

第3章

软件项目成本估算

无法评估,就无法管理。

——琼·玛格丽塔(Joan Magretta)

对于一个大型的软件项目,由于项目的复杂性及软件项目的独特性,开发成本的估算不是一件容易的事情,它需要进行一系列的估算处理,因此,主要依靠分析和类比推理的方法进行。

软件项目开发成本估算方法是一个很重要的问题,因为管理好成本才能避免造成人力、物力和资源的浪费,而软件项目开发成本的首要任务是先进行成本估算。所以,在软件开发前期对软件开发成本的估算就显得十分重要。

本章以软件项目开发工程的角度介绍成本估算在软件项目管理过程中如何进行成本估算及其估算过程、估算方法、估算等级等。本章共分为六部分对软件项目成本估算进行介绍。3.1节对项目成本估算的挑战进行概述;3.2节介绍项目成本估算的内容;3.3节介绍规模估算方法;3.4节详细介绍工作量估算方法;3.5节介绍开发工期估算;3.6节介绍成本估算方法。



视频讲解

3.1 项目成本估算的挑战

项目成本估算(Project Cost Estimate)指根据项目的资源需求和计划,以及各种项目资源的价格信息,估算、确定项目各种活动的成本和整个项目总成本的一项项目成本管理工作。

项目成本管理为使项目在批准的预算内完成而对成本进行规划、估算、预算、融资、筹资、管理、控制的各个过程,从而确保项目在批准的预算内完工。项目成本管理过程如下。

- (1) 规划成本管理:确定如何估算、预算、管理、监督、控制项目成本的过程。
- (2) 估算成本:对完成项目活动所需货币资源,进行近似估算的过程。

(3) 制定预算：汇总所有单个活动或工作包的估算成本，建立批准的成本基准过程。

(4) 控制成本：监督项目状态，以更新项目成本和管理成本基准变更的过程。

估算不但对于项目计划和管理是非常有用的，而且在其他方面也至关重要。

厂家与客户之间的合同，取决于成本和时间表的估算值。如果没有这些估算值，那么客户也就丧失了对建议书进行评判的基础。合同的条款一般都会包括成本和时间表的估算值，而这些条款一般都被认为是固定的。所以，厂家就必须获得准确的估算值，因为任何低估的情况都会对供应商造成损失，也就是说，客户将只支付作为合同基础的估算成本，而不会承担任何额外开支。因此，良好的估算对于任何软件企业或组织而言都是极其重要的，特别是对于那些为第三方开发软件产品的企业更是如此。

工作量的估算是任何项目管理中最困难和最重要的活动之一。软件成本管理如图 3-1 所示。

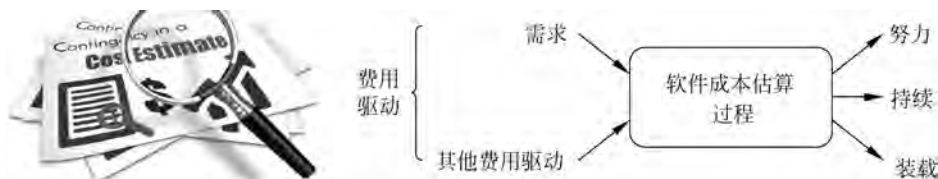


图 3-1 软件成本管理

虽然说工作量和时间表的估算值都是制订计划所必不可少的，但是，如果我们能够知道工作量的估算值，那么时间表的估算就会变得更加容易。所以，软件项目中的重点主要在于工作量的估算。如果不能估算出执行一个项目到底需要付出多少工作量和时间，那么也就不可能进行有效的项目计划和管理。

目前，被人们所普遍认可的观点是，软件的规模是确定到底需要付出多少工作量才能创建出软件的主要因素。但是，当在构想一个项目时，实际软件的规模是未知的，而软件本身也是不存在的。所以，如果需要采用规模作为工作量估算模型的输入信息，那么，在一开始进行估算时，就首先必须估算出规模。换言之，这种方法可将工作量估算问题简化为规模估算问题。有了规模的估算值，通常也就能采用某种公式来获得工作量的估算值了。

估算的基本活动是，以数值的形式获知正在开发的软件的某些特性的输入信息，然后，再利用这些输入信息数值，来估算出项目的工作量。一个软件估算模型，可以精确地定义项目需要哪些数值以及应该如何利用这些数值来计算出工作量。从一定意义上说，工作量估算模型也就是一个获得某些输入信息（如软件特性的数值等），而后再输出工作量估算值的函数。软件生命周期各个阶段所做的估算和详细程度是不同的，在每次估算中如规模、工作量等估算的先后顺序是软件过程改进的重要内容。

3.2 项目成本估算的内容

为了使开发项目能够在规定的时间内完成，而且不超过预算，成本估算的管理控制是关键。

软件开发成本估算主要指软件开发过程中所花费的工作量及相应的代价。不同于传统



视频讲解

的工业产品,软件的成本不包括原材料和能源的消耗,主要是人的劳动的消耗。另外,软件也没有一个明显的制造过程,它的开发成本是以一次性开发过程所花费的代价来计算的。因此,软件开发成本的估算应从软件计划、需求分析、设计、编码、单元测试、集成测试到认证测试,整个开发过程所花费的代价作为依据的。

同样,软件项目开发的成本估算过程也不是一蹴而就的,这也许与传统的工业产品生产过程的成本估算过程相似,但因为软件项目的开发成本主要在人力成本上,对人力成本的估算也是软件项目开发成本估算的主要内容,而人力成本主要以工作量或以时间计费,所以先要对软件规模、工作量、开发进度等进行估计,这些过程可以利用历史项目数据作为参考,完成上述步骤后再结合现有成本数据就可以进行成本估算,成本估算不仅是在项目开发工作之前进行,为了保证成本估算结果的准确性,在软件项目过程中也要进行成本估算,可以迭代进行估算过程,如图 3-2 所示。

