

## 第3章



# 《软件研发成本度量规范》标准解读

---

电子行业标准 SJ/T 11463—2013《软件研发成本度量规范》于 2013 年 10 月正式发布,同年 12 月正式实施。SJ/T 11463—2013 规定了软件研发成本度量的方法、过程及原则,用于规范软件研发项目各利益相关方在项目成本度量方法上达成一致。本章按照 SJ/T 11463—2013 的章节顺序对标准进行详细解读,帮助读者更好地理解和应用。

## 3.1 标准概述及结构说明

### 3.1.1 概述

在软件研发成本度量(包括估算与测量)方面,对于软件规模本身的评价是首要任务。根据软件行业的实践,目前评价软件规模的方法主要分为两种:基于业务视角和基于开发视角。基于业务视角的方法是从用户角度出发,与软件开发技术无关,如:功能点、故事点、用例点、对象点等方法;基于开发视角的方法是从开发者角度出发,如软件源代码行、数据库表、函数数量等方法。

基于开发视角的软件规模评价方法,优点是操作简单、实施容易,但不容易在项目干系人之间达成一致,往往会引起较多分歧。基于开发视角的评价方法虽然在实际工作中也有着普遍的应用,但更多地局限于软件开发团队内部。如果要在

业务部门与开发部门、甲方与乙方等外部组织约定软件开发的工期或费用等关键项目目标,则需要从业务视角出发,对软件项目规模进行标准、一致的评价与估算。而且,在系统初始阶段,用户功能需求是唯一真正可以得到的信息。任何程序大小或代码行数的猜想实际上都是从系统要提供的功能性推演出来的。

表 3.1 展示了几种常用的软件规模度量方法的对比,从表 3.1 中可以看出,综合起来功能点方法最优。

表 3.1 软件规模度量方法对比

分类	比对项目	功能点	对象点	用例点	故事点	代码行
方法有效性	业务价值分析	★★★★	★★	★★	★★	★
	产能分析与评估	★★★★	★★	★★★★	★★	★★
	项目早期估算	★★★★	★★★★	★★	★★	★
	项目中后期估算	★★★★	★★	★★★★	★★	★★
	项目范围管理	★★★★	★★★★	★★★★	★★★★	★★
	团队绩效评价	★★★★	★★	★★	★★	★
	行业基准比对	★★★★	★	★	★	★★
应用难度	方法学习难度	★★	★★★★	★★	★★★★	★★★★
	方法导入成本	★	★	★	★★	★★★★
	方法应用一致性	★★★★	★★	★★	★	★

自 Allan J. Albrecht 在 20 世纪 70 年代末提出功能点方法以来,功能点在软件行业的应用与实践已超过 30 年,在 Albrecht 的功能点模型基础之上,经过进一步的应用与发展,功能点标准演进为 ISO/IEC 14143“信息技术软件度量功能规模度量”系列标准及 IFPUG、Mk II、COSMIC、NESMA、FiSMA 五个具体操作方法的标准,这也是 SJ/T 11463—2013 引入这五种度量标准作为规范性引用文件的基础。表 3.2 从用户角度给出了这五种功能点规模度量方法的比较。

表 3.2 功能点方法比较(用户角度)

名称	应用领域	方法易用性	用户广泛性	可靠性	综合评价
IFPUG	★★★★	★★★★	★★★★★	★★★★	★★★★★
Mk II	★★	★★	★★	★★★★★	★★
COSMIC	★★★★★	★★★★★	★★★★	★★	★★★★
NESMA	★★★★	★★★★★	★★★★	★★★★	★★★★★
FiSMA	★★★★★	★★	★★	★★	★★

从表 3.2 中可以看出, NESMA 功能点方法不但易学易用、快速、经济, 而且容易开发, 便于用户建立自己特有的估算模型。并且, 在这五项国际标准中, 只有 NESMA 方法定义了 3 种应用场景以支持不同粒度的估算, 并且随着项目的进展和需求的完善, 估算者可以不断修正之前的结果, 进行持续的软件度量。当然 NESMA 方法针对不同项目的实际操作性不一定完全适用, 在估算上也存在一些主观性, 它本身也在不断演化过程中, 用户可以不断地进行定制和裁剪, 只要采用统一的计数和计算标准, 就可以很好地运用。

### 3.1.2 标准的结构

SJ/T 11463—2013 规定了软件研发成本度量的方法、过程及原则, 旨在帮助涉及软件研发的各方能够在软件研发成本度量方法上达成一致, 以满足软件产业健康发展的需要。

SJ/T 11463—2013 共分为 6 章和 1 个附录。

标准的前 3 章为标准的必备要素。第 1 章为标准的范围, 第 2 章为标准的规范性引用文件, 第 3 章为标准用到的术语和定义。

第 4 章至第 6 章为标准的主体内容。其中, 第 4 章为软件研发成本的构成以及各个组成部分, 第 5 章为软件研发成本的度量原则和过程, 第 6 章为标准的典型应用场景。

该标准的附录 A 为标准中典型应用的度量要求及过程。

### 3.1.3 标准的范围

#### 【标准原文】

本标准规定了软件研发成本度量的方法及过程, 包括软件研发成本的构成、软件研发成本度量过程、软件研发成本度量的应用。

本标准适用于度量成本与功能规模密切相关的软件研发项目的成本。

**注:** 对于以非功能性需求为主, 或包含大量复杂算法, 或以创意为主的软件研发项目, 在进行成本估算时, 可参考本标准估算软件规模, 并估算除算法研究、高度创意及非功能需求之外的软件研发工作成本; 也可不估算软件规模, 参考本标准描述的方法(如类比法、类推法)和原则直接估算软件研发项目的工作量、工期及成本。

#### 【标准解读】

编制 SJ/T 11463—2013 的主要目的在于明确软件研发成本度量的方法及过

程。因此, SJ/T 11463—2013 主要内容包括软件研发成本的构成(即什么是软件研发成本)、软件研发成本度量过程(即应该依据什么原则、方法和步骤去估算或测量软件研发成本)、软件研发成本度量的应用(即在不同的应用场景使用 SJ/T 11463—2013 的要点是什么)。

在遵循 SJ/T 11463—2013 进行软件研发成本度量,尤其是在使用类比法或方程法进行工作量、成本、工期估算时,通常需要使用历史数据或估算模型。由于行业数据在不断变化,基于行业数据所建立的估算模型每年也会根据统计分析结果进行修订,为了保证行业标准的稳定性,这些数据与模型并没有纳入标准正文或列为附录。读者可通过附录 A 提供的途径获取每年最新行业数据。

由于制定 SJ/T 11463—2013 的目的之一是指导相关组织或个人科学、一致地开展软件研发成本估算活动,而规模估算是进行软件研发成本估算的重要基础,在进行规模度量时, SJ/T 11463—2013 遵循了功能规模度量方法的相关国际标准。因此,标准主要适用于与功能规模密切相关的软件研发项目的成本估算。

以功能性需求为主的项目均可以遵循该标准进行成本估算。对于此类项目,在估算了功能规模后,可以通过引入合理的调整因子(参见 SJ/T 11463—2013 中 5.1.3.2)进行有效的成本估算,也可以对项目中的特殊任务(如某关键算法研究)单独估算或调整。

对于以非功能性需求为主的项目,如果其成本与功能规模依然有很高的相关性,则依然可以遵循该标准进行成本估算。例如,某些行业应用软件,虽然包含大量复杂算法,但因其主要功能均包含复杂算法,且算法的复杂程度大致相当,则此类软件依然可以根据功能规模推算成本(但需要对生产效率基准值进行适当调整)。

而对于性能优化、纯算法研究类的软件项目,则不建议对其功能规模进行度量,但可以采用类推法、类比法等方法估算项目规模、工作量及工期,从而依据该标准对项目成本进行估算。

对于软件研发成本实际结果的测量,所有类型的项目均可遵循 SJ/T 11463—2013。

### 3.1.4 标准中的规范性引用文件

#### 【标准原文】

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件,仅注日期的版本适用于本文件。凡是不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

ISO/IEC 19761 软件工程——COSMIC: 一种功能规模度量方法(Software

engineering—COSMIC: a functional size measurement method)

ISO/IEC 20926 软件和系统工程——软件度量——IFPUG 功能规模度量方法 2009(Software and systems engineering—Software measurement—IFPUG functional size measurement method 2009)

ISO/IEC 20968 软件工程——Mk II 功能点分析——计数实践手册(Software engineering—Mk II Function Point Analysis—Counting Practices Manual)

ISO/IEC 24570 软件工程——NESMA 功能规模度量方法 2.1 版——功能点分析应用定义和计数指南(Software engineering—NESMA functional size measurement method version 2.1—Definitions and counting guidelines for the application of Function Point Analysis)

ISO/IEC 29881 信息技术——系统和软件工程——FiSMA1.1 功能规模度量方法(Information technology — Systems and software engineering—FiSMA 1.1 functional size measurement method)

#### 【标准解读】

ISO 是国际标准化组织(International Organization for Standardization)的简称,IEC 是国际电工委员会(International Electrotechnical Commission)的简称。这两家国际组织联合制定和发布了大量计算机和软件领域的国际标准(通常冠以 ISO/IEC 的标号)。我国国家标准(GB)有不少是参考这些国际标准制定的。

在软件规模估算过程中,标准引用了 ISO/IEC 的五项国际标准,并注明“下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件,仅注日期的版本适用于本文件。凡是不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。”作为限制说明。

ISO/IEC 19761、ISO/IEC 20926、ISO/IEC 20968、ISO/IEC 24570 和 ISO/IEC 29881 分别提供了五种功能点度量的方法标准,在对软件规模进行功能点度量时,标准的使用方应选择其中一种或多种参考标准中的具体方法进行度量。

## 3.2 软件研发成本构成

### 3.2.1 概述

#### 【标准原文】

本标准中软件研发过程包括从项目立项开始到项目完成验收之间的需求分

析、设计、编码、集成、测试、验收交付活动及相关的项目管理、支持活动。软件研发成本仅包括软件研发过程中的所有直接成本和间接成本(如图 3.1 所示),不包括数据迁移、软件维护等成本。直接成本包括直接人力成本和直接非人力成本,间接成本包括间接人力成本和间接非人力成本。本标准中所涉及工作量、工期也仅为软件研发过程所用工作量、工期。

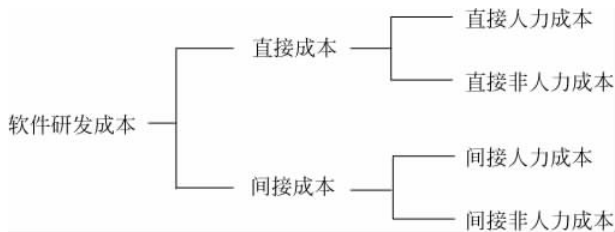


图 3.1 软件研发成本构成

#### 【标准解读】

在 SJ/T 11463—2013 中,软件研发过程包括从项目立项开始到项目完成验收之间的需求分析、设计、编码、集成、测试、验收交付活动及相关的项目管理、支持活动,不包括数据迁移、软件维护、硬件购置等非软件研发活动。因此,SJ/T 11463—2013 中软件研发成本仅包括开发方在软件研发过程中的各工作任务所花费的资源总和,并且可以按照上述划分的活动或阶段进行估算或测量。

