} = A * (Size)^E * \prod_{i=1}^n EM_i \quad (2)$$

其中,PM表示以人月(Person Month, PM)为单位的工作量;根据情况的不同,工作量可用标称进度(NS, Nominal-schedule)表示,也可用调整进度(AS, Adjusted-Schedule)表示。调整进度还考虑了项目面临的进度压力。 PM_{NS} 表示软件项目的标称进度工作量;常数A可以根据历史项目数据进行校准,在COCOMOII.2000中取值2.94。Size表示软件项目的规模,单位用源代码的千行数(KLOC)表示; EM_i 为工作量乘数,表示成本驱动因子对开发工作量的影响程度,在后体系结构模型中n的值取17,在早期设计模型中n取7。E是规模指数,体现了五个比例因子(SF, Scale Factors)的作用,体现了不同规模的软件项目具有的规模经济性,当E大于1时,说明工作量的增加速度大于软件规模的增加速度,从而体现出规模的不经济性;反之,当E小于1时,体现软件规模的经济性。指数E的计算公式如公式(3)所示:

$$E = B + 0.01 * \sum_{j=1}^5 SF_j \quad (3)$$

其中,常数B可以根据历史项目数据进行校准,在COCOMOII.2000中取值0.91。每一个比例因子SF都有一个等级范围,从“很低”到“极高”,不同的等级具有不同的权重,具体如表2所示:

表2 五个比例因子的等级权重

比例因子	很低	低	标称	高	很高	极高
先例性(PREC)	6.20	4.96	3.72	2.48	1.24	0.00
开发灵活性(FLEX)	5.07	4.05	3.04	2.03	1.01	0.00
体系结构/风险级别(RESL)	7.07	5.65	4.24	2.83	1.41	0.00
团队凝聚力(TEAM)	5.48	4.38	3.29	2.19	1.10	0.00
过程成熟度(PMAT)	7.80	6.24	4.68	3.12	1.56	0.00

由上面工作量估算的计算公式(2),可以很好的印证COCOMOII模型的估算原理:软件项目开发的工作量是软件项目规模和成本因子的函数。

通过前面对COCOMOII模型对工作量估算公式和估算原理的介绍,从而引出使用COCOMOII模型进行工作量估算的步骤:首先采用功能点分析法估算出项目的功能点数,然后通过调整后的功能点数转换为千行源代码数KLOC,从而得到软件项目的规模;再根据与规模指数相关的五个比例因子,经过相关的计算,将规模转化为工作量;最后通过17个成本驱动因子对工作量进行调整。有了软件项目的工作量,采用进度计算公式,还可以计算出该项目的开发进度和人数。使用COCOMOII模型进行工作量估算的步骤还可以使用下面的流程图1表示:

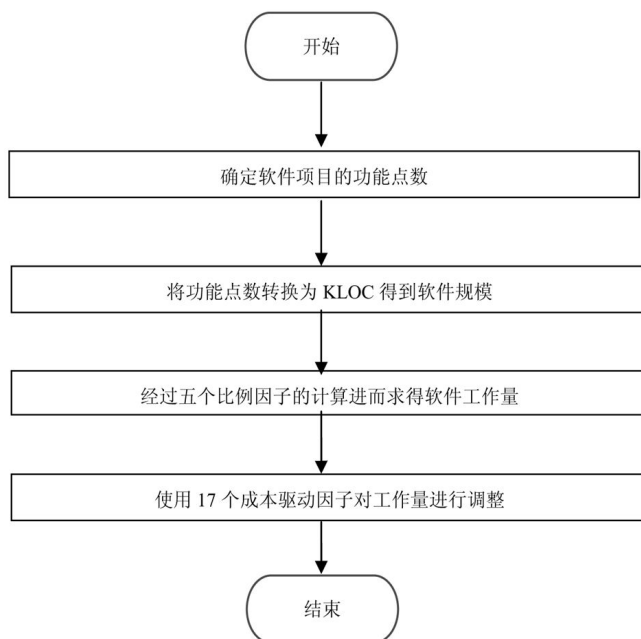


图1 COCOMOII模型工作量估算流程

软件规模的估算将在下面的章节详细介绍;上面我们给出了工作量计算的公式、与规模指数有关的五个比例因子的等级权重等详细信息;COCOMOII的后体系结构模型中提供了17个成本驱动因子,用来调整标称的工作量PM,