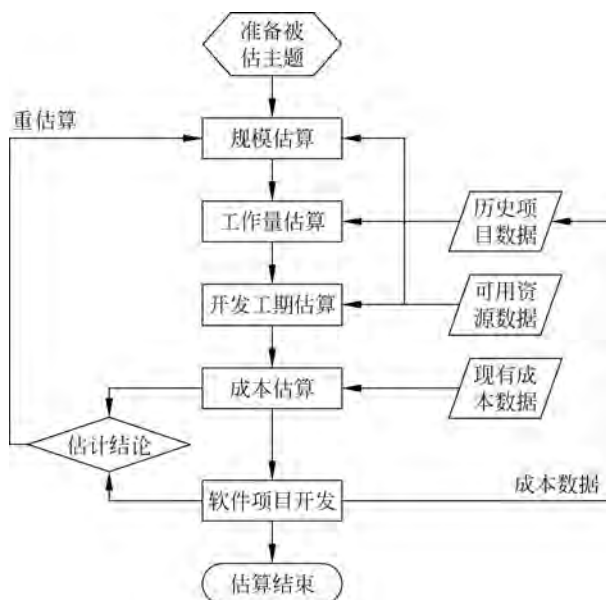


图 3-2 估算流程图

成本估算的过程是在确定被估算主题之后,参照历史项目数据,先进行规模估算、工作量估算、开发工期估算,这些过程是后来进行成本估算的准备过程,在成本估算之后,再将实际软件项目开发成本与估算的成本进行比较,选择是否需要重新估算,并把实际软件项目开发结果数据作为下一次估算的历史数据。



视频讲解

3.3 规模估算

衡量软件规模最常用的单位是源代码行数(Line of Code, LOC)和功能点数(Function Point, FP)。LOC 是指所有可执行的源代码行,包括可交付的工作控制语言(Job Control Language, JCL)语句、数据定义、数据类型声明、等价声明、输入/输出格式声明等。

规模估计的方法有德尔菲方法、功能点估计方法、PERT 估计法、类比估算法等,由于篇

幅有限,本章只介绍德尔菲法和类比估算法。

3.3.1 德尔菲法

德尔菲法(Delphi Method)也称专家调查法,由美国兰德公司(Rand)创始实行,本质上是一种反馈匿名函询法。德尔菲这一名称,起源于古希腊有关太阳神阿波罗的神话,传说中阿波罗具有预见未来的能力。1946年,兰德公司首次用这种方法进行预测,后来该方法被迅速广泛采用。

该流程是在对所预测的问题征得专家的意见之后,进行整理、归纳、统计,再匿名反馈给各专家,再次征求意见,再集中,再反馈,直至得到一致的意见,如图3-3所示。为保证该方法的成功实施,对传统德尔菲步骤进行了扩充和细化。

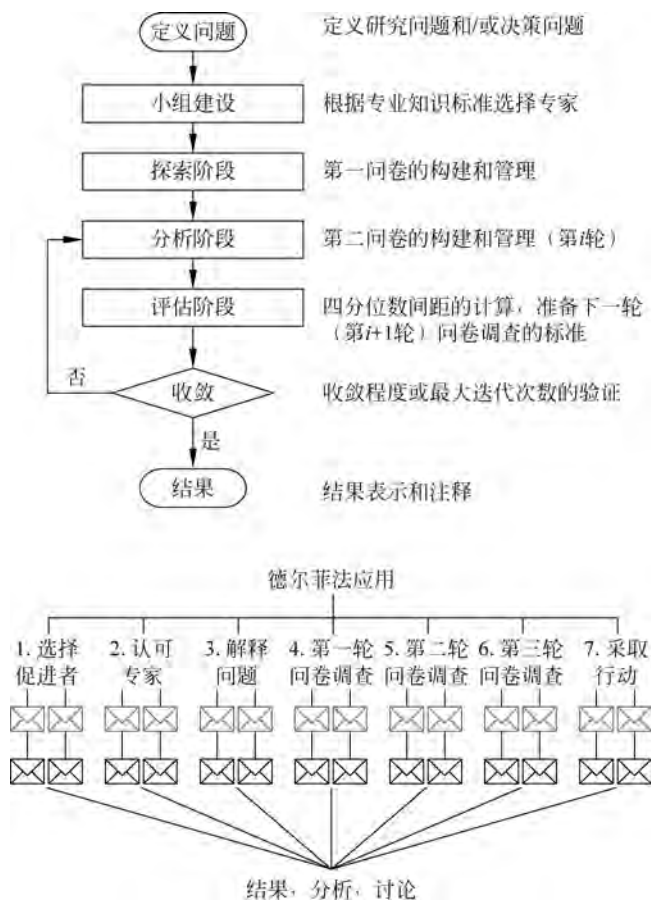


图 3-3 德尔菲法

- (1) 协调员给每位专家一份规格说明书和一张记录估计值的表格。
- (2) 协调员召集小组会议,专家与协调员以及专家之间对估计问题进行讨论。
- (3) 专家无记名地填写表格。
- (4) 协调员对专家填写在表上的估计结果进行小结。

(5) 协调员召集小组会议,让专家对差异很大的估计项进行讨论。

(6) 专家重新无记名地填写表格,该过程要适当地重复多轮。

如图 3-3 所示,德尔菲法分别将所需解决的问题单独发送到各个专家手中,征询意见,然后回收汇总全部专家的意见,并整理出综合意见。随后,将该综合意见和预测问题再分别反馈给专家,再次征询意见,各专家依据综合意见修改自己原有的意见,然后再汇总。多次反复,逐步取得比较一致的预测结果。

德尔菲法依据系统的程序,采用匿名发表意见的方式,专家之间不得互相讨论,不发生横向联系,只能与调查人员发生关系,通过多轮次调查专家对问卷所提问题的看法,经过反复征询、归纳、修改,最后,汇总成专家基本一致的看法,作为预测的结果。这种方法具有广泛的代表性,较为可靠。

表 3-1 给出了估算结果表。

表 3-1 德尔菲法估计结果表

单元	专家 1	专家 2	专家 3	最大值	最小值	平均值	偏差率	接受	最终
A									
B									
C									
D									
E									
估算日期			第 N 轮估算结果			可接受总数		合计	

其中,偏差率的计算方法为:

$$\text{MAX}(\text{最大值} - \text{平均值}, \text{平均值} - \text{最大值}) / \text{平均值} \times 100\%$$

3.3.2 类比估算法

类比估算法也被称作自上而下的估算(Top Down Estimates),是一种通过比照已完成的类似项目的实际成本,去估算出新项目成本的方法。

类比估算法适合评估一些与历史项目在应用领域、环境、复杂度方面相似的项目。约束条件在于必须存在类似的具有可比性的软件开发系统,估算结果的精确度依赖于历史项目数据的完整性、准确度,以及现行项目与历史项目的近似程度。