软件研发项目按活动进行划分,主要包括需求、设计、构建、测试、实施、项目管理、配置管理、质量保证、培训。

软件研发项目按阶段进行划分,主要包括:计划、需求分析、设计、编码、测试、维护。

软件研发项目按角色进行划分,主要包括:项目经理、需求分析人员、设计人员、开发人员、测试人员、部署人员、用户文档编写人员、质量保证人员、配置管理人员。

数据迁移,是指原有的旧系统升级为新系统时,将其使用期间积累的大量珍贵历史数据进行清洗、转换,并装载到新系统中的过程。

软件维护,是指根据用户需求和 Service Level Agreement 承诺,向对方提供的纠正性维护、适应性维护、完善性维护或预防性维护等综合服务。

硬件购置,是指为保证软件正常的运行,所采购的有形的信息化设备和物理环境,以及维护、保养、更换部件等操作。

### 3.2.2 直接成本

#### 【标准原文】

直接人力成本包括开发方项目组成员的工资、奖金、福利等人力资源费用。其中,项目成员包括参与该项目研发过程的所有研发或支持人员,如项目经理、需求分析人员、设计人员、开发人员、测试人员、部署人员、用户文档编写人员、质量保证人员、配置管理人员等。对于非全职投入该项目研发工作的人员,按照项目工作量所占其总工作量比例折算其人力资源费用。

直接非人力成本包括:

- (1) 办公费,即开发方为研发此项目而产生的行政办公费用,如办公用品、通讯、邮寄、印刷、会议等;
- (2) 差旅费,即开发方为研发此项目而产生的差旅费用,如交通、住宿、差旅补贴等;
- (3) 培训费,即开发方为研发此项目而安排的特别培训产生的费用;
- (4) 业务费,即开发方为完成此项目研发工作所需辅助活动产生的费用,如招待费、评审费、验收费等;
- (5) 采购费,即开发方为研发此项目而需特殊采购专用资产或服务的费用,如专用设备费、专用软件费、技术协作费、专利费等;
- (6) 其他,即未在以上项目列出但确系开发方为研发此项目所需花费的费用。

#### 【标准解读】

软件研发的直接成本是为了达成特定研发项目所支出的各类资源总和。这些资源与研发项目是强关联的,一旦项目结束或中止,则这些成本不再发生。直接成本又分为直接人力成本和直接非人力成本。

直接人力成本通常是指开发方项目组成员的人力资源费用。例如:

- (1) 项目组成员的工资、奖金、加班费、社会保险、公积金。
- (2) 项目组成员的正常工作餐费、伙食费。

直接非人力成本通常是指为特定研发项目所支出的费用。例如:

- (1) 项目组封闭开发租用会议室而产生的费用计入直接非人力成本的办公费。
- (2) 对于异地的客户,项目成员出差是在所难免的,因出差所产生的交通、住宿、补贴等费用计入直接非人力成本的差旅费。
- (3) 开发方为了完成特定研发项目,给项目成员提供了必要的培训。这种培训是为了提升项目成员的相关研发技能,来更好地完成本项目研发工作。这部分

费用计入直接非人力成本的培训费。

(4) 项目研发过程中产生的一些辅助研发活动费用,如招待费、团队建设活动经费、评审费、验收费等,这些费用计入直接非人力成本的业务费。

(5) 项目研发过程中,需要独立采购特定的设备或软件,这部分费用的支出计入直接非人力成本的采购费。

### 3.2.3 间接成本

#### 【标准原文】

间接人力成本指开发方服务于研发管理整体需求的非项目组人员的人力资源费用分摊。包括研发部门经理、项目管理办公室(PMO)人员、工程过程组(EPG)人员、产品规划人员、组织级质量保证人员、组织级配置管理人员等的工资、奖金、福利等的分摊。

间接非人力成本指开发方不为研发某个特定项目而产生,但服务于整体研发活动的非人力成本分摊。包括开发方研发场地房租、水电、物业,研发人员日常办公费用分摊及各种研发办公设备的租赁、维修、折旧分摊。

**注:**在制定软件项目预算、报价或结算时,除软件研发成本外,考虑开发方合理的毛利润水平是必要的。对于需要提供其他支持服务的项目或产品,还需要考虑支持活动所需的各种成本,如数据迁移费、维护费等。

#### 【标准解读】

软件研发的间接成本是指组织级成本,服务于整体研发活动,只要该组织还有软件研发活动,间接成本就会发生,而不会因某个特定项目的结束或中止而不再发生。间接成本又分为间接人力成本和间接非人力成本。

间接人力成本指开发方服务于研发管理整体需求的非项目组人员的人力资源费用分摊。包括研发部门经理、项目管理办公室(PMO)人员、工程过程组(EPG)人员、产品规划人员、组织级质量保证人员、组织级配置管理人员等的工资、奖金、福利等的分摊。这些人员一般是组织级的研发管理人员,并不承担特定项目研发工作。

间接非人力成本指开发方不为研发某个特定项目而产生,但服务于整体研发活动的非人力成本分摊。包括开发方研发场地房租、水电、物业,研发人员日常办公费用分摊及各种研发办公设备的租赁、维修、折旧分摊。例如:

(1) 研发部门日常办公用的设备及软件成本,这部分费用可以按照间接非人力成本进行分摊。

(2) 研发部门办公场地的租用费用,可以按照间接非人力成本进行分摊。

### 3.2.4 成本构成常见问题解读

#### 1. 直接成本与间接成本的区别

在区分直接成本与间接成本时,可以假设当前项目中止或取消,未发生的相关费用是否还会继续产生,如果是,则属于间接成本,而如果相关费用不会发生,则属于直接成本。例如,某项目开发需要使用某项特殊技术,而项目组成员并不了解相关技术,需要进行培训以保证开发工作顺利进行,则此培训费用应计入直接成本;而如果是研发技能或研发知识的通用培训,例如研发管理流程培训,则此培训费用计入间接成本。

#### 2. 研发费用与非研发费用的区别

在“软件研发成本”范畴内,所有的费用一定是为研发活动服务的。如果有些费用与项目相关,但与研发活动无关,则不计入软件研发成本。例如,如果某项目研发需要采购特殊的硬件设备,则相关费用应计入直接非人力成本,而运行这个系统所需的设备,不属于软件研发成本,通常计入硬件购置费。

#### 3. 毛利润问题

毛利润通常包含开发方直接成本和间接成本之外的经营管理费用分摊、市场销售费用分摊、应承担的各种税费及税后净利。在制定软件项目预算、报价或结算时,除软件研发成本外,考虑开发方合理的毛利润水平是必要的。

## 3.3 软件研发成本度量过程

### 3.3.1 软件研发成本估算

#### 1. 基本流程与原则

##### 【标准原文】

图 3.2 展示了成本估算的基本流程。在依据此流程进行软件研发成本估算时应考虑以下情况:

- 在需求极其模糊或不确定时,宜采用类比法或类推法,直接粗略估算工作量和工期,也可直接粗略估算成本;

- 对于有明确工期要求的项目,在采用方程法估算工作量时,工期要求有可能是方程的参数之一;
- 间接成本是否与工作量估算结果相关取决于间接成本分摊计算方式;
- 工期估算结果与直接人力成本估算结果及其他成本估算结果相互关联并可能互相影响。如工期估算的结果有可能导致重新估算工作量和直接非人力成本,并最终改变软件研发成本估算结果。

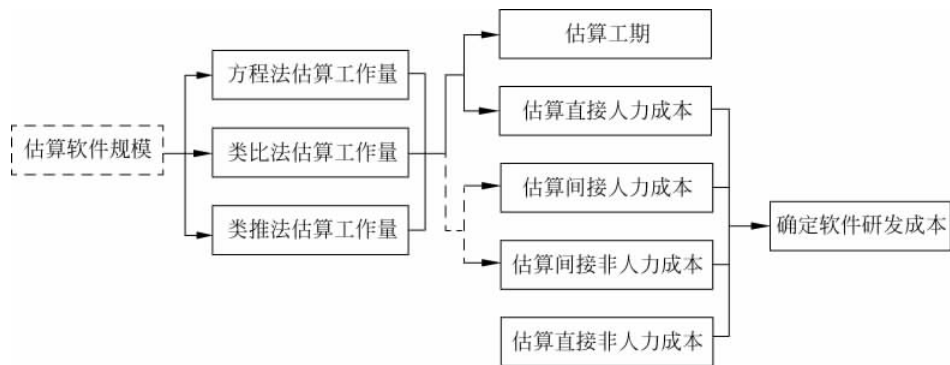


图 3.2 软件研发成本估算基本流程

在成本估算过程中,应遵循以下原则:

(1) 充分利用基准数据,采用方程法、类比法或类推法,对工作量、工期、成本进行估算。对于进行规模估算的项目,宜采用方程法估算工作量、工期和成本;

(2) 在规模估算时,应根据项目特点和需求的详细程度选择合适的估算方法;

(3) 工作量、工期、成本的估算结果宜为一个范围而不是单一的值;

(4) 成本估算过程中宜采用不同的方法分别估算并进行交叉验证。如果不同方法的估算结果产生较大差异,可采用专家评审方法确定估算结果,也可使用较简单的加权平均方法。

#### 【标准解读】

软件研发过程的特殊性决定了软件研发成本的估算方法既不同于制造业产品的成本估算方法,也不同于建设项目的财务评价方法。为指导软件研发相关方进行科学统一的估算,在 SJ/T 11463—2013 中,软件研发成本估算过程可进一步细分为估算规模、估算工作量、估算工期和估算成本四个过程,其中成本估算需要对直接人力成本、间接人力成本、间接非人力成本及直接非人力成本分别进行估算。

以下对四个估算过程分别给予说明。

#### 1) 估算规模

通常情况下,规模估算是软件成本估算过程的起点。估算规模是后续计算软件项目的工作量、成本和进度的主要输入,是项目范围管理的关键,因此,在条件允许的情况下,应进行规模估算。在规模估算过程中,需要注意以下情况:

(1) 在规模估算开始前,应根据可行性研究报告或类似文档明确项目需求及系统边界。项目需求除包含最基本的业务需求外,还应进行初步的子系统/模块划分,并对每一子系统/模块的基本用户需求进行说明,以保证可以根据项目需求进行规模预估。

(2) 依据项目特点和需求详细程度不同,通常估算人员在选择估算方法时应采用纳入国际标准的功能点方法进行功能规模估算,如 IFPUG 方法、Mk II 方法、COSMIC 方法、NESMA 方法及 FiSMA 方法。

(3) 若当前的项目需求极其模糊或不确定,可不进行规模估算,而直接采用类比法或类推法估算工作量、工期和成本。

#### 2) 估算工作量

在完成规模估算后,应当开展工作量估算工作,若当前项目未开展规模估算,也可直接启动工作量估算工作。工作量估算时,可采用方程法、类比法和类推法。

方程法:即基于基准数据建立参数模型,通过输入各项参数,确定估算值。

类比法:即将待估算项目的部分属性与类似的一组基准数据进行比对,进而确定估算值。

类推法:即将待估算项目的部分属性与高度类似的一个或几个已完成项目的数据进行比对,并进行适当调整后确定估算值。

在开展工作量估算的过程中,需要注意以下情况:

(1) 当需求极其模糊或不定时,如果此时具有高度类似的历史项目,则可直接采用类推法,充分利用历史项目数据来粗略估算工作量。

(2) 当需求极其模糊或不定时,如果此时具有与本项目部分属性类似的一组基准数据,则可直接采用类比法,充分利用基准数据来粗略估算工作量。

(3) 对于规模估算已经开展的项目,可采用方程法,通过输入各项参数,确定待估算项目的工作量。若客户或高层对项目的工期有明确的要求时,在采用方程法估算工作量时,工期要求有可能是方程的参数之一。

(4) 通常情况下,基于 WBS 的估算方式准确度相对较高,但是,这种方法要求项目需求完整、任务间逻辑清晰,且需要进行大量计算,工作量较大。

(5) 为追求估算的准确性,建议在条件允许的情况下,可采用两种估算方法,对估算结果进行交叉验证,若估算结果差别不大,可直接使用估算结果,若差别较大,可进一步采用专家评审法等方法来处理。