从而使软件项目工作量的估算精度达到最大限度。这些成本驱动因子分为四个大类:产品因子、平台因子、人员因子和项目因子,每个大类下面又有若干个小类。各个成本驱动因子的等级和权重如表3所示:

表3 成本因子的等级权重

成本驱动因子		很低	低	标称	高	很高	极高
产品因子	软件可靠性 (RELY)	0.82	0.92	1.00	1.10	1.26	
	数据库规模 (DATA)		0.90	1.00	1.14	1.28	
	产品复杂性 (CPLX)	0.73	0.87	1.00	1.17	1.34	1.74
	可复用开发 (RUSE)		0.95	1.00	1.07	1.15	1.24
	匹配生命周期的文档编制 (DOCU)	0.81	0.91	1.00	1.11	1.23	
平台因子	执行时间约束 (TIME)			1.00	1.11	1.29	1.63
	主存储约束 (STOR)			1.00	1.05	1.17	1.46
	平台易变性 (PVOL)		0.87	1.00	1.15	1.30	
人力因子	分析员能力 (ACAP)	1.42	1.19	1.00	0.85	0.71	
	程序员能力 (PCAP)	1.34	1.15	1.00	0.88	0.76	
	人员连续性 (PCON)	1.22	1.12	1.0	0.90	0.81	
	应用经验 (APEX)	1.19	1.10	1.00	0.88	0.81	
	平台经验 (PLEX)	1.20	1.09	1.00	0.91	0.85	
	语言和工具经验 (LTEX)	1.29	1.09	1.00	0.91	0.84	
项目因子	软件工具的使用 (TOOL)	1.17	1.09	1.00	0.90	0.78	
	多地点开发 (SITE)	1.22	1.09	1.00	0.93	0.86	0.80
	要求的开发进度 (SCED)	1.43	1.14	1.00	1.00	1.00	

4、模型中参数和相关因子修正的意义

从上文的分析可知,影响软件项目工作量估算的因素很多,同一个成本估算模型用在不同企业的软件项目中,或者用在一个企业内部不同时期、不同类型、不同内容或者不同性质的软件项目中,其最终的估算结果都很可能存在很大的差异。因此,这就要求我们在使用估算模型进行软件项目的工作量估算之前,要对模型中的参数和相关因子进行本地化的修正。

三、软件项目规模的度量方法

1、常见度量方法

目前业界比较流行的软件规模度量方法只有两种:代码行估算法 (Lines Of Code, LOC) 和功能点分析估算法 (Function Point Analysis, FPA),前者是站在开发者的角度进行代码行的估算和规模的度量,这样的估算结果不易与用户(不懂技术)进行沟通,并且代码行的估算与估算人员的项目经验、技术水平、所采用的开发工具和开发方法、所使用的开发平台和开发语言等因素密切相关;而后者从交付

给用户的功能角度出发,与具体的实现无关,相对比较客观。FPA是目前度量软件规模所有方法中的最受欢迎、也是使用最广泛的方法^[6-9]。

2、基于功能点分析的软件规模量方法

功能点分析法是用一个标准的单位来度量软件系统的功能,从而使用度量出来的功能数量转化为代码行数后,进而表示软件系统的规模。

功能点分析度量方法 FPA 主要是依据软件系统的复杂性和系统的特性来度量软件系统的规模,并且它不依赖于软件项目使用的开发语言和开发工具,尤其适用于在软件项目开发的早期阶段对软件的规模进行度量。