适合评估一些与历史项目在应用领域、环境和复杂度等方面相似的项目,通过新项目与历史项目的比较得到规模估计。类比法估计结果的精确度取决于历史项目数据的完整性和准确度,因此,用好类比法的前提条件之一是组织建立起较好的项目后评价与分析机制,对历史项目的数据分析是可信的。其基本步骤如下。

(1) 整理出项目功能列表和实现每个功能的代码行。

(2) 标识出每个功能列表与历史项目的相同点和不同点,特别要注意历史项目做得不够的地方。

(3) 通过步骤(1)和(2)得出各个功能的估计值。

(4) 产生规模估计。

软件项目中用类比法,往往还要解决可重用代码的估算问题。

估算可重用代码量的最好办法就是由程序员或系统分析员详细地考查已存在的代码,估算出新项目可重用的代码中需重新设计的代码百分比、需重新编码或修改的代码百分比以及需重新测试的代码百分比。根据这三个百分比,可用下面的计算公式计算等价新代码行:

$$\text{等价代码行} = [(\text{重新设计}\% + \text{重新编码}\% + \text{重新测试}\%) / 3] \times \text{已有代码}$$

例如,有 10 000 行代码,假定 30% 需要重新设计,50% 需要重新编码,70% 需要重新测试,那么其等价的代码行可以计算为:

$$\text{等价代码行} = [(30\% + 50\% + 70\%) / 3] \times 10\,000 = 5000$$

即重用这 10 000 代码相当于编写 5000 代码行的规模。

当然,这 5000 行代码,并不都是旧项目的,还包括新项目的部分。比如,在项目开发过程中,一期做了三个模块,二期又要做三个模块,但其中二期与一期有部分重用的代码,根据这个方法可以估算出实际的规模。

3.4 工作量估算



视频讲解

根据规模估计结果,并定义了项目开发周期和裁剪项目过程后,需要项目过程中各阶段的工作量和总工作量。目前,可以参考的历史数据包括:

- (1) 有历史项目的准确数据;
- (2) 至少有一个历史项目与现有项目规模类似;
- (3) 现有项目将与类似的历史项目采用类似的生命周期、开发过程、开发技术和工具,类似技能和经验的项目成员。同时可以参照业界公布的经验数据。

工作量的估计可采用下面的公式进行:

$$\text{工作量(人} \cdot \text{月)} = [\text{规模(LOC)} / \text{生产率(LOC/人} \cdot \text{天)}] / 22(\text{天/月})$$

参考历史项目数据中各阶段工作量所占百分比,可估算出各阶段工作量:

$$\text{各阶段工作量(人} \cdot \text{月)} = \text{总工作量(人} \cdot \text{月)} \times \text{各阶段工作量百分比}$$

此外还有很多基于算法模型的方法,如普特纳姆算法模型、经验估算模型等,其基本思想是,找到软件工作量的各种成本影响因子,并判定它对工作量所产生影响的程度是可加的、乘数的还是指数的,以期得到最佳的模型算法表达形式。当某个因子只影响系统的局部时,一般说它是可加性的。

例如,如果给系统增加源指令、功能点实体、模块、接口等,大多只会对系统产生局部的可加性的影响。当某个因子对整个系统具有全局性的影响时,则说它是乘数的或指数性的,例如,增加服务需求的等级或者不兼容的客户等。下面是对普特纳姆算法模型、经验估算模型的分析。

3.4.1 普特纳姆模型

这是 1978 年普特纳姆(L. H. Putnam)提出的一种动态多变量模型,通用的形式为:

$$K = L^3 / (Ck^3 \times TD^4)$$

其中,

L : 源代码行数(以 LOC 计)。

K : 整个开发过程所花费的工作量(以人·年计)。

TD : 开发持续时间(以年计)。

Ck : 技术状态常数,它反映“妨碍开发进展的限制”,取值因开发环境而异,见表 3-2。

表 3-2 Ck 的典型值及开发对应环境

Ck 的典型值	开发环境	开发环境举例
2000	差	没有系统的开发方法,缺乏文档和复审
8000	好	有合适的系统的开发方法,有充分的文档和复审
11 000	优	有自动的开发工具和技术

还可以估算开发时间:

$$TD = [L^3 / (Ck^3 \times K)]^{\frac{1}{4}}$$

3.4.2 经验估算模型

经验估算模型又叫 COCOMO 模型(Constructive Cost Model),这是由 TRW 公司开发,Boehm 提出的结构化成本估算模型,是一种精确的、易于使用的成本估算方法。COCOMO 模型主要从两方面来构建:以源代码行(Source Line Of Code, SLOC)统计的软件规模和成本驱动因子(Cost Driver)。这些因素可以被归入产品、平台、人员和项目四方面。

在 COCOMO 模型中按项目开发的不同环境,软件开发项目的总体类型可分为以下三类。

(1) 组织型(Organic): 相对较小、较简单的软件项目。开发人员对开发目标理解比较充分,与软件系统相关的工作经验丰富,对软件的使用环境很熟悉,受硬件的约束较小,程序的规模不是很大(<50 000 行)。

(2) 嵌入型(Embedded): 要求在紧密联系的硬件、软件 and 操作的限制条件下运行,通常与某种复杂的硬件设备紧密结合在一起。对接口、数据结构、算法的要求高。软件规模任意。如大而复杂的事务处理系统,大型、超大型操作系统,航天用控制系统,大型指挥系统等。

(3) 半独立型(Semidetached): 介于上述两种软件之间。规模和复杂度都属于中等或更高。最大可达 30 万行。

按照项目开发的详细程度也可分为三级:基本型、中间型和详细型。所采用的计算通式为:

$$PM = a \times \text{Size}^E \times \prod EM_i, \quad E = B + 0.01 \sum F_i$$

其中:

PM : 以人·月为单位的工作量。

a : 固定常数或称为 Calibration Factor 校准因子, $a \approx 2.94$ 。

Size : 每 1000 代码行(Kilo Source Lines Of Code, KSLOC)计算的规模。

EM_i : Effort Multiplier, 工作量因子。

B : 常数, 为 0 或 1 或其他固定值。

F_i : Scale Factor, 比例因子, 具体的值取决于建模等级(即基本、中等或详细)以及项目的模式(即组织型、半独立型或嵌入型)。表 3-3 给出了 EM_i 的参考取值范围。