(6) 工作量的估算结果宜为一个范围,而不是单一的值。

### 3) 估算工期

在工作量估算结束后,可根据工作量采用科学的方法进行工期估算过程。在估算工期的过程中,需要注意的情况是:

(1) 类推法、类比法、方程法同样适用于工期估算。

(2) 工期估算的结果有可能导致重新估算工作量。比如,当工期估算结果长于期望工期时,压缩工期会增加项目工作量。

(3) 工期估算结果与直接人力成本估算及其他成本估算结果相互关联并可能相互影响,可能导致重新估算直接非人力成本,从而最终改变软件研发成本估算结果。比如,为了满足工期要求,项目团队通常会加班,而由此产生的加班费和餐费等,会分别引起直接人力成本和直接非人力成本的增加。

(4) 工期的估算结果通常为一个范围,而不是单一的值。

### 4) 估算成本

在获得了工作量和工期结果后,可采用科学的方法进行成本估算。在成本估算过程中,需要注意的情况有:

(1) 类比法和类推法,同样适用于需求极其模糊或不确定时的成本估算。

(2) 间接成本是否与工作量估算结果相关取决于间接成本分摊计算方式。在间接成本中,服务于研发管理整体需求的非项目人员的间接人力成本的分摊计算方式就直接关系到工作量估算结果,比如部门经理、项目管理办公室(PMO)、工程过程组(EPG)人员等的成本。而不为某个特定项目而产生的服务于整体研发活动的间接非人力成本的分摊计算方式就与工作量估算结果无关。

(3) 成本估算结果,也通常为一个范围,而不是单一的值。

在软件研发成本估算过程中,相关人员还需要注意:

- 基准数据:在采用方程法、类比法或类推法对工作量、工期和成本进行估算过程中,应充分利用基准数据。

对于委托方和第三方,建议使用或参考行业基准数据进行估算。估算模型的调整因子的增减或取值有可能随着行业基准数据的变化而变化。

对于开发方,在引入行业基准数据的基础上,可逐步建立组织级基准数据库,以提高估算精度。组织级基准数据定义应与行业基准数据定义保持一致,以便于与行业基准数据进行比对分析,并持续提升组织能力。

- 交叉验证：为提高估算结果的准确性和科学性，通常在规模、工作量、工期和成本的估算过程中，可以选择类比法、类推法或方程法等多种不同的方法对项目分别进行估算，并将多个方法的估算结果进行交叉比对分析，从而对估算结果进行验证。如果不同方法的估算结果差异不大，则说明当前项目的估算结果是可用的，可直接使用，或者使用平均值作为估算结果。如果不同方法的估算结果差异较大，可进一步分析和调查差距产生的原因，并通过专家评审法和加权平均法对差异情况做进一步处理，从而获得最终的估算结果。

## 2. 估算软件规模

### 1) 规模估算

#### 【标准原文】

在规模估算前，应根据项目范围明确系统边界。对于尚未确定的需求，应该在规模估算前确定估算原则。

估算人员应根据已确定的系统边界和需求描述估算软件规模。

规模估算所采用的方法，应根据项目特点和估算需求，选用国际标准化组织 ISO/IEC 已发布的以下五种规模度量标准中的一种，即：

- ISO/IEC 19761(COSMIC-FFP 方法)；
- ISO/IEC 20926(IFPUG 方法)；
- ISO/IEC 20968(Mk II 方法)；
- ISO/IEC 24570(NESMA 方法)；
- ISO/IEC 29881(FiSMA 方法)。

在规模估算时，应考虑可能的需求变更程度，并对规模估算结果进行适当调整。

**注：**根据相关国际标准中的方法适用范围声明，COSMIC 方法适用于商业应用软件和实时系统；IFPUG 方法适用于所有类型软件的功能规模度量；Mk II 方法适用于逻辑事务能被确定的任何软件类型；NESMA 方法与 IFPUG 方法非常类似，但对功能点计数进行了分级，以便在估算的不同时期选择不同精度的方法进行估算；FiSMA 方法适用于所有类型软件的功能规模度量。

#### 【标准解读】

在规模估算开始前，应根据可行性研究报告或类似文档明确项目需求。项目需求除包含最基本的业务需求外，还应进行初步的子系统/模块划分，并对每一子系统/模块的基本用户需求进行说明，以保证可以根据项目需求进行规模预估。如

何编写软件需求,可参考 GB/T 9385—2008《计算机软件需求规格说明规范》。

(1) 对于已确定的项目需求,其次应明确系统边界。系统边界包含如下含义:

① 用于划分系统与其他系统,特别是相邻系统关系的一种方法,将项目分割成系统内和系统外,系统内属于项目创建内容,系统外不需要创建,但需要考虑和它们之间的接口;

② 应说明哪些元素属于系统内,哪些元素属于系统外部环境;

③ 除了能确定系统内元素外,还应界定本系统对外的输入与输出,即本系统与外部环境的关系。

(2) 对于尚未确定的项目需求,应该在规模估算前根据项目具体特点和商务因素确定估算原则,如:

① 将尚未确定的需求按 0.5 进行计数或为降低竞标价格按 0 进行计数,后续再通过需求沟通继续明确项目范围;

② 如预算暂无金额限制,为避免功能缺失,未确定的需求也按 1 进行计数;

当项目需求、系统边界及项目特点确定后,估算人员即可从国际标准化组织 ISO/IEC 已发布的五种功能规模度量标准中选择合适的标准估算软件功能规模。

## 2) 非功能需求规模度量

非功能需求,指软件产品为满足业务需求而必须具有的,且除功能需求以外的特性。非功能用户需求是描述软件如何实现功能而不是具备什么功能。非功能特性包括产品必须具备的质量属性和必须遵守的约束。例如:软件性能需求、软件安全性需求、软件可用性需求等。

相对于功能规模,非功能需求的规模更加难以度量。为了有效评估非功能规模对于项目资源代价的影响,行业内通常有两类处理方式:宏观方式和微观方式。宏观方式即不对非功能规模直接度量,而是以功能规模为基础,通过对基准数据的细分,确定特定类型软件的软件因素调整因子,进而估算项目所需的工作量、成本、工期,通过此种方式估算的结果,已包含此类软件通常所涉及的非功能要求对于项目资源的影响;微观方式则是对非功能规模直接度量,一般采用两种方式,一是可以通过对功能点方法进行定制,定量评估非功能规模。例如,在金融行业,由于性能等方面的考虑,大量账务处理是通过后台批量程序定时完成的,通过对功能点方法适当定制,可以有效地对此类需求进行规模度量;二是引入专用的非功能规模度量方法(如 SNAP),此类方法针对非功能规模提出了明确的评估规则,但由于方法产生较晚,相关行业实践及数据较少,在实际应用时,还需要开展相关分析工作,以保证和功能规模数据有效结合,进而获得准确的估算结果。

### 3) 规模估算要点解读

#### (1) 如何理解用户

用户是指使用本系统的人或其他系统。

#### (2) 确定系统边界

系统边界识别规则：

- 边界的定义应基于用户的视角,边界必须是用户能够理解和描述的；
- 应用之间初始的边界不会因为功能点分析而改变；
- 在行业实践中,通常产品型组织(如金融、电信等行业客户)会根据产品划分系统边界,而项目型组织(如软件外包公司)通常会根据项目划分系统边界,无论以何种方式划分系统边界,均应在功能点计数项识别之前,同时在计数过程中保持不变；
- 如果组织的系统边界划分方式发生改变,应对影响范围进行评估,并对历史数据做出适当调整。

#### (3) 识别逻辑文件(ILF/EIF)

ILF 内部逻辑文件识别规则：

- ILF 指在待开发系统内部逻辑上的一组数据；
- 用户可以理解和识别 ILF,对 ILF 的操作是用户的业务需求；

EIF 外部接口文件识别规则：

- 从用户角度出发识别的一组逻辑数据；
- 这组数据是在应用程序外部,并被应用程序引用的；
- 这组数据是作为另一个应用程序中的 ILF 被维护的；

在识别逻辑文件时的常见错误包括(但不限于)：

- 将编码数据识别为逻辑文件,例如,将用户无法维护的业务规则识别为逻辑文件；
- 多计逻辑文件,例如,将关联表识别为独立的逻辑文件；
- 将物理文件的技术修改当作逻辑文件的变化,例如,逻辑文件的某个属性从 20 个字符扩展为 80 个字符；
- 将 ILF 与 EIF 混淆,例如,将临时存储的外系统数据识别为 ILF。

#### (4) 识别基本过程(EI/EO/EQ)

用户可以理解的、业务上的原子操作就是独立的基本过程。每个基本过程应该产生业务价值,同时系统进入相对稳定的状态。

区分 EI、EO、EQ 首先应根据基本过程的主要目的,如果该基本过程的主要目的是为了维护内部逻辑文件或改变系统行为,则该基本过程一定为 EI。

如果一个基本过程的主要目的是为了向系统边界之外(包括用户或其他系统)发送或呈现信息,则该基本过程为 EO 或 EQ。区分 EO 或 EQ 应根据该基本过程的行为。如果该基本过程包含计算、产生衍生数据、维护内部逻辑文件、改变系统的行为这四种处理逻辑中的任何一种,则该基本过程为 EO,否则为 EQ。

在识别基本过程时的常见错误包括(但不限于):

- 将仅针对编码数据的操作识别为基本过程,例如,页面显示静态信息;
- 将一个基本过程的多个处理逻辑识别为独立的基本过程。例如,将一个同步处理的报文转发功能识别为报文接收、报文数据格式转换、报文发送三个基本过程;
- 将一个基本过程的多个处理分支识别为独立的基本过程,例如,将审批识别为审批同意和审批不同意两个基本过程;
- 基本过程重复计数,例如,可通过菜单、按钮等多种方式调用的同一功能被重复计数;
- EO 与 EQ 混淆,例如,将含有筛选处理逻辑的查询功能识别为 EO;
- EI 与 EO/EQ 混淆,例如,将包含维护逻辑文件处理逻辑的统计功能(主要目的是统计结果展示)识别为 EI。

#### 4) 功能点方法规则定制

虽然功能点方法已形成国际标准,但在深入应用功能点方法时,通常还需要对计数规则进行一定的定制,主要原因如下:

- 在一些具体计数场景下,标准的规则存在一定的歧义,甚至不同功能点组织也有不同的解释。例如,对于一些业务操作之前的前导查询功能,是否计数及如何计数;
- 采用标准规则难以满足组织的管理需求。例如,对于某后台系统,要向 10 种不同的前台系统提供同一交易功能,但每个前台系统的传入参数略有不同。如果按照标准的功能点规则,则应该根据后台系统所提供的接口数量识别事务功能。但由于后台系统的接口定义通常是由后台系统的开发者确定,采用标准功能点计数规则进行计数,可能会导致后台系统开发者将一个接口故意拆分成多个接口实现,这样既降低了系统效率,又无法获得客观的规模计数结果。

在进行功能点方法规则定制时,应遵循以下原则:

- 定制规则应该可以被清晰一致地执行;
- 应评估定制规则后获得规模数据与采用标准方法计数获得规模数据的差异,在行业对标时,应对相关数据进行规格化处理。

通常需要进行规则定制的系统或功能包括(但不限于):

- 地理信息系统(如图层的计数);
- 消息转发类系统(如消息中间件);
- 数据仓库类系统;
- 嵌入式系统;
- 批处理程序。

### 3. 估算工作量

“软件系统能够按时和在预算内交付,并能满足质量要求”,通常我们称这样的项目是成功的。那么“按时和在预算内交付”的前提是软件工作量的正确估算,因此,工作量估算是进行软件估算的主要对象。工作量与软件研发成本存在直接关系,工作量估算是软件成本估算的基础。

