

功能点估算法及其在电信行业软件中的应用

白秀广 杨洪敏 吴俊

北京邮电大学工业与信息化部信息经济与管理重点实验室 北京 100876

摘要 随着信息时代的来临,信息系统在企业信息化的进程中所起的作用越来越大,然而目前对信息系统规模的估算还没有很好的方法,特别是甲乙双方都认可的方法。本文在比较目前较流行的软件规模估算方法的基础上,引入了目前逐渐被国内外软件业认同的规模估算方法——功能点方法,并进行了详细阐述和实证分析。由于功能点法简单易操作,且能通过估算软件规模进而估算软件成本,从而大大提高了甲方的议价能力,因此越来越受到重视。

关键词 软件规模;功能点;信息系统

引言

随着知识经济、信息时代的来临,计算机软件业迅猛发展,对软件开发实施有效管理是决定软件产业发展的至关重要的因素。而成功的软件系统开发意味着按时、按预算地交付符合用户和组织需求的软件系统,从国内外软件项目的开发经验来看,影响系统开发成败的因素有多种,而软件开发成本估算和管理控制是其成功的关键因素之一。如果估算周期过长、费用过高,就会造成企业资源的巨大浪费;如果估算的周期过短、费用过低,就会造成整个项目失控,远远超过预算和交付日期^[1]。因此,在一个软件项目开发之前或初期,对该项目的规模大小、所需资源、进度等按照软件工程的要求做出一个合理估算非常重要。另外,对于客户来说,如何估算自己需要开发的软件规模进而推算出成本也很重要,特别是对于现在的买方市场,尤其是像电信企业级的信息化系统,功能繁多,资金庞大,如何了解开发费用而不被开发商的报价所迷惑也是很必要的,因此,找出一种既被开发商认可又适合用户的估算方法势在必行。

1 评估方法的比较

自20世纪80年代,对软件规模测算的各种各样的方法就开始被提出来,但他们都较依赖于开发语言,

从目前来看,常用的软件估算方法有:专家判定法、代码行^[2]、COCOMOII^[2-5]模型、功能点^[4-6]等,这些方法各有优缺点,可总结为如图1。

从图1可以看到,在这四种方法中,专家法最简单易行,但误差最大,不适合开发商使用;代码行由于与开发语言有关,且误差也较大,对有经验的开发商有利,但对客户来说没法掌握;而COCOMOII模型测算的比较准,但是参数的取值比较困难,所以也不太适合客户评价;最近发展起来的功能点法则较好的解决了这些问题,可以同时被开发商和客户所接受,对于开发商来说,通过应用功能点法凭借自己的经验可以使误差不超过30%,而对于客户来说,可以不必详细了解功能的具体开发和实现过程,而根据功能的复杂度粗略估算出系统的规模,通过较多的经验的积累,也可以较好的估算出系统的规模进而了解开发成本。下面我们将针对功能点法给出详细的介绍。

2 功能点简介

功能点FP (Function Point) 的概念是工程师A.J. Albercht在1989年提出的,他还建立了工程预算功能点分析法FPA (Function Point Analysis)。近年来,功能点测算法已成为国际主流的软件规模测算法,并于2003年被国际标准化组织接受为国际标准^[7]。

相对于传统代码行法(SLOC),功能点估算(FPA)

	优点	缺点
Delphi估算法	<ul style="list-style-type: none">• 没有历史数据的情况下,这种方式适用于评定过去与将来,新技术与特定程序之间的差别• 测算快速,测算本身的成本低	<ul style="list-style-type: none">• 依赖于专家的技术指导• 主观性较强• 误差较大,适用于准备开发阶段
代码行	<ul style="list-style-type: none">• 方便计算、容易监控• 能反映程序员的思维能力	<ul style="list-style-type: none">• 代码行数的含糊不清且与语言有关• 不能正确反映一项工作的难易程度以及代码的效率
COCOMOII模型	<ul style="list-style-type: none">• 模型简单、概念明确、应用简便• 适用于三种类型的软件项目:组织模式——较小的、简单的软件项目• 半分离模式——一个中等的软件项目• 嵌入模式——必须在一组严格的硬件、软件及操作约束下开发的软件项目	<ul style="list-style-type: none">• 传统的COCOMOII模型无法处理COTS产品的集成过程中产生的成本,具有一定的局限性• 参数取值较困难
功能点法	<ul style="list-style-type: none">• 功能点度量和具体的技术、环境、平台无关,甚至和人员无关,可以度量一个功能• 功能点度量可以用于项目管理,会使项目管理包括项目范围的管理都能给出一个非常客观量化的数据• 功能点可以较准确的估计系统的规模,成本及进程等,偏差较小• 面向功能而不使用代码行进行估算,更适用于前期的估算	<ul style="list-style-type: none">• 功能点问题就是计算的复杂性• 功能点功能方法在实际应用中还有不断的变化和改进的,针对不同的项目有的改进还比较复杂• 在描述人工成本、开发时间和规模之间的内在联系方面较为逊色且有一定的主观性