FPA 度量方法在下面的情况中特别适用:

(1)度量新的软件开发项目,而不是系统维护阶段的规模度量;

(2)度量包含很多输入输出和文件活动的应用软件;

(3)在软件开发的较早阶段,而不是软件开发的后期阶段,因为在开发的后期阶段,

系统细节逐渐明确,此时使用源代码行度量比使用功能点度量估算精度更高;

(4)在功能点估算方面,拥有经验丰富的专家;

(5)拥有相对充分的数据资料,并可以准确地将功能点转换为代码行。

FPA 估算方法需要对软件的内部基本功能和外部基本功能,然后根据技术复杂度因子对它们进行量化,最终产生软件的规模^[19,20]。

3、FPA 估算方法的估算步骤

下面首先介绍 FPA 方法的估算公式,进而引出 FPA 方法的估算步骤。FPA 方法的估算公式如公式(4)所示:

$$FP=UFP*TCF \quad (4)$$

其中,UFP(Unadjusted Function Point Count)表示未调整的功能点数,TCF(Technical Complexity Factor)表示技术复杂度因子,二者相乘之后得到调整后的功能点数 FP,并将调整后的功能点数作为软件规模估算的功能点数。通过 FPA 方法的计算公式,我们发现首先要求得 UFP,然后计算 TCF,最后才能得到用作软件规模的功能点数。

下面给出 FPA 方法的估算步骤^[5,19-22]:

第一步:确定计算范围和应用程序的边界。

第二步:统计五类功能计数项中,由一组需求所表达的功能点数。这五类功能计数项分别是外部输入(External Input, EI)、外部输出(External Output, EO)、外部查询(Exter-

nal Query, EQ)、外部接口文件(External Interface File, ELF)和内部文件(Internal Logical File, ILF)。其中,屏幕、表单、对话框、控件、文件等属于外部输入;报表、出错信息等属于外部输出;外部接口文件描述的是软件系统间传递或共享的文件应被每一个系统计算为外部接口文件类型;内部逻辑文件指的是每个逻辑的主文件,如数据的一个逻辑组合,它可能是某个大型数据库的一部分,也可能是一个独立的文件。

第三部:确定未调整功能点数(UFP)。UFP反映了软件中某个应用向用户提供的功能的数量。每一类功能计数项按照它们的复杂程度不同,分为简单、一般和复杂三个等级。所用功能计数项加权求和之后,就得到了未调整功能点数。下面给出每一类功能的复杂度权重,如表4所示:

表4 五类功能计数项的复杂度权重

功能类型复杂度	简单	中等	复杂
外部输入(EI)	3	4	6
外部输出(EO)	4	5	7
外部查询(EQ)	3	4	6
外部接口文件(EIF)	5	7	10
内部逻辑文件(ILF)	7	10	15

第三步:计算技术复杂度因子(TCF)。未调整功能点数尚不能表现去系统的全部特征,还需要通过14个技术复杂度因子对软件功能点进行调整。按照每个因子对系统的影响程度不同,每个因子的取值范围从0到5共分为六个等级,这14个技术复杂度因子分别是:备份和恢复的可靠性、使用配置的多少、操作的简便性、界面的复杂程度、可复用性、多重站点、数据通信、性能、联机数据输入、在线升级、数据处理额复杂性、安装的简易性、修改的难易程度和分布式等。TCF的计算公式如公式(5)所示:

$$TCF = 0.65 + 0.01 * \sum_{i=1}^{14} F_i, TCF \in [0.65, 1.35] \quad (5)$$

第四步:使用公式(4)求得调整后的功能点数。通常将计算得到的调整后的功能点数,作为用于估算软件规模的功能点数。

第五步:将功能点数转换为源代码行数。具体实施可以参考文献^[23]给出的换算关系,如表5所示:

四、软件项目工作量估算“三步法”

1、“三步法”介绍

从前面对COCOMOII工作量估算模型和软件规模度量方法FPA的介绍中,我们可以发现,对软件项目工作量估

表5 不同编程语言下功能点数和源代码行数的换算

编程语言	LOC/FP
Java	53
Visual C++	34
VisualBasic	29
PowerBulider	16
Unix Shell Scripts	107
...	...