表 3-3 EM 的参考取值范围

很低	低	正常	高	很高	极高
0.70	0.85	1.00	1.15	1.30	1.65

3.4.3 功能点分析

功能点分析法是从软件用户的角度来评估一个软件系统的功能, 它将软件的功能分为 5 个基本要素。其中两个表示终端用户的数据需求: 内部逻辑文件和外部接口文件; 另外三个表示用户对数据的获取处理的事务功能: 用户输入、用户输出、用户查询。它们的详细定义如下。

(1) 内部逻辑文件(Internal Logical Files, ILF): 是一个用户可识别的逻辑相关的数据组, 它在应用程序边界内, 由用户输入来维护。它可能是某个大型数据库的一部分或是一个独立的文件。

(2) 外部接口文件(External Interface Files, EIF): 是一个用户可识别的逻辑相关的数据组, 但仅仅是起参考的作用, 且数据完全存在于软件边界之外, 由另一个应用程序进行维护, 是另一个应用程序的内部逻辑文件。

(3) 用户输入(External Inputs, EI): 来自软件外部的数据输入, 可以是控制信息, 也可以是事务数据输入。如果是事务数据, 它必须维护一个或多个内部逻辑文件。也就是说, 那些最后没有保存的中间计算结果和消息发送, 都不算作数据输入单元。输入数据可来自一个数据输入屏幕或其他应用程序。

(4) 用户输出(External Outputs, EO): 是“经过处理”的数据, 由程序内部输出到外部。这里“经过处理”是指其区别于用户查询数据, 是将一个或多个 ILF、EIF 中取出数据经过一定的组合、计算、总结后得出的输出数据。

(5) 用户查询(External Inquiries, EQ): 是一个输入输出的组合过程, 从一个或多个 ILF、EIF 中取出数据输出到程序外部。其中的输入过程不更新任何 ILF, 输出过程不进行任何数据处理。

对软件项目进行估算的有效性和准确性, 取决于所掌握的有关项目的原始资料的完备性。这些原始资料包括需求说明书、系统规格说明书或者软件需求说明书等。从这些原始资料中可分析得出以上 5 类要素。如果以上 5 类要素的数据不准确, 将直接影响到评估的结果。

3.4.4 功能点计算

软件功能计算中一旦估算出应用程序中每个功能要素的数量后, 就可以将每个计数与

一个复杂度值(加权因子)相乘,最后进行合计,算出一个初步的总的功能点数(Function Points Count,UFC)。表 3-4 给出了复杂度加权因子表。

表 3-4 功能要素复杂度加权因子表

功 能	复 杂 度		
	低	平 均	高
外部输入数(EI)	3	4	6
外部输出数(EO)	4	5	7
外部查询表(EQ)	3	4	6
内部逻辑文件数(ILF)	7	10	15
外部接口文件数(EIF)	5	7	10

例如,假设每个功能要素的复杂度都是平均的。一个由 25 个数据登记表、5 个接口文件、15 个报告、10 个外部查询和 20 个逻辑内部表单组成的系统,其功能点为:

$$UFC=25\times 4+5\times 7+15\times 5+10\times 4+20\times 10=450$$

每个功能要素的复杂度可通过表 3-5 进行分析判断。

表 3-5 功能要素复杂度判别表

ILF(内部逻辑文件)和 EIF (外部接口文件)				EO(用户输出)和 EQ (用户查询)				EI(用户输入)			
记录 单元	数据单元			文件 类型	数据单元			文件 类型	数据单元		
	1~19	20~50	51+		1~5	6~19	20+		1~4	5~15	16+
1	低	低	平均	0 或 1	低	低	平均	0 或 1	低	低	平均
2~5	低	平均	高	2~3	低	平均	高	2~3	低	平均	高
6	平均	高	高	4	平均	高	高	4	平均	高	高

从表中可以看出,EI(外部输入)、EO(外部输出)和 EQ(外部查询)是由文件类型和数据单元的数量决定的。而 ILF(内部逻辑文件)和 EIF(外部接口文件)则是由记录单元和数据单元决定的。通过上面的二维表即可确定各个功能要素的复杂度是低、平均,还是高。

表中三种数据项定义如下。

(1) 记录单元类型(Record Element Type,RET): 指在 ILE 或 EIF 中,用户可识别的数据域的子集,可以通过检查数据中的各种逻辑分组来识别它们。例如,一个客户文件包括客户姓名、地址等个人信息,以及客户的各种信用卡和卡号。一个客户一般有多张信用卡,信用卡需同客户信息相连才有意义。因此,这个客户文件含有两个记录单元:客户信息和信用卡信息。

(2) 文件引用类型(File Type Referenced,FTR): 指在一个事务过程中所引用到的各种文件,可以是内部逻辑文件,也可以是外部接口文件。

(3) 数据单元类型(Data Element Type,DET): 是用户可识别的无递归、不重复的信息单元。DET 是动态的,而非静态的,读自于文件,或由 FTR 的数据单元创建。另外,一个 DET 也可以是对一个事务处理过程的唤醒,或是事务的有关信息。如果 DET 存在递归或

重复,只计算其中的一个。如上例中的客户姓名、地址就是两个 DET。在可视化编程中,用于唤醒事务处理的添加、修改按钮,也算 DET。

(4) 确定技术复杂度因子(Technology Complexity Factor, TCF): 算出功能点总数 UFC 后,还需要根据项目具体情况对技术复杂度参数进行调整。如表 3-6 所示,技术复杂度一共考虑了 14 个调节参数。

表 3-6 技术复杂度因子

序 号	调 节 参 数	描 述
1	E1	数据通信(Data Communications)
2	E2	软件性能(Performance)
3	E3	可配置性(Heavily Used Configuration)
4	E4	事务效率(Transaction Rate)
5	E5	实时数据输入(Online Data Entry)
6	E6	用户界面复杂度(End User Efficiency)
7	E7	在线升级(Online Update)
8	E8	复杂运算(Complex Processing)
9	E9	代码复用性(Reusability Ease)
10	E10	安装简易性(Installation Ease)
11	E11	操作方便性(Operations Ease)
12	E12	跨平台要求(Multiple Ease)
13	E13	可扩展性(Facilitate Change)
14	E14	分布式数据处理(Distributed Functions)

