软件成本进度预计模型概述及 Sage 模型

黄锡滋　陈光宇
(成都电子科技大学)

摘要 正确估计软件开发成本和进度，是软件开发企业和研制管理部门必须解决的重要课题。通过介绍美国软件成本进度预计模型的发展状况和其中一个典型的 Sage 模型，对我国软件开发和研制管理提供参考。

关键词 软件开发　软件采办　软件成本进度预计模型

现代高技术信息化设备，毫无例外包含了大量指挥控制和通信软件，并构成这些高技术装备的核心。现在高技术装备软件的功能和规模日趋复杂和庞大,用于与软件有关的采办费用随之急剧增加。怎样正确估计软件开发成本和进度，成为软件开发企业和研制管理部门必须解决的重要课题。本文介绍美国软件成本进度预计模型的发展状况和其中一个典型的 Sage 模型。

软件开发成本,涉及到相关的计算机、场地等固定资产的投入和消耗，开发和研制管理部门对这一部分成本已经具有丰富的核算经验，不存在任何困难,所以关键的问题在于怎样正确估算软件开发的,高技术人力资源投入和消耗的时间。各种软件成本和进度预计模型,都是围绕这个主题展开。

1 软件成本进度预计模型的起源和发展

软件成本和进度模型化的研究，始于上世纪 70 年代,1974 年，美国 TRW 公司的 Ray Wolverton 在 IEEE Transactions on Computer 发表了软件成本研究文章,奠定了 Constructive Cost Model (COCOMO)模型的基础。

1977 年 Doty 联盟的 Herd, J,R. 等,为美国空军完成了题为"软件成本研究"的报告,为空军的软件开发和采办提供了分析和评估方法。

1974 年到 1981 年间，若干现在仍然使用的成本预计模型原型陆续出现。此后直至 1995 年,其发展基本上局限于原有基础上改进算法,没有实质性改变。

随着软件开发技术的发展和多年的数据积累，1995 年后又出现了若干新的模型，例如 Sage 模型，COCOMO Ⅱ模型和 Cost Xper 模型等，这些模型继承和改进了早期的模型和方法，提高了模型的使用价值,深受采办机构的欢迎,并一直援用至今。

2 软件成本进度预计模型的基本框架

各种软件成本进度预计模型从其框架及其共性的角度看,可区分为一阶模型,二阶模型和三阶模型。

2.1 一阶模型

一阶模型属于最基本型。其结构非常简单,仅由两项构成。第一项是生产效率系数,代表软件开发机构开发软件的能力,第二项是软件的结构参量,可以是开发软件的源代码行数 (SLOC) 或功能点数 (FPs) 等，但是最常用的参量是有效源代码行数 (ESLOC)。需要投入的人力资源和时间用人-时表示。对于一阶模型,其基本关系式是:

$$E_d=G_kS_e \qquad (1)$$

其中,

E_d:用人-时(PH)表示的开发投入。

G_k:生产效率系数(PH/ESLOC)。

S_e:ESLOC 数

生产效率系数通常是由产品类型，历史上的开发能力,或根据实践经验综合上述因素导出。这个简单的公式,广泛地用来对开发投入做出粗略的估计。

70 年代末期，这个公式的一个拓展获得了较多应用:

$$E_d=C_k(S_{new}+0.75S_{midified}+0.2S_{reused})PH \qquad (2)$$

在这个扩展公式中，用不同的系数区分了新编制的代码，修改的代码和重用的代码对人力资源投入的影响。

一阶模型简单、易用,缺点是对于程序规模变化导致的资源需求反应迟钝,与实际情况有差距。

2.2 二阶模型

二阶模型的特点是,引入熵系数,用来反映程序规模增大后,导致生产效率下降的现象。

程序规模增大的必然后果是，开发人员需求数量增加,导致程序开发过程中的通讯通道急剧增加。对于由 n 个开发人员组成的开发团队，其通讯通道为 n(n-1)/2。

二阶模型的典型形式是:

$$E_d=C_kS^{\beta}_e \tag{3}$$

其中,β 是熵系数。

熵系数等于 1 表示生产效率内不随程序规模发生变化,熵系数小于 1 表示程序规模增大,生产效率增加,与实际相矛盾,所以 β 取值应大于或等于 1。熵系数大于 1 表示程序规模增大,生产效率下降。在上世纪 80 年代,美国使用的各种成本进度估计模型中,采用的熵系数为 1.2 。

二阶模型的主要缺点是，无法体现开发环境变化对生产力的影响。

2.3 三阶模型

三阶模型中引入了范围广泛的若干环境系数,用来调整开发环境变化对生产效率的影响。三阶模型的典型形式如下:

$$E_d=C_k(\prod_{i=1}^{n}f_i)S^{\beta}_e \tag{4}$$

式中,f_i代表了各种环境因素对生产力的影响。环境因素的类型和数目并无定论，由各个开发机构根据企业的实际情况选取。

3 Sage 模型

Sage 模型有若干不同的名称，最初的全称是 Software Evaluation and Estimation of Resources - Software Estimating Model, 简称为 Seer-Sem 模型。模型最早由美国休斯飞机公司空间和通讯部利用军事装备的数据导出和使用,由 Jensen 于 1980 年在一个国际会议上发表,所以又名 Jensen 模型。1995 年,模型在美国软件工程研究所 Sage 软件估计系统中实施,进行了重要的改进,于是又被称为 Sage 模型。