算的工作十分的复杂和繁琐。其中,影响FPA规模度量的技术复杂度因子有14个;还要使用17个成本驱动因子对标称工作量进行调整。为了简单准确地估算软件项目的工作量,王养廷^[24]提出了使用“三步法”来估算软件项目的工作量。

“三步法”估算工作量的计算公式,如公式(6)所示:

$$E = a * Size^b * EAF \quad (6)$$

其中,E表示项目的工作量,单位为月;Size表示项目的规模,使用源代码千行数表示,文中并没有给出Size的求解说明和计算过程,所以,从文中我们无从得知Size是从何而来的;a,b是两个调节常数,论文通过企业中三个典型的历史项目来确定a和b的值;EAF是工作量调节因子,是项目开发采用技术的难易程度、开发人员的水平高低以及环境的复杂程度三个因素权重的乘积,如公式(7)所示:

$$EAF = M * T * Env \quad (7)$$

其中,M、T、Env分别表示人员系数、技术系数和环境系数。三个因素根据不同的等级,具有不同的权重,王养廷^[24]称其为类比系数表,具体如表6所示:

表6 类比系数表

	人员(M)	技术(T)	环境(Env)
高	0.8	1.3	1.2
中	1	1	1
低	1.6	0.88	0.9

使用“三步法”进行工作量估算的步骤为:

第一步,首先选择基本COCOMO模型作为工作量估算模型,从而简化复杂的项目参数。

第二步,通过历史项目数据修正模型参数。

第三步,根据修改后的模型采用类比法对工作量进行估算。

2、“三步法”存在的问题和不足

从前面对王养廷^[24]提出的“三步法”的介绍,我们可以发现:

(1)论文并没有给出“三步法”适用的项目类型或者项目所处的阶段;

(2)在公式(6)中,用Size表示项目的规模,却没有给出Size的求解方法和计算过程,所以,从文中我们无从得知Size是从何而来的;

(3)对于调节常数a,b,论文通过企业中三个典型的历史项目来确定a和b的值,文中并没有说明这“三个典型”的历史项目是怎么找出来的、应该具有什么样的特征或者说与待估算项目相比应该在哪些方面具有较大的相似性,另外,如果在现有的历史项目中找不够满足一定条件的历史项目或者说遇到崭新的、前所未有的项目怎么办;

(4)对于工作量调节因子EAF而言,论文只考虑了技术、人员和环境三方面的因素,这三个因素足够吗、在不同的项目中都是作为主要因素考虑吗?

3、对“三步法”的完善和改进

本节针对上面“三步法”存在的问题和不足,提出自己的思考和建议,并据此提出新的软件项目工作量估算步骤,希望在一定程度上提高算法估算的准确性。下面给出改进后的软件项目工作量估算的步骤:

第一步:根据可参考历史项目的多少选用合适的软件工作量估算方法。对于崭新的软件项目、在本企业数据库中几乎没有相同或者类似的历史项目数据或者说历史项目中能够参考的信息十分有限时,可以采用基于专家经验的软件工作量估算方法,具体可以参考1.1小节中的介绍;反之,对于有大量的与待估算项目类型相同或者相似的历史项目数据时,可以使用回归模型COCOMOII。本文主要讨论后者。

第二步:根据软件项目的类型选用合适的软件规模度量方法。如果待估算项目是开发类类型的软件项目,尤其是在开发的早期阶段及2.2小节中提出的FPA适用的其他情况下,可以优先选用功能点分析估算方法FPA度量软件的规模,此时要对影响规模度量的14个技术复杂度因子进行本地化处理,以更好的适应本单位项目的特点,从而得到更高的估算精度;如果待估算项目是维护类的项目或者在软件开发的后期阶段,项目的很多细节都十分明确时,