工作量是指从项目立项开始到项目完成验收之间开发方的需求、设计、构建(包括编码、集成)、测试、实施及相关的项目管理、支持活动的工作量。

需求活动包括需求调研、需求分析、原型开发、编制各种需求文档、需求评审和需求变更等活动;

设计活动包括架构设计、技术方案选择、概要设计、详细设计、设计评审和设计变更等活动;

构建活动包括编码、代码走查、集成等活动;

测试活动包括测试计划、测试用例编写、测试用例评审、测试用例变更、测试环境准备及验证、单元测试、集成测试和系统测试等活动;

实施活动包括用户支持文档编写及验证、验收测试、系统安装部署和用户培训等活动;

其他活动:是指在上述活动中没有包含的项目中的其他活动,例如项目管理、质量保证、配置管理、项目组内部培训、技术讨论及交流等活动。

在 SJ/T 11463—2013 中,已明确指出项目成员包括参与该项目研发过程的所有研发或支持人员,如项目经理、需求分析人员、设计人员、开发人员、测试人员、部署人员、用户文档编写人员、质量保证人员、配置管理人员等。此处需要注意的是,项目组成员包括该项目的 QA 及配置管理人员,但不包括客户或用户。因此,项目组工作量的统计也不包括客户、用户或其他项目组外人员的工作量。

#### 1) 估算准备

##### 【标准原文】

在进行工作量估算前,应:

(1) 对项目风险进行充分分析。风险分析时应考虑技术、管理、资源、商业多方面因素。例如需求变更、外部协作、时间或成本约束、人力资源、系统架构、用户接口、外购或复用、采用新技术等。

(2) 对待实现功能复用情况进行分析,识别出复用的功能及可复用的程度。

(3) 根据经验或相关性分析结果,确定影响工作量的主要属性。

委托方应考虑的主要因素包括(但不限于):

- ① 软件规模;
- ② 应用领域,如委托方组织类型、软件业务领域、软件应用类型等;
- ③ 质量要求,如可靠性、可使用性、效率、可维护性、可移植性等。

开发方除考虑以上因素外,还应考虑的因素包括(但不限于):

- ① 采用技术,如开发平台、编程语言、系统架构、操作系统等;
- ② 开发团队,如开发方组织类型、团队规模、人员能力等;
- ③ 过程能力,如开发方过程成熟度水平、管理要求等。

(4) 选择合适的工作量估算方法。对于难以进行规模估算的项目,宜采用类比法或类推法;对于已经进行了规模估算的项目,宜采用方程法。

#### 【标准解读】

在进行工作量估算前,应从以下几个方面进行准备。

##### (1) 风险分析

风险分析是项目管理中的重要活动,其目的在于协助项目开发组织识别项目运行过程中的潜在问题,并提前采取措施。项目的风险可能来自许多方面,一般建议从技术、管理、资源、商业等方面进行考虑。在进行工作量估算前进行风险分析,旨在使用风险分析所得结果对工作量估算的结果进行适当的调整。

**技术风险:**指软件在设计、实现、接口、验证和维护过程中可能发生的潜在问题,如需求规格说明的二义性、采用陈旧或不成熟的技术等因素对软件开发项目带来的危害。

**管理风险:**指项目管理周期中因需求、合同、进度、流程、沟通、协作等因素威胁项目顺利进行的风险。

**资源风险:**指由于项目在预算、人员、资源等方面的原因对软件开发项目产生的不良影响。

**商业风险:**指与市场、企业产品策略等因素有关的风险。

一般的风险管理方法中,通常使用风险发生概率与风险影响程度的乘积作为风险系数,便于开展风险管理。在进行工作量估算前,同样可以使用该方法获得风险系数,从而对工作量进行调整。

例如,采用方程法进行工作量的估算,可在方程中设置反映风险分析结果的参数,根据风险分析的结果对参数进行调整,从而影响工作量估算的结果;采用类推法进行工作量的估算,在找到高度相似的历史项目估算工作量时,也应根据风险分析的结果对估算结果进行适度的调整。

### (2) 复用程度分析

在现代的软件开发过程中,软件复用是避免重复劳动的良好解决方案。其出发点是软件开发不再采用一切从零开始的模式,而是以已有的工作为基础,充分利用过去软件开发中积累的知识和经验。如:需求分析结果、设计方案、源代码、测试计划及案例等。

为了提高软件开发的效率和质量,现在大部分软件企业都已将某些通用功能转化为可重用功能,或者开发组织具备某方面项目的开发经验,遗留下了可复用的组件,这些情况都可能降低开发所需的工作量。

对复用情况的分析原则,可以考虑从系统功能的复用度入手,结合功能点方法,对于每个逻辑文件的复用程度给出明确的定义和系数。可应用在规模估算之后,在未调整规模的基础上进行复用程度的调整,从而间接实现对工作量的调整。

要判断复用度,可以从企业的实际情况出发,定义适合本组织的复用度。

逻辑文件复用度定义示例见表 3.3 和表 3.4。

表 3.3 ILF 复用调整因子

判断条件	数 值
现有产品中没有处理这类数据	3/3
现有产品处理过这些数据,但提供的 EI/EO/EQ 与需求有一定差异	2/3
现有产品处理过这些数据,提供的 EI/EO/EQ 完全达到或超过需求	1/3

表 3.4 EIF 复用调整因子

判断条件	数 值
现有产品从未引用过类似数据	3/3
现有产品曾引用过类似数据,但引用方式有较大差异	2/3
现有产品曾采用相同方式引用过类似数据	1/3

从组织实际应用的角度出发,可以定义更多级别的复杂度,但需要考虑在判断复杂度方面所付出的成本。

### (3) 软件因素和开发因素分析

软件因素可理解为是待开发软件或系统本身所具有的特性,是客观存在的,不

会随着不同的开发者而不同。可被直观地识别,也可通过某些方法被识别。对于待开发软件或系统,这些特性更多的表现为一种约束,任何开发者在进行开发时,都必须将软件因素作为一种约束条件考虑在内。

典型的软件因素如下:

① 规模:可以通过功能点方法来进行估算,并根据历史数据分析规模对生产率的影响。虽然传统的估算理论及模型认为随着项目规模的增加,项目复杂程度变大,导致生产效率有所降低。但依据行业基准数据库(CSBMK-201610)<sup>①</sup>对国内外数据的分析结果,在通常的商业应用开发中(规模介于100~10 000FP之间),项目生产率会随着系统规模的增加而缓慢提高。

② 应用领域:主要对于委托方而言,其组织类型(政府、银行、大型企业等),待开发项目的业务领域(金融、政务、生产制造等),应用类型(OA、ERP、MIS等)等都是可以直观识别的软件属性。

③ 质量属性:一般指可靠性、可使用性、效率、可维护性、可移植性。开发组织应特别注意明确委托方对于质量的需求,并将其作为工作量调整的重要考虑因素。

④ 开发因素更多的是由开发团队的特性决定的。不同的开发团队,因自身特点不同,在完成同样的软件项目时,所消耗的工作量也不相同。一般常见的开发因素有:

- 采用技术:如开发该系统所用的语言、开发平台、系统架构、操作系统等。
- 开发团队:开发方的组织类型、团队规模、个人能力等。
- 过程能力:开发方的成熟度(如所具备的CMMI成熟度)、管理要求等。

#### (4) 估算方法的选择

估算方法主要指类比法、类推法、方程法。一般情况下估算工作量应由规模估算的结果作为输入,然后采取方程法进行估算。但在一些特殊情况下,如需求非常模糊导致无法进行规模估算时,可直接采用类比法或类推法直接估算工作量。

#### 2) 估算与调整

##### 【标准原文】

在进行工作量估算时,应:

(1) 根据风险分析结果,对估算方法或模型合理调整。如调整估算模型中影响因子的权重或取值,或根据风险分析结果进行软件完整性级别定义并根据完整

<sup>①</sup> 注:有关行业基准数据库(CSBMK-201610)相关内容见本书附录A。

性级别调整工作量估算结果；

(2) 根据可复用的规模及可复用程度对工作量估算进行调整；

(3) 采用不同的工作量估算方法时，分别遵循以下原则：

① 在使用类推法时，参考的历史项目应和待估算项目有高度的相似性。在估算时应识别出待估算项目与参考历史项目的主要差异并对估算结果进行适当调整；

② 在使用类比法时，应根据主要项目属性对基准数据进行筛选；当用于对比的项目数量过少时，宜按照不同项目属性分别筛选对比，综合考虑工作量估算结果；

③ 在使用方程法时，宜基于基准数据，并采用回归分析方法，建立回归方程。可根据完整的多元方程(包含所有工作量影响因子)，直接计算出估算结果；也可根据较简单的方程(包含部分工作量影响因子)，计算出初步的工作量估算结果，再根据其他调整因子，对工作量估算结果进行调整。

宜采用不同的方法分别估算工作量并进行交叉验证。如果不同方法的估算结果产生较大差异，可采用专家评审方法确定估算结果，也可使用较简单的加权平均方法。

在估算工作量时，宜给出估算结果的范围而不是单一的值。例如，可采用基准对比方法，根据基准数据库中 25 百分位数、50 百分位数和 75 百分位数的功能点耗时率数值，分别计算出工作量估算的合理范围与最有可能值。

示例：

假设基于基准数据建立的回归方程为

$$UE = C \times S^{0.9}$$

式中：

UE——未调整工作量，单位为人时(ph)；

C——生产率调整因子，单位为人时每功能点(ph/FP)；

S——软件规模，单位为功能点(FP)。

假设根据相关性分析和经验确定调整后工作量计算公式为

$$AE = UE \times A \times L \times T$$

式中：

AE——调整后工作量，单位为人时(ph)；

A——应用领域调整因子，取值范围 0.8~1.2；

L——开发语言调整因子，取值范围 0.8~1.2；

T——最大团队规模调整因子，取值范围 0.8~1.2。

假设待估算项目的规模为 1000FP,参考基准数据的功能点耗时率 25 百分位数、50 百分位数和 75 百分位数, $C$  取值分别为 8ph/FP、10ph/FP、14ph/FP,则计算出未调整工作量合理范围介于 4009.50ph 与 7016.62ph 之间,未调整工作量最有可能值为 5011.87ph。

假设根据参数表确定应用领域调整因子取值为 1,开发语言调整因子取值为 0.8,最大团队规模调整因子取值为 1.1,则计算出调整后工作量合理范围介于 3528.36ph 与 6174.63ph 之间,调整后工作量最大可能值为 4410.45ph。

因项目变化导致需要重新进行工作量估算时,应根据该变化的影响范围对工作量估算方法及估算结果进行合理调整。

### 【标准解读】

#### (1) 风险估算

在进行工作量估算时,应首先考虑风险分析的结果。通过风险分析的结果,对工作量估算模型中的因子设置不同的权重和取值。

此处需要注意分析需求变更对工作量产生的影响。需求变更是软件开发过程中普遍存在的问题,并且给软件成本估算带来了不小的影响。大多数企业需求变更评估人员不是软件开发人员,无法很清楚地了解需求状态及相关信息,导致需求变更后工作量评估的不确定性,有时不能做出合理的决策,加大了软件项目风险。

目前,大多数软件开发组织都已进行了需求风险管理,如编写需求风险文档、定义风险级别、评估风险危害值等,这些措施在一定程度上帮助软件开发组织规避和缓解了风险。快速功能点度量方法倡导通过设置不同的影响因子值来分析需求变更给工作量带来的影响。