图1 各估算方法优缺点比较

法更加侧重于从业务的视角来分析软件的规模大小。功能点分析得出的软件功能点数代表的是软件的逻辑规模(logical size),这正是甲方所关心并能理解的,而代码行估算法估计得出的软件代码行数反映的是软件的物理规模(physical size),这难以为甲方所理解。

在FPA中,任何一个软件都被看作是由外部输入处理、外部输出处理、外部查询处理、内部逻辑文件和外部参照文件五种要素组成,如图2所示。

外部输入处理EI(External Input)是获得数据的过程,对用户的输入进行相关的处理,它是第一项交易功能,指用户可以根据需要通过增、删、改来维护内部逻辑文件。

外部输出处理EO(External Output)是反馈数据的过程,完成对票据、报表等的输出,显示结果就是经过调用维护数据和参考数据获得的。

外部查询处理EQ(External Inquiry)是针对用户的查询请求,输出相应的检索结果。为了获得这项结果,用户要输入选择信息抓取符合条件的数据。此时没有对数据的处理,是直接从所在的文件抓取信息。

内部逻辑文件ILF(Internal Logical File)是在信息系统内部,为了完成相关功能使用的逻辑文件,包括顺序文件、数据库表、临时文件等,它在系统中的逻辑分组是由最终用户维护的。

外部参照文件EIF(External Interface File)是该系统和外部其他信息系统为了交换数据而使用的接口文件。在这种情况下,用户不负责维护数据,数据在另一系统中驻留由其他用户进行维护。该数据只供系统用户参考使用。

其中前三种(EI、EQ、EO)属于事务功能型文件(Transaction function type file),它们涵盖了终端用户和信息系统对象之间存在的所有可能的交互内

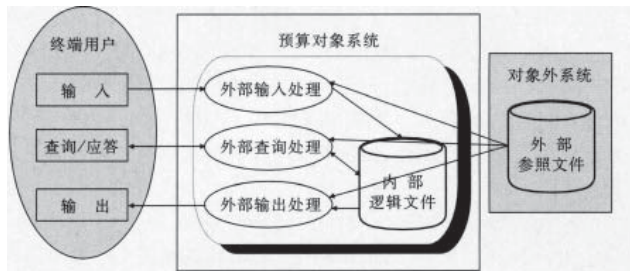


图2 软件的5种元素

容，只要明确了该系统和终端用户的接口就可以正确估算上述三种处理。后两种属于数据功能型文件(Data function type file)，系统功能确定后，使用文件个数也可以被估算出来。

利用FPA法测算软件规模可细分为6个步骤进行：

步骤一：确定EI、EQ、EO、ILF、EIF个数

1) EI的判定基准

① 数据是从系统边界外部获取的

② 事务处理是对ILF的插入、修改、删除操作，

但当数据是控制信息时，此规则不适用

2) EQ的判定基准

① 从系统边界外部获取数据

② 向系统边界外部输出数据

③ 如果包含下列处理逻辑，则不是EQ：

◆ 一个以上的数学运算处理

◆ 由基础数据生成新的数据

◆ 对一个以上的ILF进行插入、修改、删除操作

◆ 执行系统动作的变更

3) EO的判定基准

① 向系统边界的外部输出数据

② 一般可以包含下列业务处理逻辑：

◆ 一个以上的数学运算处理

◆ 由基础数据生成新的数据

◆ 对一个以上的ILF进行插入、修改、删除操作

◆ 执行系统动作的变更

4) ILF的判定基准

在计测系统范围内，有检索操作，同时也有插入、更新、删除操作的数据

5) EIF的判定基准

① 在计测范围内的系统检索，在计测范围外的系统保存的数据

② 在计测系统范围内，没有插入、更新、删除等操作的数据

③ 是计测范围外系统的ILF

步骤二：确定各要素的复杂程度

三种处理EI、EQ、EO的复杂程度通常是用该处理

中使用文件个数(通常对应为数据库表数)以及用到的文件中的项目数(通常对应为数据库表的字段数)来度量的，复杂程度与文件数和项目数成正比。即用到的文件数约多，项目数越多，复杂程度就越高。