可以直接选用代码行估算方法LOC进行软件规模的度量。

第三步:运用COCOMO模型提供的软件工作量计算公式计算软件项目的工作量。

第四步:使用本地化的成本驱动因子对软件工作量进行调整。不同企业的软件项目、同一个企业不同时期和不同类型的软件项目,采用的成本驱动因子都具有一定的差异性。这体现在成本因子的选取个数以及相同成本因子本地化的取值的不同等方面。

上述估算步骤可以使用下面的流程图表示,如图2所示:

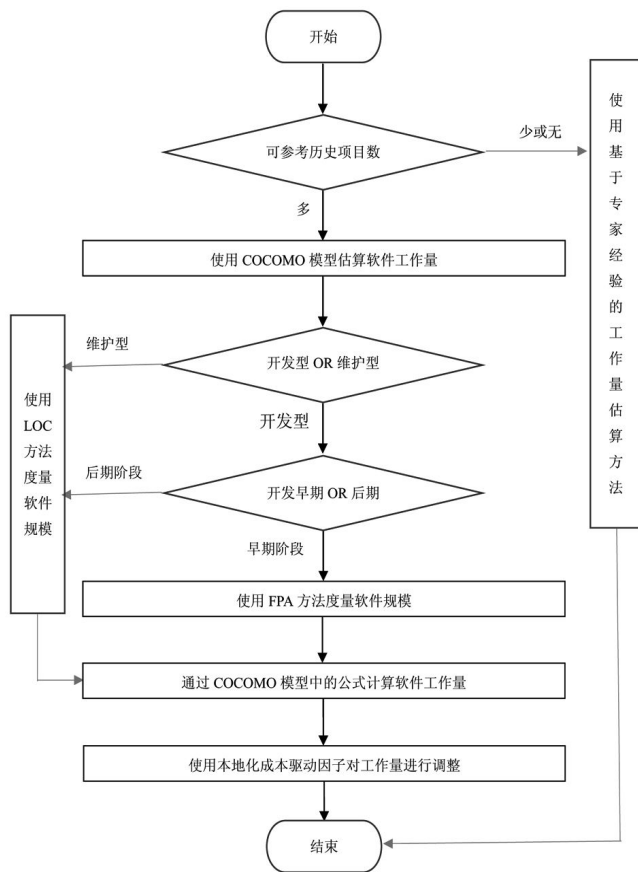


图2 本文提出的工作量估算步骤

4、案例分析

在文献^[24]中所提供的应用案例只是简单的介绍了“三步法”是如何使用的,并没有给出案例的实际工作量,因此,也没有将案例计算的结果和实际工作量进行比较,从而给出估算误差。因此,无法使用其所提供的案例来验证本文算法和其算法的估算结果哪个更接近案例的实际工作量。

五、结语

本文首先介绍了软件项目工作量估算的常用方法及其使用范围,重点介绍了COCOMO估算模型和COCOMOII估算模型;然后,本文介绍了软件规模度量的常用方法,重点

介绍了基于功能点分析的软件规模度量方法FPA;最后,本文介绍了前人提出的软件工作量估算“三步法”,针对算法中存在的问题和不足,本文给出了完善的建议,并提出了新的工作量估算步骤。

参考文献

[1]于洪玉,孟庆顺,杨富玉.应用IFPUG功能点分析方法进行软件规模估算实践[J].金融电子化,2010,(11):73-75.

[2]李莉.基于功能点的COCOMO II估算模型研究和应用[D].厦门:厦门大学,2008.

[3]王平,丁浩芳,李韬.结构型软件成本估算模型的研究与改进[J].计算机工程,2002,28(12):88-89.