各个复杂度参数的取值范围为 0~5,表示该项对功能点总数的影响从没有到极高。各个参数默认值为 0,也就是该项不影响功能点调整。

每个参数都是对总功能点数的线性调整,设 E_i 为根据 14 方面的调节参数对软件系统的影响程度,则功能点技术复杂度因子为:

$$TCF = 0.65 + 0.01 \times \sum E_i, \quad (i = 1, 2, \dots, 14)$$

$$E_i \in (0, 5), \text{ 则: } TCF \in (0.65, 1.35)$$

工作量指在软件项目建设过程中需要投入的人力和时间,一般用人·月数进行度量。项目建设阶段一般可分为:开发阶段、实施阶段、运行维护阶段。故工作量需分阶段进行估算。

$$\text{工作量} = \text{开发工作量} + \text{实施工作量} + \text{维护工作量}$$

由于在软件项目开发过程中,因需求变更导致工作量改变的情形不可避免,故可分别在立项阶段进行工作量预算,在项目完成阶段进行工作量核算。

3.4.5 开发阶段工作量估算

开发工作量是计算实施阶段和维护阶段工作量的基础。主要有两种估算方法。

1. 功能点估算法

该方法主要是依据软件项目的功能需求来评估开发工作量。

通过分析系统需求计算项目规模(功能点数),再乘以各阶段完成每个功能点所需要投入的人工时(开发成本系数),就可计算出完成项目所需要的人·月数。适用于立项阶段需求分析比较详细的项目或者用于项目完成阶段的最终工作量估算。

开发工作量 $D(\text{人} \cdot \text{月}) = (\text{项目功能点 } FP \times \text{开发成本系数 } k / H / W)$

其中, H 是指国家规定的一天工作时数, W 指一个月工作天数。

功能点 FP 的估算采用——软件项目功能点估算法。

开发成本系数 k 的大小主要是考虑项目的非技术难度,如开发周期、协调难度、业务的复杂程度、需求的不确定性等因素。如表 3-7 所示为根据对实际数据的测算,开发成本系数 k 的取值范围。

表 3-7 开发成本系数 k 取值范围

功能点数(FP)	开发成本系数(人工时/ FP)
≤ 3000	3.5~4.0
$3000 \leq FP \leq 8000$	4.0~4.5
> 8000	4.5~5.0

针对个别项目,如果有特殊情况(如某些用户业务的特殊要求是一般项目中从未出现过的、开发人员需要到用户现场开发等),则经专业咨询机构或者专家评估,开发成本系数可以超出此范围上限的限制。

2. 任务估算法

任务估算法,是把软件项目功能分解为若干个相对独立的任务,再分别估计完成每个任务需要的人员搭配比例及投入时间,每个人员的工作量之和就是该任务的工作量。最后,将各个任务的工作量累加起来就得出软件项目的总工作量。该方法适用于立项阶段的工作量估算。

依据软件工程的观念、国内软件开发行业的惯例及经验值,软件开发工作可分为:设计、编码、测试。

设计各个岗位人员工作量可基于以下标准计算。

- (1) 以程序员的工作量为标准。
- (2) 高级程序员的工作量为标准工作量的 1.5 倍。
- (3) 系统分析员的工作量为标准工作量的 2.5 倍。
- (4) 测试工程师的工作量为标准工作量。
- (5) 高级测试工程师的工作量为标准工作量的 1.5 倍。
- (6) 项目管理人员的工作量为标准工作量的 3 倍。
- (7) 市场营销人员的工作量为标准工作量。
- (8) 技术支持工程师的工作量为标准工作量。
- (9) 文秘的工作量为标准工作量的 0.5 倍。

例如,完成某个任务的人员投入和时间需求如表 3-8 所示,则其工作量为 60.5 人·月。

表 3-8 某任务的工作量估算表

开 发 阶 段	投入人员情况	时间/月	工作量/人·月
需求分析	系统分析员 2 人	2	$2 \times 2 \times 2.5 = 10$
系统设计	系统分析员 1 人	2	$1 \times 2 \times 2.5 = 5$
	高级程序员 2 人	2	$2 \times 2 \times 1.5 = 6$
编码	高级程序员 2 人	1	$2 \times 1 \times 1.5 = 3$
	程序员 4 人	1	$4 \times 1 \times 1 = 4$
测试	测试工程师 4 人	2	$4 \times 2 \times 1 = 8$
项目管理	项目管理人员 1 人	7	$1 \times 7 \times 3 = 21$
文案工作	文秘 1 人	7	$1 \times 7 \times 0.5 = 3.5$
合计: 60.5(人·月)			

3.4.6 实施阶段工作量估算

软件项目的实施范围因项目而异。有些项目只实施一个单位,有些需要实施多个单位,有些甚至需要全市、全省甚至全国实施。所以,实施阶段的费用也会有很大的差异,甚至有的项目会出现实施费用超过开发费用的情形。

(1) 实施阶段的工作量可依据开发阶段工作量、实施系数来计算。

$$\text{实施工作量(人·月)} = \text{开发工作量 } D \times \text{实施系数 } s$$

根据项目是集中式实施还是分布式实施,实施系数 s 的取值有所不同。

集中式实施的项目实施系数 s 与“用户数”相关。设 n 为用户数,一般情况下:

当 $0 < n \leq 100$ 时, $s = 0.2$;

否则, $s = 0.2 + ((n - 100) / 100) \times q$ (四舍五入取两位小数)。 q 是调节因子,取值范围为 $0.03 \leq q \leq 0.05$,具体取值依项目实施难度而定。

(2) 分布式实施的项目。

实施系数 s 与“实施单位(点)数”相关。设 n 为需要实施的单位(点)数,一般情况下:

$$s = 0.2 + (n - 1) \times q$$

其中, q 是调节因子,一般取值范围为 $0.08 \leq q \leq 0.15$,具体取值依项目实施难度而定。

(3) 个别项目如果对实施有特殊要求(这些特殊要求是一般项目中从未出现过的或有本地化开发工作的),或者实施环境、条件、难度等方面因素的影响,则经专业机构或者专家评估,实施系数可以超出此范围上限的限制。

(4) 如果软件项目是系统集成项目中的一部分,实施时需要整体考虑,则可将实施费抽出另算。一种是将软件实施费并入到整个集成项目的实施费用中,另一种就是在软件实施费中加入项目集成的实施费用。