在项目估算早期,存在软件需求模糊、不明确,未来有很多隐含需求及需求变更的情况,依据行业数据,规模变更因子通常取值为 1.5;

在项目估算中期,项目需求已较清晰,综合项目将来有可能的需求变更和目前开发工作中的复用,依据行业数据,规模变更因子通常取值为 1.26;

在项目估算后期,项目需求已基本完成确定或项目已基本开发完成,依据行业数据,规模变更因子通常取值为 1.0。

#### (2) 调整复用度

在对整个工作量估算模型进行调整后,需要针对复用程度来对工作量进行调整。这种调整建议在进行了规模估算后进行。例如,在用快速功能点方法估算出内部逻辑文件(ILF)和外部接口文件(ELF)后,可考虑针对每个逻辑文件分析其复用程度,然后对逻辑文件的功能点数进行调整。这种调整虽然表现为调整功能

点数,但其实质目的还是为调整工作量。对每一个逻辑文件进行调整,可更好地反映出不同的功能复用对整体工作量的影响。

### (3) 工作量估算

**类推法:**是属于以“估”为主的方法。将待评估项目与过去的一个或多个项目进行比较推算,确定特别相似和不同的地方,最后基于这种差异来进行实际工作量的调整。

采用类推法时应注意,所选择的历史项目与待评估项目一定是高度相似的,历史数据尽量选择本组织内的数据,并且一定要对差异之处进行调整。虽然类推法是迄今为止理论上最可靠的估算方法,由于它是以“估”为主,脱离不了评估人员的主观性,所以估算结果也是经常产生极大偏差。

**类比法:**是属于以“算”为主的方法。当待评估项目与已完成项目在某些项目属性(如应用领域、系统规模、复杂度、开发团队经验等)相类似时,可使用类比法。它是基于大量历史项目样本数据来确定目标项目的预测值。

采用类比法时应注意,当供选择的样本数量不足时,可以通过选择单个项目属性进行筛选比对,根据结果综合进行工作量的调整。

**方程法:**是属于以“算”为主的方法。在考虑开发组织实际情况下建立可靠的估算模型。

#### ① 一元线性模型

$$\text{工作量} = \text{规模} \times \text{生产率} + C \quad (3.1)$$

其中,C为一个常量,多数情况下为0。这是最简单的估算模型。

#### ② 多元线性模型

$$\text{工作量} = \text{规模} \times \text{生产率} \times \text{复用率} \times \text{复杂度} \times \text{人员能力系数} \times \dots + C \quad (3.2)$$

其中,C为一个常量,多数情况下为0。多元线性模型需要估计软件项目的属性。

#### ③ 行业级模型

$$\begin{aligned} \text{工作量} = & \text{规模} \times \text{生产率} \times \text{软件因素调整因子} \\ & \times \text{开发因素调整因子} \end{aligned} \quad (3.3)$$

**交叉验证:**以“估”为主的方法和以“算”为主的方法在实际中一般混合使用,以互相补充、互相印证。两类方法各有优缺点,一般不可以只采用一种方法进行估算或只有一人进行估算。采用多种方法进行估算,将结果进行比对,若不同的估算方法得出的估算值,偏差范围在5%以内,即可以接受估算结果;若偏差较大,则需要分

析差异原因,以提高估算的合理性。

工作量估算的结果:工作量估算的结果受到各种因素影响,很难得到一个固定的值。进行工作量估算的主要目的,更多的是了解待开发系统在功能规模一定的情况下可能的工作量水平。所以工作量估算的结果一般以一个范围的形式呈现,表示出工作量的最可能值以及合理的范围。可参考统计方法中的百分位法,以P50来表示最可能的值,以P25、P75来表示合理范围值的下限和上限。采用类推法、类比法的时候,可直接取得相应的值。如使用方程法,则需要在基准数据中选择功能点耗时率的P50(代表有50%的数据不大于该值)、P25(代表有25%的数据不大于该值)、P75(代表有75%的数据不大于该值)三个值,然后以功能点数分别乘以这三个值,即得到工作量的范围:

功能点数×功能点耗时率 P25=下限

功能点数×功能点耗时率 P50=最可能值

功能点数×功能点耗时率 P75=上限

工作量估算的结果是建立项目目标及承诺的基础。在实际的项目过程中,应根据项目特点及约束选择合适的估算结果。例如,在制定项目预算时,如果为了保证项目有充足的预算并按时按质交付,则可依据估算结果的上限编制预算,而在编制项目计划时,可以依据估算结果的最可能值。

### 3) 工作量估算要点解读

#### (1) 交叉验证

在现阶段的软件工程领域内,由于没有一种估算方法可以足够准确地解决所有问题,所以很多情况下我们需要使用两种以上的估算方法来交叉验证。在方法数量上,至少要选择两种以上方法,尽量包含以“估”为主的方法和以“算”为主的方法。其次,如果交叉验证产生的不同的估算结果差异不大,可以简单处理(如以某种方法为准或取平均值);如果差异较大,则需要对差异进行原因分析,并对估算假设条件合理调整,来逐步减少结果差异,规避风险。

#### (2) 变更因素

由于在早期估算时,项目范围尚未明确,因此需考虑需求变更对规模的影响。根据行业数据统计,即使在合同签订之后,软件规模通常仍会有26%左右的变更。所以,一方面我们可以采用功能点方法更为高效地澄清关键业务需求,通过挖掘隐含需求减少变更;另一方面,在工作量估算之前也应对规模进行合理调整,以获得更为准确的估算结果。

## 4. 估算工期

### 1) 工期估算方法

#### 【标准原文】

在估算工期时,应:

(1) 根据工作量估算结果和资源情况,对工作任务进行分解并制定工作时间表。在制定工作时间表时,应充分考虑关键路径任务约束对工期的影响。如用户参与需求沟通活动的资源投入情况、委托方对试运行周期的要求等;

(2) 利用基准数据估算合理的工期范围。可利用基准数据,建立“工作量-工期”模型,使用方程法估算合理的工期范围;也可使用类比法,估算合理的工期范围;

(3) 将委托方的期望工期或开发方初步制定的工作时间表中的工期与工期估算结果进行比较;

(4) 如果委托方期望工期或工作时间表中的工期短于估算出的工期下限时,应分析原因,必要时需对人力资源安排或项目范围进行调整,再重新估算工作量、工期,并制定新的工作时间表。

#### 【标准解读】

对于一个软件工程项目来说,制定项目进度计划的第一步是估算每项工作从开始到完成所需要的时间。工期估算制定和控制工作计划的基础,无论是委托方还是开发方,都需要编制工作进度计划,估算项目每一种活动所需要的时间,以确定整个项目的时间结构。

项目工期估算是根据工作量估算结果、客户期望、项目人力和物力资源情况列出项目活动所需要的工期,估算的工期应该遵循符合实际、有效实施和保证质量三大原则。在估算工期时要充分考虑活动清单、合理的资源需求、团队能力、客户要求及环境因素对项目工期的影响,在对每项活动的工期估算中也应充分考虑风险因素对工期的影响。项目工期估算完成后,可以得到量化的工期估算数据。在估算工期时应包含如下步骤:

(1) 根据工作量估算结果和资源情况,对工作任务进行分解并制定工作时间表。在制定工作时间表时,应充分考虑如下因素:

- 关键路径任务约束对工期的影响。如用户参与需求沟通活动的资源投入情况、委托方对试运行周期的要求等;
- 识别干系人,并理解他们对项目的影响力也是至关重要的,不同的项目干系人可能对哪个因素最重要有不同的看法,从而使问题更加复杂,如果这

项工作没有做好,将可能导致项目工期延长或成本显著提高。例如,没有及时将法律部门作为重要的干系人,就会导致因重新考虑法律要求而造成工期延误或费用增加。

(2) 利用基准数据估算合理的工期范围。可利用基准数据,建立“工作量-工期”模型,使用方程法估算合理的工期范围;也可使用类比法,估算合理的工期范围。

在掌握大量数据的基础上,可利用回归分析法,通过数理统计方法建立因变量(工期)与自变量(工作量)之间的回归关系函数表达式,即回归方程。建立了“工作量-工期”模型后,可利用此模型对项目工期进行预测,预测结果建议作为参考,不要直接用于制定项目计划,须按(1)描述考虑项目具体因素进行调整。

通过分析行业数据统计的工期数据,不同工期下的实际成本见图 3.3。

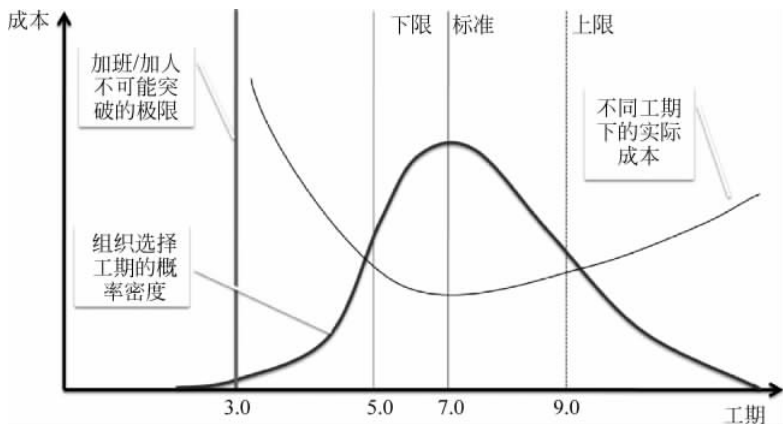


图 3.3 不同工期下实际成本

图 3.3 中下限、标准、上限值分别对应行业工期数据统计的 P25、P50、P75,代表此工期下成功交付项目的比例数,也代表项目成功的概率。

- 如委托方的期望工期或开发方初步制定的工作时间表中的工期长于模型标准值时,开发方只需要考虑资源投入。
- 如委托方的期望工期或开发方初步制定的工作时间表中的工期短于或等于模型标准值时,开发方则需要压缩工期并考虑相关的项目风险。
- 如委托方期望工期或开发方初步制定的工作时间表中的工期短于估算出的工期下限时,开发方应分析原因,必要时须对人力资源安排或项目范围进行调整,再重新估算工作量、工期,并制定新的工作时间表。

从图 3.3 可看出,随着资源的增加,工期并不能随之减少,工期存在通过加班/

加人不可能突破的极限。

## 2) 工期估算要点解读

压缩工期是指在不改变项目范围的前提下,缩短项目的进度时间,以满足进度制约因素、强制日期或其他进度目标。

压缩工期通常需要人员加班、提高预算、增加额外资源,从而在较短时间内完成同样的工作量。如无法提高预算,则只能缩小范围或降低质量,以便在较短时间内以同样的预算交付产品。压缩工期可能导致风险或成本增加,如增加的额外资源不能及时到位等,项目经理需要关注项目具体的制约因素及这些因素间的关系,合理安排项目工期,否则可能影响项目成功交付。

压缩工期并非总是切实可行的。通过权衡成本与进度,确定如何以最小的成本来最大限度地压缩进度,以规避风险,确保项目成功。

## 5. 估算成本

### 1) 成本估算方法

#### 【标准原文】

#### 估算直接人力成本

根据工作量估算结果和项目人员直接人力成本费率估算直接人力成本。直接人力成本费率是指每人/月的直接人力成本金额,单位通常为元/人月或万元/人月。直接人力成本的计算宜采用以下两种方式之一:

(1) 根据不同类别人员的直接人力成本费率和估算工作量分别计算每类人员的直接人力成本,将各类人员的直接人力成本相加得到该项目的直接人力成本;

(2) 根据项目平均直接人力成本费率和估算的总工作量直接计算该项目的直接人力成本。

直接人力成本的计算公式为

$$DHC = \sum_{i=1}^n (E_i \times IF_i) \text{ 元}$$

式中:

DHC——直接人力成本,单位为元;

$n$ ——人员类别数量,取值为不小于1的自然数;

$E_i$ ——第 $i$ 类人员的工作量,单位为人月;

$IF_i$ ——第 $i$ 类人员的直接人力成本费率,单位为元/人月。

在估算项目直接人力成本费率时,应考虑不同地域人员成本的差异。委托方可参照同类项目的直接人力成本费率数据;开发方应优先使用本组织的直接人力

成本费率数据。

### 估算直接非人力成本

宜根据项目情况,按照本标准第4章的要求分项估算直接非人力成本,也可依据基准数据或经验估算。

示例1:项目成员因项目加班而产生的餐费宜计入直接非人力成本中的办公费,而项目成员的工作午餐费宜计入直接人力成本。

示例2:项目组封闭开发租用会议室而产生的费用宜计入直接非人力成本中的办公费,而研发部例会租用会议室产生的费用宜按照间接非人力成本分摊。

示例3:为项目采购专用测试软件的成本宜计入直接非人力成本中的采购费,而日常办公用软件的成本宜按照间接非人力成本进行分摊。

### 估算间接人力成本

宜根据项目情况,按照第4章的要求分项估算间接人力成本。间接人力成本宜按照工作量比例进行分摊。

示例:质量保证部门的质量保证人员甲负责组织级质量保证工作和3个项目(A、B、C)的项目级质量保证工作。其中,用于项目A、B、C的工作量各占总工作量的1/4,用于组织级质量保证工作和其他工作的工作量占其总工作量的1/4;同时,项目A的研发总工作量占该组织所有研发项目总工作量的1/3,则质量保证人员甲的人力资源费用中,1/4计入项目A的直接人力成本,1/12(占质量保证工程师甲1/4的组织级质量保证工作和其他工作中,只有1/3计入项目A的成本)计入项目A的间接人力成本。

### 估算间接非人力成本

宜根据项目情况,按照第4章的要求分项估算间接非人力成本。间接非人力成本宜按照工作量比例进行分摊。

示例:公司甲有员工200人,1年的房屋租赁费为人民币120万元,则每人每月的房租分摊为500元,如果项目A的总工作量为100人月,则分摊到项目A的房屋租赁费为人民币5万元(即100人月×500元/人月)。

### 确定软件研发成本

软件研发成本的计算公式通常为

$$SDC = DHC + DNC + IHC + INC \text{ 元}$$

式中:

SDC——软件研发成本,单位为元;

DHC——直接人力成本,单位为元;

DNC——直接非人力成本,单位为元;

IHC——间接人力成本,单位为元;

INC——间接非人力成本,单位为元。

在估算软件研发成本时,可根据直接人力成本费率估算人力成本费率(即每人月直接人力成本与分摊到每人月的间接成本之和),计算公式为

$$F = IF \times (1 + DP)$$

式中:

$F$ ——人力成本费率,单位为元每人月;

$IF$ ——直接人力成本费率,单位为元每人月;

$DP$ ——间接成本系数,即分摊到每人月的间接成本占每人月直接人力成本的比例。

委托方和第三方宜参照行业基准数据确定  $DP$  的取值。

如果已经获得了人力成本费率,则可以依据工作量估算结果和人力成本费率直接计算出直接人力成本和间接成本的总和,然后再计算软件研发成本,计算公式为

$$SDC = \sum_{i=1}^n (E_i \times F_i) + DNC \text{ 元}$$

式中:

$SDC$ ——软件研发成本,单位为元;

$n$ ——人员类别数量,取值为不小于1的自然数;

$E_i$ ——第  $i$  类人员的工作量,单位为人月;

$F_i$ ——第  $i$  类人员的人力成本费率,单位为元/人月;

$DNC$ ——直接非人力成本,单位为元。

委托方可根据行业基准数据确定每人月直接人力成本与分摊到每人月的间接成本的比例,进而估算人力成本费率。

对于委托方,如果已经确定了规模综合单价,则可以根据规模综合单价和估算出的规模直接计算出直接人力成本和间接成本的总和,然后计算软件研发成本,计算公式为

$$SDC = P \times S + DNC \text{ 元}$$

式中:

$SDC$ ——软件研发成本,单位为元;

$P$ ——规模综合单价,单位为元每功能点;

$S$ ——软件规模,单位为功能点;

$DNC$ ——直接非人力成本,单位为元。

### 【标准解读】

确定软件研发成本主要有以下三种方法：

(1) 分别计算直接人力成本、直接非人力成本、间接人力成本、间接非人力成本,然后求和,计算得到软件研发成本。

(2) 依据工作量估算结果和平均人力成本费率直接计算出直接人力成本和间接成本的总和,再加上直接非人力成本,计算得到软件研发成本。

对于委托方,也可利用不含毛利润的开发方人力成本费率(即只包含直接人力成本和间接成本)估算软件研发成本,再根据开发方毛利润水平,确定预算费用。更常用也更简单的方式是根据该地区该行业的平均人力成本费率测算费用,然后对人员能力和岗位配比提出合理要求或限制,而不是直接根据开发方上报人员分级计算。

(3) 依据规模估算结果和规模综合单价直接计算出直接人力成本和间接成本的总和,加上直接非人力成本,计算得到软件研发成本。

实际应用中很少采用第一种四类成本求和的估算方法,多采用第二种方法来确定软件研发成本。如果委托方和开发方对规模估算方法有一致认可,且均能熟练掌握,也可采用第三种方法。此种方法更能适应项目范围存在较大变更概率的项目,可支撑委托方的费用预算审批,也可保护开发方的利益。应用第三种方法时,规模估算结果必须作为附件提交,如采用功能点方法进行规模估算的项目,上报预算时还应附上功能清单及对应功能点数。

在项目早期估算时,可根据平均人力成本费率确定人力成本。平均人力成本费率受物价指数、行业、人力资源供给状况、企业所在地、工作性质、人员级别等因素影响。如可根据不同角色进行估算,一般情况下总架构师高于需求分析师,需求分析师高于编程工程师,而同种角色会有多个人员级别设置,级别越高,平均人力成本费率越高。

据相关行业基准数据(CSBMK-201610),北京 2016 年平均人力成本费率为 2.43 万元/人月,该人月费率代表该北京地区行业中位数(P50),费用包含软件的直接人力成本、间接成本及合理利润,但不包括直接非人力成本。不同地域、不同城市间的平均人力成本费率也有较大差异。2016 年各典型城市人月费率见表 3.5。

直接非人力成本通常与工作量没有关系,有些项目直接非人力成本可忽略不计,有些项目需根据项目实际情况进行估算,如项目异地开发,则差旅费会较多。

#### 2) 成本估算要点解读

委托方在制定预算时,可根据费用情况及管理导向选择标准值或上限值。例如,对于质量、进度要求较高的项目,为了确保项目成功可按照预算金额的上限值

上报预算。如无特殊情况,不应以低于预算金额下限或高于预算金额上限的金额上报预算。在进行招标时,建议选择标准值,保证多数报价接近招标价格。

表 3.5 2016 年典型城市人月费率基准数据明细

典型城市	人月费率/(万元/人月)
北京	2.43
上海	2.44
广州	2.23
深圳	2.39
南京	2.06
苏州	2.18
济南	1.73
成都	1.69

### 3.3.2 软件研发成本测量

#### 1. 测量规模、工作量、工期

##### 【标准原文】

在项目研发过程中和项目结束后,应对项目的实际规模、工作量、工期进行测量。

在以下里程碑宜对规模进行测量:

- 需求完成;
- 设计完成;
- 编码完成;
- 内部测试完成;
- 项目结束后。

规模测量方法宜与规模估算所采用的方法一致。

应定期或事件驱动地对项目工作量、工期进行测量。

除对总工作量、总工期进行测量外,还宜对项目不同活动、不同阶段的工作量、工期分别进行测量。

##### 【标准解读】

从软件研发成本度量的角度来看,在完成了对软件研发项目的规模、工作量、工期和成本的估算后,并不意味着度量工作的结束。相反,在整个软件研发项目的生命周期中,还需要持续不断地对软件研发成本进行测量和分析。这些测量和分

析的工作,不仅仅是单个软件研发项目成功的关键因素,也是组织软件研发能力提升的基础。

因此,在软件研发项目前期完成项目的规模、工作量、工期的估算后,为满足项目计划和监控的要求,在项目研发过程中应该对实际规模、工作量、工期进行测量。为满足项目收尾的要求,在项目结束后对项目进行决算时,也需要对项目的实际规模、工作量、工期进行测量。

**规模测量:** 软件项目的规模一般会随着需求的逐渐清晰而不断明确,规模测量的时间点可以在软件研发项目的里程碑点进行。常见的项目里程碑点包括:

(1) 需求完成: 需求调研和分析完成,形成基线化的需求文档。需求完成后,功能规模理论上已确定,但考虑到实际的需求文档质量,可能要在设计完成后才可以进行详细功能点计数。

(2) 设计完成: 设计人员根据用户需求文档,完成设计文档并形成基线,以便提交开发人员进行编码和开发。

(3) 编码完成: 开发人员根据设计完成编码。

(4) 内部测试完成: 一般可选择系统测试完成时。

(5) 项目结束后: 系统上线并通过试运行,或通过用户验收测试等活动,得到用户可以结项的确认之后。

在各个里程碑点进行规模测量时,所采用的方法建议与规模估算所采用的方法一致。如规模估算阶段采用 IFPUG 方法,则后续各个里程碑点进行规模测量时,也应采用 IFPUG 方法。

此外,除了上述项目里程碑点之外,当项目发生正式的需求变更时,也有必要对规模进行测量。测量结果既是变更评估的依据,也将是之后项目计划调整的输入。

**工作量、工期测量:** 当项目的规模发生变化的情况下,典型情况如发生需求变更后,毫无疑问要对工作量、工期进行测量,以保证规模变化之后工作量和工期的准确性。

有时在软件规模不发生大的变化的情况下,软件项目在具体执行过程中的工作量和工期,仍可能受到技术和人员等多方面的影响。如一个软件在研发过程中遇到重大技术问题需要攻克,即便软件规模本身没有大的变化,仍需要对工作量进行调整,而工期也需要相应地变化。

由于工作量、工期受影响的因素较多,因此需要较为频繁地对工作量、工期进行测量。一般来说,可以按下述两种时间点对工作量、工期进行测量:

(1) 定期。随着项目的进行,可定期对工作量、工期进行测量,常见的频率为

每周、每半月或每月。如项目管理过程中本身有定期的报告制度,如项目周报、月报等,可随项目报告的周期进行工作量、工期的测量。其测量的结果也会对项目报告以及后续项目计划造成影响。

(2) 事件驱动。除定期地对工作量、工期进行测量之外,如在项目过程中出现较为重大的事件,也应随着事件的发生而对工作量、工期进行重新测量。需求变更之后的工作量、工期测量就是典型的事件驱动。除此之外,如上文提到的例子,在软件开发过程中突遇重大技术问题,可能需要投入人力加以解决,势必对工作量、工期造成影响,需要重新测量工作量和工期。又如在项目开发过程中,发生设备故障、人员损失(离职或生病)等情况,则可能会对工期造成影响,需要重新测量工期。

此外,对于工作量和工期的测量,除了对项目总体的工作量、工期进行测量之外,应建议对项目的不同活动、不同阶段的工作量、工期分别进行测量。例如,对不同类型的活动如需求活动、设计活动、开发活动等进行单独测量,也可以对策划阶段、设计阶段、开发阶段等不同阶段进行单独测量。这样做的目的,一方面是为了支持项目管理工作,为项目计划的调整带来更准确的输入;另一方面可以积累各个活动和阶段的度量数据,为组织级的度量分析工作做好数据准备,也可以指导后续项目的策划。