[4]王鸣涛.软件规模估算方法综述[J].安阳师范学院学报,2012,(05):56-60.

[5]夏晓翔.软件项目估算管理方法研究[D].南京:南京理工大学,2006.

[6]曹济,温丽.软件项目功能点度量方法与应用[M].北京:清华大学出版社,2012:4.

[7]ISO/IEC 14143-6:2012. Information Technology-Software measurement-Functional Size Measurement-Part6: Guide for use of ISO/IEC 14143 series and related International Standards[S]. 2012.

[8]COSMIC FSM Method v4.01 Measurement Manual [EB/OL], 2015.04, <http://www.cosmic.com>.

[9]计春雷.全功能点方法和功能规模度量统一规模的研究与应用[D].武汉:华中理工大学,2010.

[10]赵颖,任永昌,邢涛.基于功能点的软件规模估算方法研究[J].辽宁工业大学学报(自然科学版),2008,28(5):281-283.

[11]李奇,舒风笛,伍书剑,等.软件工作量最优线性组合估算方法[J].计算机工程与设计,2008,29(23):6029-6036.

[12]马剑.软件开发工作量估算模型研究及其在项目中的应用[D].保定:华北电力大学,2012.

(上接第32页)

[3]顾正刚,毕海峰.高校网络教育资源建设与共享机制研究[J].中国教育信息化,2009(19):44-46.

[4]史文崇.我国网络教育资源的建设机制及运作[J].教育信息化,2005(9):54-55.

[5]李朝福.区域性高校体育网络教育资源共享机制的构建[D].牡丹江:牡丹江师范学院,2013.

[13]Tadayon N. Neural network approach for software cost estimation[C]. Proceedings of the International Conference on Information Technology, 2005, 10(2): 10-30.

[14]Yang D, Wan YX, Tang ZN, et al. An extension of COCOMO II for cost estimation with uncertainty[C]. Proceedings of Software Process Workshop/Workshop on Software Process Simulation and Modeling, 2007, 8(9): 5-15.

[15]Barry W. Boehm. Software Engineering Economics [M]. New York: Prentice-Hall, 1981: 2-20.

[16]李师贤等译. Barry W. Boehm. 软件工程经济学[M]. 北京:机械工业出版社, 2004: 5-15.

[17]Norman E. Fenton, Shari Lawrence Pfleeger. Software Metrics: A Rigorous and Practical Approach[M]. Boston: Thomson Learning, 1998: 82-88.

[18]人月神话.COCOMOII成本估算模型[EB/OL]. http://blog.sina.com.cn/s/blog_493a8455010005jy.html, 2006-08-23.

[19]Jeffery D R, Low G C. A comparison of function point counting techniques [J]. IEEE Transactions on software engineering, 1993, 19(5): 529-532.

[20]覃征,杨利英,高勇民,等.软件项目管理[M].第1版.北京:清华大学出版社,2004.

[21]韩万江,姜立新.软件项目管理案例教程[M].第1版.北京:机械工业出版社,2005.

[22]付雅芳,杨任农,刘晓东,等.基于灰色关联分析的软件工作量估算方法[J].系统工程与电子技术,2012(11): 2384-2389.

[23]Barry W Boehm. 软件成本估算 COCOMOII 模型方法[M]. 北京:机械工业出版社,2005.

[24]王养廷.软件项目工作量估算方法研究与应用[J].华北科技学院学报,2014,11(4):54-56.

作者简介

亓卓亚(1984~),男,硕士,研究方向:软件开发与软件项目管理。

[6]柯和平,李春林.基于网络技术的区域性教育资源库共建共享机制研究[J].电化教育研究,2005(6):19-20.

作者简介

于本成(1981.08~),男,汉,辽宁大连人,硕士副教授;

臧博(1981.08~),女,汉,江苏徐州人,硕士讲师;

宋培森(1986.02~),男,汉,江苏徐州人,硕士助教。