其中评价EI的复杂度见表1：

表1 EI的复杂度

项目数	0-1	2	>2
1-4	低	低	中
5-15	低	中	高
>15	中	高	高

评价EO、EQ的复杂度见表2：

表2 EO、EQ的复杂度

项目数	0-1	2-3	>3
1-5	低	低	中
6-19	低	中	高
>19	中	高	高

文件ILF、EIF的复杂程度通常是用该文件的纪录种类数和项目数来度量的，记录种类越多，项目数越多，复杂程度就越高(见表3)。

表3 ILF、EIF的复杂度

项目数	1	2-5	>5
1-19	低	低	中
20-50	低	中	高
>50	中	高	高

步骤三：确定处理和文件的功能点权值

确定了复杂程度后，要对每种复杂程度的处理和文件赋予权值，以便计算出相应的功能点数。例如：EI处理中复杂度为高时，其权值为6，也就是说每个该种处理可以计算为6个功能点(6FP)，见表4。

表4 权值分布

要素	低	中	高
EI	3	4	6
EO	4	5	7
EQ	3	4	6
ILF	7	10	15
ELF	5	7	10

步骤四：未调整的功能点的计算

在上述功能分析的基础上，可由以下公式计算出系统的未调整功能点数：

$$FP = \sum \text{各个复杂度等级的信息域数量} \times \text{加权值}$$

步骤五：计算调整后的功能点

“未调整的功能点数”还不能完整表现出系统的全部基本特征，功能点方法通过14项系统基本特征进行调整，得到“已调整的功能点数”。14项系统基本特征如下：

- 1) 数据通信：应用程序与处理器之间直接通信的程度。
- 2) 分布式数据处理：应用程序部件间数据传输的程度。
- 3) 性能：需要考虑的响应时间和吞吐量对应用程序开发的影响程度。
- 4) 重度配置：计算机资源限制对应用程序开发的影响程度。
- 5) 处理速率：业务处理速率对应用程序开发的影响。
- 6) 在线数据输入：数据通过交互处理进入系统的程度。
- 7) 最终用户使用效率：被测应用程序中对人性化

14) 支持变更：修改处理逻辑或数据结构的容易程度。

每项系统基本特征的影响程度(DI)的取值范围都是0到5。

- 0 不考虑此特性或者对系统功能没有影响
- 1 偶尔对系统特性有影响的
- 2 有一定影响的
- 3 有比较多的影响的
- 4 有很重要的影响的
- 5 自始至终都有很强影响的

14项系统基本特征值相加就得到总影响程度(TDI)。值调整因子(VAF)及已调整功能点数的计算方法如下： $VAF = (TDI * 0.01) + 0.65$ ；

已调整功能点数 = $VAF * \text{未调整功能点数}$

步骤六：根据功能点进行软件投资规模测算。

3 实证研究

下面我们以某省的电信运营商的某个支撑系统作为例子给出功能点的应用实例：

步骤一：将系统的软件功能清单中列举的功能划入EI、EQ、EO、ILF、ELF中，并计算相应的个数，划分结果如下(见表5)：

表5 个数划分

外部输入处理	外部查询处理	外部输出处理	内部逻辑文件	外部参照文件
10	8	176	5	7

因素和易用性的考虑程度。

- 8) 在线升级：内部逻辑文件的在线更新程度。
- 9) 复杂处理：处理逻辑对应用程序开发的影响。
- 10) 可复用性：应用程序在其它应用程序中加以利用的程度。
- 11) 易安装性：转换或安装到新环境对应用程序开发的影响程度。
- 12) 易操作性：应用程序对操作方面的关注程度。
- 13) 多场所：为多个地方和用户机构开发应用程序的程度。