3.4.7 维护阶段工作量估算

软件项目通过验收,交付使用后,需进行一年的系统维护。维护内容包括:运行管理、系统平台维护、应用软件维护、数据维护等。根据不同的用户要求,系统维护服务可分为以下两种情形。

1. A 级

软件企业派出技术人员常驻用户处,解决日常运行中发生的问题,则其工作量由派驻人员的数目和派驻的时间决定。

$$\text{软件(系统)维护工作量} = \text{派驻的人员数} \times \text{时间(月)}$$

2. B 级

软件企业在国家规定的正常工作时间,按双方约定的条件和时间到达现场,且每月(或定期)派技术人员到现场进行软件(系统)性能调试,使之运行处于良好状态。则 B 级的维护工作所需工作量依据开发工作量、实施工作量、维护系数来计算。

$$\begin{aligned} \text{运行维护工作量(人} \cdot \text{月)} &= (\text{开发工作量} + \text{实施工作量}) \times \text{维护系数 } w \\ &= (\text{开发工作量} + \text{开发工作量} \times \text{实施系数 } s) \times \text{维护系数 } w \\ &= D \times (1 + s) \times w \end{aligned}$$

维护系数 w 取值范围为 0.15~0.20,具体取值依据项目维护难度而定。

针对个别项目,如果对维护有特殊要求(这些特殊要求是一般项目中从未出现过的),则经专业机构或者专家评估,维护成本系数可以不受此限制。

系统后期维护:系统运行一年之后的系统维护,需另行签订系统维护合约。为了有利于保证用户的利益和扶植软件企业,在维护工作范围不变的前提下,如果新的维护合同的维护费用不超过上一年度维护金额的 115%,则用户应该和原开发商直接签订维护合同,否则可进行招投标并确定新的维护合同的项目承担单位。



视频讲解

3.5 开发工期估算

项目估计的第三步,是根据工作量制订项目计划,包括人员安排、工作量分解、开始和完成时间等。可以根据自己的历史数据或行业模型决定所需的人力资源并落实到项目计划。

如果没有以上信息,可以使用估计模板中的方法来进行计算:完成工作量估计后,根据现在的资源情况,确定在每个阶段投入的资源,根据相关的计算方法计算出进度。

$$\text{工期(天)} = \{\text{工作量(人} \cdot \text{天)} \times (1 - \text{并行工作比例})\} + \text{工作量} \times \text{并行工作比例} / \text{投入资源(人)}$$

另外,若考虑开发时间则可以把学习曲线考虑进去:产量以倍数增加时工作效率以固定比率提高,这一比率称为学习率。

例如,一个员工在进行一项工作第一次需要 10 个工作日,第二次需要 8 个工作日,则其学习率为 80%,当次数继续增加时,即重复 4 次,可以期望第 4 次工作时间为 6.4 天,第 8 次只需要 4 天。

那么,得到第 n 次完成产出需要的时间公式为:

$$T_n = T_1 n^{-r}$$

其中:

T_n : 第 n 次完成产出所需要的时间。

T_1 : 第 1 次完成产出所需要的时间。

n : 所完成产出的次数。

$r = \lg(\text{学习率}) / \lg 2 = \log_2(\text{学习率})$ 。

开发 n 个类似软件项目所需要的总时间 T 为:

$$T_{\text{平均}} = \frac{T_1}{1-r} n^{1-r}$$

平均开发时间为:

$$T_{\text{平均}} = \frac{T_1}{1-r} n^{1-r}$$

那么,学习曲线如图 3-4 所示。

由于新项目与历史项目在应用领域、环境和复杂度方面具有相似性,所以可以把学习曲线考虑进去,结合学习曲线法进行估算。

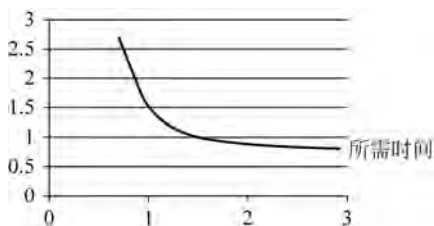


图 3-4 学习曲线



视频讲解

3.6 成本估算方法

成本估算是指对完成项目各项活动所必需的各种资源的成本做出的估算。

估算计划活动的成本,涉及估算完成每项计划活动所需的资源,包括人力资源、设备、材料、服务、设施和特殊条目,如通货膨胀准备金和应急准备金等的近似费用。在估算时需考虑估算偏差可能的原因如风险等。计划活动成本估算是针对完成计划活动所需资源的可能费用进行的量化评估。

将软件项目开发过程与有形的产品进行类比,软件项目开发总成本可以分为可确定成本与变动成本,单位总成本可分为单位可确定成本与单位变动成本等,其关系如下。

$$\text{总成本} = \text{总可确定成本} + \text{总变动成本}$$

$$\text{单位总成本} = \text{单位可确定成本} + \text{单位变动成本}$$

其中,固定成本表示能够确定的成本,如软件开发工具平均费用分摊、硬件资源平均费用分摊、数据库费用、团队建设费等。软件项目开发过程成本主要为变动成本,变动成本是不可确定的其他成本,如培训费用、管理费用、差旅费等。这与软件项目开发过程工作量与进度或开发工期有关,如:

$$\text{人力成本} = \text{工作量(以人} \cdot \text{年或人} \cdot \text{月计)} \times \text{员工平均工资}$$

$$\text{培训费用} = \text{开发工期} \times \text{培训单价} \times \text{培训频率}$$

$$\text{管理费用} = \text{开发工期} \times \text{管理费用单价}$$

$$\text{差旅费用} = \text{开发工期} \times \text{平均差旅次数} \times \text{平均差旅单价}$$

那么,可以得到总的成本为:

$$\begin{aligned} \text{总成本} &= \sum \text{单位总成本} = \sum \text{单位可确定成本} + \sum \text{单位变动成本} \\ &= \sum \text{单位软件开发工具平均费用分摊} + \sum \text{单位硬件资源平均费用分摊} + \\ &\quad \sum \text{单位人力成本} + \sum \text{单位培训费用} + \sum \text{单位管理费用} + \sum \text{单位差旅费用} \end{aligned}$$