## 2. 测量成本

### 【标准原文】

在项目研发过程中,宜定期或事件驱动地对已发生的直接成本进行测量。

在项目结束后,宜按照本标准第4章的要求对各项成本分别进行测量。

对于可以按照交付软件规模进行结算的项目,应根据交付软件规模及规模综合单价计算实际成本。

### 【标准解读】

**项目研发过程中测量成本:**如 SJ/T 11463—2013 中第4章所述,软件研发成本分成直接成本和间接成本。其中间接成本方面,包括间接人力成本和间接非人力成本,都是不为特定研发项目而产生,但服务于整体研发活动的费用分摊,因此在特定研发项目过程中对间接成本进行测量的意义不大。而直接成本方面,包括直接人力成本和直接非人力成本,都是为特定研发项目而投入的,因此需要在软件研发过程中进行测量。而直接人力成本最直接的测量因素就是工作量,因此在软件研发过程中,可以只跟踪直接非人力成本和工作量。

在软件研发过程中,对软件研发直接成本的测量周期,也可分为定期和事件驱动两种形式,其原则可参考上文工作量、工期测量的内容。在事件驱动方面,需求

变更自不必说,承接上文提到的例子,如在软件开发过程中突遇重大技术问题,其解决方案无论是投入额外的人力,还是外购解决方案,都会对直接成本造成影响。又如在项目开发过程中,发生设备故障、人员损失(离职或生病)等情况,无论是设备维修或更换、还是人员重新雇佣等,也都需要重新测量直接成本。

**项目结束后测量成本:**在软件项目结束后,为了解软件开发项目的整体成本状况,有必要根据 SJ/T 11463 中第 4 章的要求,对各项成本分别进行测量。即除了直接成本中的直接人力成本和直接非人力成本外,也需要对间接成本中的间接人力成本和间接非人力成本进行分摊和测量。这些数据除了作为本项目评价的重要内容之外,也是组织级度量数据库的重要输入。特别是间接人力成本和间接非人力成本数据的积累,对今后组织的估算准确性具有非常重要的意义。

对于可以按照交付软件规模进行结算的项目,应根据交付软件规模及规模综合单价计算实际成本。此处交付软件规模应为项目结束后所测量的软件规模,其测量方法应与规模估算所采用的方法一致。此处公式可参见 SJ/T 11463 中 5.1.5.5 中的公式 5 的内容。

### 3. 软件研发成本分析

#### 【标准原文】

软件研发成本分析的内容主要包括:

- 成本估算偏差;
- 成本构成;
- 成本关键影响因素相关性分析;
- 成本估算方程回归分析。

在项目研发过程中,应定期检查实际发生成本与估算成本的偏差。如通过挣值分析方法监控项目进度。

数据分析的结果应与利益相关方充分沟通,并采取有效纠正措施。

项目结束后,应对成本及相关数据进行分析,并用于:

- 项目评价;
- 建立或校正成本估算模型;
- 过程改进。

项目规模、工作量、工期、成本等估算及实际数据应有效管理并保存在基准数据库中。

#### 【标准解读】

对软件研发成本进行分析有助于研发组织了解自身研发过程的情况,提高估

算准确性、控制成本并为改进提供重要的决策信息。

根据难度以及对成本分析的深入程度,可以分为:

(1) 成本估算偏差: 分析估计成本与实际成本的偏差及原因。

(2) 成本构成: 分析研发成本的构成情况,可以参考以下方式对成本进行分类:

- 参考 SJ/T 11463—2013 中第 4 章的描述,分为直接成本和间接成本,其中直接及间接成本中再分为人力成本和非人力成本;
- 根据生命周期分为需求、设计、编码、测试、交付等阶段成本;
- 根据工作类型划分为开发成本、项目管理成本、质量保证成本等。

通过以上分类的统计并结合“成本估算偏差”的信息了解成本估算产生偏差的主要原因。

(3) 成本关键影响因素相关性分析: 软件研发过程中多种因素对成本产生不同的影响,这些因素对在各组织中的影响程度不尽相同。因此,通过分析研发过程的因素与成本间的关系可以让管理者了解本组织成本控制的重点。

相关性分析通常使用的方法包括散点图以及 Pearson 相关分析:

- 散点图将实验或观测得到的数据用点在平面图上表示出来,显示了一个因素相对于另一个因素是如何变化的。如某软件项目的规模、工作量散点图示例见图 3.4。

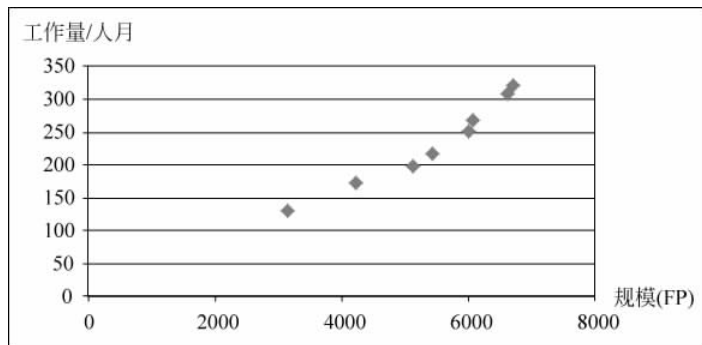


图 3.4 散点图

- Pearson 相关分析是统计学中分析变量线性相关的方法,通过计算可以得到变量间量化的相关系数,并通过相关系数判断因素对成本影响的大小。表 3.6 通过计算得到了各变量间的相关系数,用以判断其相关性的强弱。

表 3.6 各变量间相关系数

变量相关系数变量	生产率	项目规模	项目人数	项目类型
生产率	1			
项目规模	0.636 439 819	1		
项目人数	0.391 221 359	-0.040 840 657	1	
项目类型	0.136 748 977	-0.347 749 557	-0.180 472 681	1

(4) 成本估算方程回归分析：软件研发的管理者除了希望了解成本关键影响因素的影响程度以外，还希望获得成本与关键影响因素的量化关系，即关于成本与关键影响因素的估算方程。

回归分析(Regression Analysis)是确定两种或两种以上变数间相互依赖的定量关系的一种统计分析方法，常用于建立回归方程。回归分析一般的步骤包括：

- 根据预测目标，确定自变量和因变量；
- 建立回归预测模型；
- 进行相关分析；
- 检验回归预测模型，计算预测误差。

表 3.7 通过线性回归工具对数据进行回归后得到预测模型：

$$\text{生产率} = -166.69 + 0.009 \times \text{项目规模} + 39.92 \times \text{项目人数} + 42.70 \times \text{项目类型}$$

表 3.7 统计分析结果

回归统计						
Multiple R		0.90				
R Square		0.81				
Adjusted R Square		0.77				
标准误差		49.07				
观测值		20.00				
方差分析						
	df	ss	MS	F	Significance F	
回归分析	3.00	163 167.78	54 389.26	22.59	0.00	
残差	16.00	38 525.97	2407.87			
总计	19.00	201 693.75				
	Coefficient	标准误差	t Stat	P-value	Lower 95%	Upper 95%
intercept	-166.69	65.24	-2.56	0.02	-305.00	-28.39
项目规模	0.09	0.01	7.16	0.00	0.06	0.11
项目人数	39.92	8.59	4.65	0.00	21.72	58.12
项目类型	42.70	9.74	3.39	0.00	22.06	63.33

建立回归方程后,管理者可以在项目初期及项目中期对成本进行预测,并通过提前控制影响成本的关键因素达到控制成本的目的。

成本估算偏差的测量与分析可以使用挣值分析方法,该方法的核心是将项目在选定时间的计划指标、完成状况和资源耗费进行综合度量,将这些信息转化为统一的单位进行管理,如货币、工时等,从而能准确描述项目的进展状态。该方法的另一个重要优点是可以预测项目可能发生的工期滞后量和费用超支量,从而及时采取纠正措施,为项目管理和控制提供了有效手段。

挣值分析方法使用的一般步骤如下:

(1) 定期或者事件驱动地收集项目的数据,包括:

- PV(某阶段计划要求完成的工作量所需的预算费用);
- AC(某阶段实际完成的工作量所消耗的费用);
- EV(某阶段按实际完成工作量按预算定额计算出来的费用)。

(2) 计算评价指标,这些指标主要用于评价进度偏差及成本偏差:

- $CV(\text{费用偏差}) = EV - AC$ ;
- $CPI(\text{费用执行指标}) = EV / AC$ ;
- $SV(\text{进度偏差}) = EV - PV$ ;
- $SPI(\text{进度执行指标}) = EV / PV$ 。

(3) 分析各阶段的评价指标评价估计成本和实际成本的偏差及原因。

组织采用以上的成本分析方法得到结果后,应考虑结果对各利益相关方的影响,并与其就处理方法达成一致意见,包括处理问题、调整估算方法和改进开发过程等。

项目结束后,成本及相关的对于组织而言具有很大的价值,应该收集并进行分析。分析的目的和角度包括:

(1) 项目评价:根据成本估算偏差及构成评估项目组预算控制的能力以及流程执行的效率。

(2) 建立或校正成本估算模型:如上文提到的成本估算方程回归分析,项目结束后产生了新的成本及相关数据,这些数据可以用于评价回归方程的效果,并可以帮助不断优化回归方程。

(3) 过程改进:通过分析成本分布占比和各类活动成本估算偏差率等数据了解开发过程的问题,将这些数据与经验以及对组织的了解相结合,可以为管理者提供过程改进的信息。

项目规模、工作量、工期、成本等估算及实际数据还应该保存在组织内部建立的基准数据库中,以供未来项目组以及组织使用,使用的实际包括:

(1) 提供同类项目估算时参考;

- (2) 建立、评价及优化成本估算模型；
- (3) 对质量问题进行相关性分析；
- (4) 计算单位规模基准成本；
- (5) 分析组织各活动成本占比等。

组织还可以将项目组的数据提交到行业基准数据库中,为行业基准数据的不断更新提供支持。

## 3.4 标准的应用及附录

### 【标准原文】

不同利益相关方由于目的不同,宜采用的成本度量方法或过程会有所差异。本标准根据软件研发生存周期对应用场景进行划分,典型应用场景划分如下:

- (1) 预算；
- (2) 招投标；
- (3) 项目计划；
- (4) 变更管理；
- (5) 结算/决算/后评价。

关于上述 5 种应用场景的成本度量过程和要求详见附录 A。

## 附录 A

(规范性附录)

典型应用

### A.1 预算

#### A.1.1 应用范围

本标准在预算阶段的应用主要指委托方为确定项目预算而进行的成本估算活动。

#### A.1.2 依据

制定预算应依据:

- 5.1 的规定；
- 项目范围描述；
- 国家或省级、行业软件主管部门发布的相关指导办法；

- 权威机构发布的行业基准数据和人力成本基准费率相关信息；
- 委托方同类项目的基准数据；
- 其他相关资料。

#### A.1.3 估算

应由具备本标准涉及的成本估算能力的人员按照 5.1 的规定进行估算。

在预算阶段,如果需求极其模糊或不确定,可采用基准比对方法,直接估算工作量、工期、成本。

完成成本估算后,应考虑行业的平均毛利率及维护要求等因素,计算出项目的预算范围。

#### A.1.4 上报预算

应以估算的结果为基础,并根据以下因素确定上报的预算额度:

- 需求变更的风险；
- 质量要求；
- 工期约束。

例如:当项目的需求相对明确且无其他特殊要求时,上报的预算可考虑采用估算结果的中值,即 50 百分位数;如需求不明确或有较高质量、工期约束时上报的预算可考虑采用估算结果的悲观值,即 75 百分位数。