步骤二：根据功能描述，大致确定各要素的复杂程度；

步骤三：确定处理和文件的功能点权值；

步骤四：未调整的功能点的计算；

步骤二、步骤三和步骤四结果如表6所示：

步骤五：计算调整后的功能点

为14项系统基本特征赋值，这里为简单起见均赋值为4，即认为有很重要的影响，则

$$TDI = 14 * 4 = 56$$

$$VAF = (TDI * 0.01) + 0.65 = 0.56 + 0.65 = 1.21$$

表6 赋值表

要素	复杂度	数量	复杂度权值	未调整的功能点	未调整的功能点的汇总
外部输入处理 (EI)	高	0	6	0	249
	中	10	4	40	
	低	0	3	0	
外部查询处理 (EQ)	高	1	6	6	
	中	0	4	0	
	低	1	3	3	
外部输出处理 (EO)	高	10	7	70	
	中	1	5	5	
	低	0	4	0	
内部逻辑文件 (ILF)	高	5	15	75	
	中	2	10	20	
	低	0	7	0	
外部参照文件 (ELF)	高	3	10	30	
	中	0	7	0	
	低	0	5	0	

已调整功能点数 = $VAF \times$ 未调整功能点数
 $= 1.21 \times 249 = 301.29$ 。

步骤六：根据功能点试算成本：

美国单位功能点成本为1000美元，根据我国与美国软件开发人员的工资比例，及该地区的实际环境，取单位功能点成本为3000元，则该系统建设的软件成本为 $301.29 \times 3000 = 903870$ 元。而该系统建设的软件实际投资为98万。估算结果与实际投资的偏差不超过10%，比其他的软件估算方法如代码行等相比有较好的预测精度（代码行的预测精度偏差为50%~200%，与预测人员的经验有关）。

4 结语

本文首先对目前较流行的几种软件规模估算方法进行了比较，然后较详细的阐述了软件功能点的估算方法，并给出了实证研究，结果表明，功能点法在软件规模的估算上有很大的优势，且可以同时被开发商和用户所接受，特别是对用户来说，不必太了解软件的开发细节，就能得到较准确的估算规模，提高了自己的议价能力。特别是在当今信息化时代，像电信运营商这些特大型企业，支撑系统建设所占投资比重越来越大，如何积累经验，提高自己的软件估算能力从

而提高议价的能力是很重要的。相信，软件功能点法会越来越得到甲乙方的认同和流行开来。

参考文献

- [1] Magne Jnrgensen, Dag I K Sjnberg. Impact of Effort on Software Project Work[J]. Information and Software Technology, 2001(43):939-948
- [2] 周杰, 杜磊. COCOMOII—软件项目管理中的成本估算方法[J]. 计算机应用研究, 2000, (11):56-58
- [3] 胡樱. 软件成本估算[J]. 舰船电子工程, 2005, 25(6):14-18
- [4] 李明树, 何梅, 杨达. 软件成本估算方法及应用[J]. 软件学报, 2007, 18(4):775-795
- [5] 刘杰生. 软件成本估算方法与技术现状探讨[J]. 舰船电子工程, 2000, (4):13-18
- [6] 崔敬东, 左广. 软件开发成本估算方法的比较研究[J]. 技术经济, 2002(11):50-52
- [7] COSMIC. The Common Software Measurement International Consortium FFP Measurement Manual—Version 2. 2, 2003

作者简介



白秀广

北京邮电大学经济管理学院博士研究生。主要研究方向为经营决策研究，供应链管理，信息系统评价。



杨洪敏

联通宽带业务应用国家工程实验室有限公司高级研究员，北京邮电大学博士研究生。长期致力于电信运营商运营支撑系统相关规划、规范、数据模型及解决方案研究。

吴 俊

北京邮电大学经济管理学院副教授，硕士生导师。长期从事信息系统集成的研究。

Function Point Method and its Application in Software Assessment of Telecom Industry

Bai Xiuguang
Yang Hongmin
Wu Jun

Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract A software assess method that recognized gradually by the worldwide software industry on the basis of reviewing the existing assess methods is presented, and the method is explained in detail and demonstrated with an information system. The result shows that the method can assess the development cost accurately. This method can assess the development cost by assessing the software scale.

Keywords Software Scale; Function Point; Information System