3.6.1 咨询费

咨询费是指软件项目立项前期,请专业机构或者专家进行技术咨询、可行性分析、需求

分析,造价评估、方案设计、项目招标代理等方面工作所发生的费用。

该部分费用可根据项目预计投入的建设费按照一定比例计取,也可以根据所投入的人·月数进行计取,此外还可以由双方协商确定。如表 3-9 和表 3-10 所示,在招标活动中,公证处对全过程进行现场公证并对采购合同进行公证,公证费按照国家规定标准计算。

表 3-9 软件行业咨询取费标准

收费项目	收费基数	基准费率/%					
		≤100 万	101~300 万	301~500 万	501~1000 万	1001~3000 万	>3000 万
需求分析、可行性分析、系统设计等	项目预投入费	8.3	7.8	7.3	6.7	5.4	4.5
估价	项目预投入费	3.6	3.0	2.5	2.2	1.8	1.5
招标代理	中标金额	1.0	0.8	0.7	0.55	0.35	0.3
技术咨询	每人每日	1000~1500 元					

表 3-10 公证服务取费标准

标的额 m /万元	≤2	$2 < m \leq 5$	$5 < m \leq 10$	$10 < m \leq 50$	$50 < m \leq 100$	$100 < m \leq 200$	$200 < m \leq 300$	$300 < m \leq 400$	>400
费率/%	1	0.8	0.6	0.5	0.4	0.3	0.2	0.1	0.05

注:

按表 3-9 计费不足 1000 元的,按 1000 元收费。

按表 3-10 计费不足 200 元的,按 200 元收费。

技术咨询按耗用工时(日)计费,为完成委托任务发生的差旅、交通费由委托方另行支付。

招标代理收费和公证服务收费按差额定率累进法计算。

如某招标代理业务中标金额为 600 万元,计算招标代理费如下。

$$100 \text{ 万元} \times 1.0\% = 1 \text{ 万元}$$

$$(300 - 100) \text{ 万元} \times 0.8\% = 1.6 \text{ 万元}$$

$$(500 - 300) \text{ 万元} \times 0.7\% = 1.4 \text{ 万元}$$

$$(600 - 500) \text{ 万元} \times 0.55\% = 0.55 \text{ 万元}$$

则合计收费: $1 + 1.6 + 1.4 + 0.55 = 4.55$ 万元。

3.6.2 建设费

建设费包括支付给软件开发商的进行软件开发、实施、维护等方面工作的费用。主要依据工作量(完成该项目需要投入的人力,以人·月度量)和人·月成本进行估算。

$$\begin{aligned}\text{建设费} &= \text{开发费} + \text{实施费} + \text{运行维护费} \\ &= (\text{开发工作量} + \text{实施工作量} + \text{运行维护工作量}) \times \text{人} \cdot \text{月成本}\end{aligned}$$

3.6.3 服务费

1. 验收测试费

软件项目验收是一个运行环境复杂、技术难度较高、评价体系抽象的过程。

该项目验收除经过专家评审外,还应进行相应验收测试,只有两者结合才能为信息化项目验收和鉴定提供定性、定量的科学依据,才能做出较为客观准确的验收和鉴定结论。软件项目的验收测试是根据项目的特点(功能、技术需求和大小等)以及项目投入,按照评价软件质量的功能性、易用性、可靠性、可维护性、可移植性、效率、文档等7个特性进行特性裁减,分为功能确认测试和项目验收测试。

1) 功能确认测试

项目对象:省、市级信息化建设项目包括电子政务建设项目验收,各种渠道申报的与软件相关的科技项目的验收和科技成果鉴定项目。

测试内容:根据申报或鉴定的技术条款和软件操作手册及被测软件运行确定测试内容,一般只覆盖软件的功能性、易用性和文档。主要判断被测系统是否完成合同要求的功能及相关特性。

收费标准:8000~10 000 元。

2) 项目验收测试

项目对象:各类信息化建设项目包括电子政务建设项目应用发布之前的验收,各种渠道申报的与软件相关的科技项目的验收和科技成果的鉴定项目以及合同中的条款覆盖效率和可移植性等特性要求的项目。

测试内容:在模拟或实际环境下测试被测系统是否实现了用户需求,是否达到了国家标准的相关要求。依据用户需求分析、合同的技术条款、国家标准的特性要求、软件操作手册和被测软件运行确定测试内容。

收费标准:验收测试费=建设费 D × 各测试项费率之和 × 调节系数 t

各测试项的费率及收费调节系数取值如表 3-11 和表 3-12 所示。

表 3-11 验收测试项费率

序号	测试项	子 特 性	费率/ $a\%$
1	功能性	功能点 ≤ 100	$a \geq 2.8$
		功能点 > 100	$a \geq 3$
2	易用性	易理解性	$a \geq 0.07$
		易学性	$a \geq 0.06$
		易操作性	$a \geq 0.07$
3	可靠性	成熟性	$a \geq 0.2$
		容错性	$a \geq 0.2$
		易恢复性	$a \geq 0.1$

续表

序号	测试项	子 特 性	费率/ $a\%$
4	维护性	易改变性	$a \geq 0.07$
		稳定性	$a \geq 0.07$
		易测试性	$a \geq 0.06$
5	可移植性	一个环境下测试	$a \geq 0.2$
		多个测试环境, 测试环境数 n	$a \geq 0.2 + (n-1) \times 0.1$
6	效率	一般的效率指标	$a \geq 1$
		并发用户数 ≤ 50 , 测试脚本数 ≤ 3	$a \geq 1$
		每增加 50 个以内用户数或 3 个以下测试脚本数	a 递增 0.5
7	文档	用户文档	$a \geq 0.1$
		技术合同	$a \geq 0.05$
		需求规格说明书	$a \geq 0.1$

表 3-12 调节系数 t 取值范围

序 号	项目建设费 D /万元	收费折扣系数(t)
1	$D \leq 200$	≥ 1
2	$200 < D \leq 500$	≥ 0.98
3	$500 < D \leq 1000$	≥ 0.96
4	$1000 < D \leq 2000$	≥ 0.95
5	$2000 < D \leq 5000$	≥ 0.93
6	$5000 < D \leq 10\,000$	≥ 0.92
7	$D \geq 10\,000$	≥ 0.90

关于项目验收测试需要注意:

(1) 影响项目验收测试费用的因素一个是项目的大小,另一个是所选择的测试项。被选测试项多少决定测试费率 a ,项目大小决定收费调节系数 L 。

(2) 根据项目特点针对软件各个特性进行选择测试,测试费率为所选择软件特性测试费率 a 各项之和。

(3) 根据项目大小采取项目建设费越高费率越低原则进行调节。

(4) 项目验收测试最低收费为:8000 元(不含负载压力测试),2 万元(含负载压力测试)。

2. 工程监理费

软件项目监理收费既考虑了信息系统软件项目的特点,又参照了其他监理行业的收费标准、收费方式。一般地,可按照项目建设费(或合同价格)的一定百分比收费,其收费比率主要根据项目的规模、阶段、内容、复杂程度及监理成本等多方面因素综合计算。计算公式如下。

$$\text{监理费} = \text{建设费 } D \times \text{基本费率 } a \times \text{地域调整系数 } d \times \text{工期调整系数 } e$$

(1) 基本费率 a 根据项目建设费的规模进行调整。取值范围如表 3-13 所示。

表 3-13 监理基本费率 a 取值范围

序 号	项目建设费 D /万元	费率 a /%
1	$D \leq 200$	>12
2	$200 < D \leq 500$	>9
3	$500 < D \leq 1000$	>7
4	$1000 < D \leq 2000$	>6
5	$2000 < D \leq 5000$	>5
6	$5000 < D \leq 10\,000$	>4
7	$D > 10\,000$	>3

(2) 鉴于软件项目实施时分布的地域会有所不同,因此,监理的费率应在基本费率的基础上考虑地域的因素。地域调整系数 d 取值范围如表 3-14 所示。

表 3-14 地域调整系数 d 取值范围

序 号	地 域 范 围	地域调整系数
1	集中实施	1
2	地市范围	$1 \sim 1.2$
3	全省范围	$1.2 \sim 1.5$
4	全国范围	$1.5 \sim 2$

(3) 鉴于软件项目工期长短不一,因此,监理的费率应在监理的基本费率基础上考虑工期的因素。工期越长,系数越大。工期调整系数 e 取值范围如表 3-15 所示。

表 3-15 工期调整系数 e 取值范围

序 号	工程工期 T /年	工期调整系数 e
1	$T \leq 1$	$e > 0.9$
2	$1 < T \leq 2$	$e > 1.1$
3	$T > 2$	$e > 1.4$

(4) 其他。

对于非监理原因造成工程延期而产生的监理附加工作,监理单位有权获得监理附加报酬。

$$\text{监理附加报酬率} = \text{监理费} \times \text{附加工作月数} / \text{合同规定月数}$$

对于项目结束后的维护,其监理收费由用户单位和监理单位协商解决。

本参考标准未作规定的,可参考国家相关标准。

3. 数据处理费

项目中如含有大量档案、数据需要录入、处理,则需要考虑相应的数据处理服务费。收费标准可以根据所需要处理的资料的页数核计收费。

一般情况下,单纯的数据录入收费标准为 0.3~0.5 元/页。特殊要求的数据处理可依据合同约定。

4. 附加费

如果用户需要软件开发商提交源代码,则必须支付相应的知识产权费;如果所开发的项目是涉密项目,则需额外再支付给软件开发商保密费。这些费用的计算均与软件开发工作量相关,也就是与项目建设费相关,可按照项目建设费的一定比例计取,或者双方协商。

5. 需求变更估算

由于软件开发过程中,用户的需求有可能不断变化,从而导致开发工作量的变化,费用追加。故在立项阶段即要请专业机构或者专家对需求变更的风险性进行评估,以便在做项目预算时留出足够应付需求变更的经费。

项目需求变更一般发生在项目建设过程中,立项阶段的咨询服务不受需求变化的影响。但验收测试和工程监理工作量会随着需求变化而加大,所以需求变更费为:

$$\text{需求变更费} = (\text{建设费} + \text{验收测试费} + \text{监理费}) \times \text{需求变更风险系数 } f$$

风险系数 f 可依据以下因素确定。

(1) 项目的成熟度:如果是新项目,则开发过程中出现需求变更的可能性很大,需求变更幅度大,风险系数就高;如果是成熟项目,或者已经有过案例的项目,需求变化的可能性较小,即使有变化,幅度也不会太高,则风险系数就低。

(2) 项目的规模大小:如果项目规模小,需求容易确定,变更概率就小,反之就大。

(3) 用户业务的稳定性和管理的规范性:用户单位业务的变化和业务流程的调整,都有可能带来开发过程中需求的变化。

前期项目需求分析、系统设计的规范性和完善性:前期的需求分析是否全面到位、系统设计得是否规范和细致,会影响到开发过程的需求变化率。

小结

本章主要列举了两种规模估算方法:德尔菲法和类比估算法,但是这两种方法都具有局限性,本章介绍这两种方法的目的也是希望读者在做软件项目成本估算的时候可以结合这两种方法综合进行估算,即先用类比估算方法划分每个功能单元,组织德尔菲估算方法小组,对每个功能单元进行估算,同时要标识出每个功能列表与历史项目的相同点和不同点,特别要注意历史项目做得不够的地方,还要解决可重用代码的估算问题。

介绍了两种工作量估算模型:普特纳姆模型和经验估算模型。当然,普特纳姆模型也可以用于软件工期估算,实际上普特纳姆模型也是必须以软件工期估算为前提的。经验估算模型也称为 COCOMO 模型,这种模型有比较成熟的理论系统,从很多使用过 COCOMO 模型的人员评价来看,这种模型也能比较准确地对工作量进行估算,但是需要考虑的因素较多。建议的做法是使用 COCOMO 估算或是其他有效的估算方法,在重估算的时候可以考虑使用普特纳姆模型。

对于开发工期估算,除了介绍传统的估算公式外,还介绍了考虑使用学习曲线的方法进行估算,由于软件项目的特殊性,对人员和技术的依赖比较大,使用学习曲线方法进行估算可以增加估算结果的准确性。

最终,要进行软件项目开发成本的估算。由于有软件规模估算、工作量估算、软件开发

工期估算为前提在进行成本估算的时候还要考虑能产生成本的各个因素。同时适时地进行重估算的步骤也是有价值的。

思考题

1. 简述项目成本估算的重要意义。
2. 详细说明项目成本估算的内容和流程。
3. 简述规模估算方法。
4. 简述工作量估算方法。
5. 简述开发工期估算方法。
6. 简述成本估算方法。