对于需求相对明确的项目,上报预算时宜附上功能清单及对应功能点数。

#### A.1.5 审批预算

审批预算时应考虑以下因素:

- 预算的合理性；
- 可用于本项目的资金情况。

预算审批人应依据 5.1 的规定对预算的合理性进行评估,也可委托第三方机构进行评估。

如果预算审批不通过,则应将预算驳回,并要求重新进行预算。

## A.2 招投标

### A.2.1 应用范围

本标准在招投标过程中的应用主要包括:

- 招标方进行的成本估算；
- 评标基准价的设定；
- 投标方进行的成本估算和项目报价；
- 评标及合同签订。

对于采用非招标方式进行采购的委托方,宜参照本标准进行成本估算并确定合理采购价格范围。

对于采用非投标方式提供报价的开发方,宜参照本标准进行成本估算和项目报价。

#### A.2.2 招标

##### A.2.2.1 招标准备

确定详细的工作说明书,工作说明书应能满足已选定的规模估算方法所需的功能点计数要求。

##### A.2.2.2 估算

应由招标方(或受其委托的第三方机构)中具备本标准涉及的成本估算能力的人员按照5.1的规定进行估算。

进行成本估算应依据:

- 5.1的规定;
- 工作说明书;
- 国家或省级、行业软件主管部门发布的相关指导办法;
- 权威机构发布的行业基准数据和人力成本基准费率相关信息;
- 其他相关资料。

并考虑以下因素:

- 项目和潜在投标人所在地域;
- 项目所需技术要求和所属领域的应用成熟度。

招标方(或受其委托的第三方机构)完成成本估算后,应考虑行业的平均毛利率及维护要求等因素,计算出合理招标价区间。

如招标阶段的工作说明书与预算阶段约定的范围没有实质性变化,则可直接采用预算阶段的估算结果。

##### A.2.2.3 设定评标基准价/投标最低合理报价/投标最高合理报价

招标方应遵循以下原则设定评标基准价、投标最低合理报价和投标最高合理报价:

- 投标最低合理报价宜参考合理招标价区间的下限值设定;
- 投标最高合理报价宜参考合理招标价区间的上限值或项目预算值;
- 评标基准价宜采用合理招标价的中值或各投标人有效报价的平均值,有效报价指投标最低合理报价和投标最高合理报价之间的报价;
- 也可根据合理招标价区间和估算规模,计算出合理的功能点单价区间,并据此设定评标基准价、投标最低合理报价和投标最高合理报价;

——可根据行业竞争状况及潜在投标人的情况对评标基准价、投标最低合理报价和投标最高合理报价进行适当调整。

招标方应基于评标基准价制定价格评分方法。

#### A.2.2.4 形成招标文件

招标方应根据 A.2.2.2 的估算结果和 A.2.2.3 的设定价格形成招标文件相应部分的内容。

招标文件中宜明确投标方所需采用的规模估算方法、评标基准价的设定方法及投标报价的评分方法。

### A.2.3 投标

#### A.2.3.1 投标准备

投标方接到招标文件后,应对招标文件中与投标报价相关的内容进行澄清和确认,明确项目的范围和边界,并结合自身经验和项目实际情况整理出功能清单及对应功能点数。

#### A.2.3.2 估算

应由具备本标准涉及的成本估算能力的人员按照 5.1 的规定进行估算。

投标方进行成本估算应依据:

- 5.1 的规定;
- 工作说明书;
- 国家或省级、行业软件主管部门发布的相关指导办法;
- 本组织的基准数据和人力成本基准费率相关信息;
- 权威机构发布的行业基准数据和人力成本基准费率相关信息;
- 招标文件要求;
- 其他相关资料。

并应考虑以下因素:

- 本组织及项目所在地域;
- 项目所需技术的要求和本组织的技术积累。

#### A.2.3.3 确定投标报价

投标方不得以低于成本的报价竞标。投标方在确定投标报价时,应依据 A.2.3.2 的估算结果并考虑如下因素:

- 期望的利润水平;
- 商业策略;
- 行业同类项目的成本水平;
- 其他相关因素。

#### A.2.3.4 形成投标文件

投标方应根据 A.2.3.2 的估算结果和 A.2.3.3 确定的投标报价,形成投标文件中相应部分的内容。

投标文件中应包含功能清单及对应功能点数。

#### A.2.4 评价

根据 A.2.2.3 确定的价格制定评分方法并对有效报价进行价格评分。

对低于投标最低合理报价或高于投标最高合理报价的情况,应视为不合理报价,价格评分宜为 0 分。

### A.3 项目计划

#### A.3.1 应用范围

本标准在项目计划活动的应用主要包括:

- 开发方获得委托方正式的委托后,为制定详细的开发计划而开展的成本估算活动;
- 开发方在项目研发过程中,根据新的信息或项目变化重新进行的成本估算活动。

#### A.3.2 依据

在项目计划时,进行成本估算应依据:

- 5.1 的规定;
- 已确认的项目工作说明书;
- 国家或省级、行业软件主管部门发布的相关指导办法;
- 本组织的基准数据和人力成本基准费率相关信息;
- 权威机构发布的行业基准数据和人力成本基准费率相关信息;
- 其他相关资料。

#### A.3.3 估算

在项目计划时,进行成本估算应遵循以下原则:

- 应由开发方或第三方机构中具备本标准涉及的成本估算能力的人员按照 5.1 的规定进行估算;
- 估算人员还应对各任务的工作量、工期分别进行估算,估算时宜参考基准数据将已估算出的总工作量、总工期分解到各任务,并依据经验或采用专家评审方法对估算结果进行验证,不同估算方法产生的结果偏差较大时应分析原因并调整估算;
- 当估算结果与项目约束产生冲突时,应分析原因并提出处理建议。

#### A.3.4 制定项目计划

制定项目计划应以 A.3.3 的估算结果为基础,并适当调整。对每一任务的资源、时间计划进行调整时应考虑的因素主要包括:

- 交付时间要求;
- 任务难度;
- 是否属于关键路径;
- 资源限制。

项目计划应获得主要利益相关方的确认并达成一致。

#### A.3.5 维护项目计划

在项目研发过程中,在以下两种情况应重新进行成本估算并维护项目计划:

- 项目到达重要里程碑或发生变化时。例如,在需求分析完成后,可重新进行规模估算,必要时对工期、工作量、成本估算进行相应调整;
- 当成本估算的假设条件发生变化时。例如,对于迭代开发的项目,如果第一次迭代的生产率数据与估算时参考的生产率数据有较大偏差,可根据实际生产率数据重新修正成本估算结果。

### A.4 变更管理

#### A.4.1 应用范围

本标准在变更管理的应用主要指项目研发过程中,由变更引起的成本估算活动。

#### A.4.2 依据

进行变更成本估算应依据:

- 5.1 的规定;
- 国家或省级、行业软件主管部门发布的相关指导办法;
- 委托方、开发方及其相关方共同明确的变更范围;
- 组织关于变更过程的经验和数据;
- 本组织的基准数据和人力成本基准费率相关信息;
- 权威机构发布的行业基准数据和人力成本基准费率相关信息;
- 其他相关资料。

#### A.4.3 估算

变更成本估算应遵循以下原则:

- 应由具备本标准涉及的成本估算能力的人员按照 5.1 的规定进行估算;
- 委托方、开发方及相关方应对变更的范围达成一致;
- 估算人员应识别变更给成本所带来的影响。按照 5.1 的规定,估算变更

的规模、工作量、工期和成本；

- 变更成本估算结果应得到委托方、开发方及相关方的评审和确认,达成共识。当不能达成一致时,委托方、开发方及相关方应进行磋商,确定解决办法。

## A.5 结算/决算/后评价

### A.5.1 应用范围

本标准在结算/决算/后评价阶段的应用主要包括：

- 为编制结算/决算而进行的成本测量；
- 为绩效评价和过程改进等后评价活动而进行的成本数据的测量和分析。

### A.5.2 依据

进行结算/决算/后评价时应依据：

- 5.2、5.3 的规定；
- 最终的功能清单及对应功能点数；
- 预算/项目计划；
- 国家或省级、行业软件主管部门发布的相关指导办法；
- 本组织的基准数据和人力成本基准费率相关信息；
- 权威机构发布的行业基准数据和人力成本基准费率相关信息；
- 其他相关资料。

### A.5.3 结算/决算

项目的成本测量应由具备本标准涉及的成本估算能力的人员进行。

在项目验收之后应依据本标准 5.2 的规定对项目的成本进行测量,作为项目结算或决算的一部分。

### A.5.4 绩效评价

委托方应依据本标准 5.2 的规定对项目的成本进行测量,宜将项目测量的规模、成本、工期与预算进行对比,全面掌握和评价项目预算的执行情况。

开发方应依据 5.2 的规定对项目的成本进行测量,宜将项目测量的规模、成本、工期与项目计划进行对比,全面掌握项目计划执行情况,考核项目实施效果。

### A.5.5 过程改进

可将项目测量的规模、成本、工期、生产率等数据纳入组织或行业的基准数据库,为以后类似项目的成本估算提供参考数据。

可将测量的功能点耗时率、生产率等数据与组织或行业的基准数据进行比对分析,以发现改进机会。

### 【标准解读】

SJ/T 11463—2013 中第 5 章详细介绍了在软件研发项目过程中对规模、工作量、工期和成本等内容进行估算、测量和分析的基本知识。但对于不同的组织或者不同的项目而言,因为不同利益相关方的目的会有所不同,因此具体的度量方法或过程也会有所差异。

为了加强对现实工作的指导意义,SJ/T 11463—2013 中第 6 章介绍了 5 种典型的应用场景,并在后续的附录 A 中详细说明了这 5 种应用场景的成本度量过程和要求。

下列 5 种典型应用场景实践对于计划在本组织中应用 SJ/T 11463—2013 的个人或团队来说,有非常重要的参考意义。开发方和委托方可以根据实际情况,在不同场景下选择预估或估算功能点方法来计算规模。

这 5 种应用场景分别是:

(1) **预算**: 主要指委托方为确定项目预算而进行的成本估算活动。在这种场景下,需求较为模糊,更适合采用预估功能点方法,只对待开发系统中的逻辑文件数量进行计算。计算出待调整功能点后,采用固定系数对规模进行调整,得出调整后的功能点数,然后主要考虑软件因素进行调整,得出工作量估算值,再根据模型估算出工期和成本。在预算场景下,为了确保项目经费,一般使用成本估算的上限值作为预算值。

(2) **招投标**: 在招投标阶段,委托方和开发方对于项目边界和需求有了更加清晰的界定,但是清晰程度有限,所以可参考预算场景下的方法采用预估功能点方法进行估算。在得到未调整功能点后,需要考虑开发因素和软件因素,综合开发方的生产率数据,估算工作量、工期和成本的范围。便于委托方还可确定合理采购价格范围,而开发方则可用来确定项目报价。

(3) **项目计划**: 开发方获得委托方正式的委托后,为制定详细的开发计划而开展的成本估算活动;此时开发方已经获得了较为详细的需求,因而可以使用估算功能点方法进行估算,在此场景下进行规模、工作量、工期成本估算的目的,更多地为后续的项目管理活动建立基准。

(4) **变更管理**: 主要指项目研发过程中,由变更引起的成本估算活动。可使用估算功能点方法对变更进行评估,从而估算变更引起的成本。

(5) **结算/决算/后评价**: 为编制结算/决算而进行的成本测量;或为绩效评价和过程改进等后评价活动而进行的成本数据的测量和分析。应用中可根据实际情况采取预估或估算功能点方法。