3.1 影响软件开发的三个因素

在软件的开发过程中,人、过程、产品是三个重要的影响因素。这个模型的特点是全面地考虑了 3 个因素的影响,把他们用环境系数的方式,综合进模型中。

(1) 人:泛指各种开发人员的属性,例如开发人员的能力,开发人员的积极性,开发人员相互联络和沟通的能力。

(2) 过程:包括软件开发方法,应用工具,实践。

(3) 产品:包括项目的类型和对产品提出的特殊要求和约束,例如遵循的开发标准、存储要求、时间约束和安全性要求等。

软件开发过程属于脑力劳动过程，对于复杂软件系统，各个分系统之间存在错综复杂的联系和耦合,必需依靠开发团队的集体智慧来解决。这时软件开发的效率，主要依赖于开发人员之间的联系和沟通，这些沟通的方式和渠道又取决于机构的结构和管理方法,所以也是影响软件开发的重要环境因素。

3.2 模型的基本关系式

模型的基本关系式是从大量实际数据中总结导出,它由下列三个关系式构成:

$$S_e=C_{te}\sqrt{K}T_d \tag{5}$$

其中,

S_e:ESLOC 数

C_{te}:开发活动有效技术常数

K:产品寿命周期总投入,用从产品需求分析和定义开始直至产品退役为止的总投入人-年(PY)表示。

T_d:软件产品开发时间

$$D=\frac{K}{T_d^3} \tag{6}$$

其中,D:产品困难性参数

用开发投入表示从需求分析开始，至最终认证测试为止的总投入。

$$E_d=0.3945K \tag{7}$$

其中,E_d表示用人-时(PH)表示的开发投入。

3.3 Sage 模型的完整表达式

从 式(5)、式(6)、式(7)中解出

$$E_d=\frac{18.797D^{0.4}}{C_{te}^{1.2}}S_e^{1.2}\quad 人-月(PM) \tag{8}$$

3.4 开发活动有效技术常数的确定

开发活动有效技术常数C_{te}由基本技术常数和环境因素系数构成,其关系式为:

$$C_{te}=\frac{C_{tb}}{\prod_{i=1}^{24}f_i} \tag{9}$$

其中,C_{tb}是基本技术常数,代表开发机构基本开

发能力，即独立于特定程序的开发能力。

C_{tb} 的取值为 2000 至 20000。在美国软件开发机构中，正常范围为 5500 至 7500，最高达到 8650。

C_{te} 的理论值为 0 至 20000，对美国软件开发机构，其实际下限在受到产品类型和 其他诸多约束时约为 500。

式(9)中的 f_i 表示第 i 个环境系数。环境系数的类型，各开发机构不尽相同，其具体数值代表了其开发环境的水平，在公开刊物上无法获悉。表 1 是某个机构对环境系数之一的开发人员能力的评分度量。

表 1 开发人员和机构能力度量

定义	具有高度的积极性和有经验的开发机构团队	具有较高的积极性或有经验的开发机构团队	传统的开发机构	积极性较差或非关联机构	积极性缺乏和非关联机构
对成本的相对影响	0.71	0.86	1.00	1.19	1.46

综合式(8)和式(9)可以看出，Sage 模型属于典型的三阶模型。 ◇

参考文献：

[1]Wolverton, R.W. The Cost of Developing Large-Scale Software. IEEE Transactions on Computers June 1974: 615-636.

[2]Boehm, B.W. Software Engineering Economics. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1981.

[3]Herd, J.R., J.N. Postak, We. E. Russell, and K.R. Stewart. Software Cost Estimation Study - Final Technical Report. Vol. I. RADC-TR-77-220. Rockville, MD: Doty Associates, Inc., June 1977.

[4]Jensen, R.W. *Management Impact of Software Cost and Schedule.* CrossTalk July, 1996:6.

[5]Galorath, Inc. SEER-SEM Users Manual. El Segundo, CA: Galorath Inc., Mar. 2001.

[6]Software Engineering, Inc. Sage User′ s Manual. Brigham City, UT: Software Engineering, Inc., 1995.

（上接第 14 页）

对密封性能存在潜在影响，因此对于分离前有密封要求的情况要慎重选用。

与火工分离推杆相比，弹簧分离推杆具有性能稳定、易于地面检测控制、工作时冲击小等优点，但体积和重量较大。

3.5 热刀

热刀是一种特殊的非火工解锁装置，通过张紧绳压紧目标体。在解锁时，热刀的刀体电热元件通电升温(达 1000℃)，与电热元件紧密接触的张紧绳局部强度逐渐衰减，最后在预张力的作用下被拉断，目标体被释放。

热刀工作时的冲击很低，安全可靠。

4 结论

本文介绍的解锁分离装置，几乎都在航天器上实现了成功应用，有的已经非常成熟。在航天器研制过程中，充分了解、掌握这些解锁分离装置的特点，对于解锁分离装置的正确选择，或新型解锁分离装置的方案确定，具有积极的指导和借鉴意义。 ◇

参考文献

[1]杨建中，娄汉文等.航天器用可解锁连接分离装置.航天器工程，2003(1).

[2]袁家军.卫星结构设计与分析.北京：宇航出版社，2004.

[3]К.С.КОЛЕСНИКОВ, Динамик разделения ступеней летательных аппаратов, Москва МАШИНОСРОЕНИЕ, 1977.

[4]L Bement, H Multhaup, Determining Functional Reliability of Pyrotechnic Mechanical Device, 33rd AIAA/ASME/SAE Joint Propulsion Conference&Exhibit.1997.

[5]R. Gardi, M. Pastena, A Non- Pyrotechnic, SMA Based Release Mechanism for Separation System. 52nd International Astronautical Congress.

[6]J.Vazquez, J. I. Bueno, Non Explosive Low Shock Reusable 20 Hold-down Release Actuator,Proceedings of 9th European Space Mechanism and Tribology Symposium,2001.

[7]杨建中，娄汉文.保证神舟飞船机构可靠性的若干设计措施.载人航天，2006(4).