第5章 软件生产过程经济分析

5.1 软件生产函数与软件生产率

5.1.1 软件生产函数及其特性

生产函数是宏观经济学(Macro Economics)和微观经济学(Micro Economics)理论中的一个重要概念，它是研究系统规模变化对产出的影响和最优化经济效果的基础。不少西方学者在此领域内进行了深入的研究，如1972年诺贝尔经济奖获得者英国经济学家希克斯(J.M.Hicks)以及美国经济学家阿罗(K.J.Arrow)，1987年诺贝尔经济奖获得者、美国经济学家索洛(R.M.Solow)以及美国数学家柯布(C.W.Cobb)和经济学家道格拉斯(P.H.Douglas)等。他们的杰出贡献在于对人类生产过程的规律性进行了总结和数量描述，并据此作了进一步的规律性的探索。

所谓生产函数(Production Function)，是指反映生产过程中投入要素与其可能生产的最大产量之间依存关系的数学表达式。早期的生产函数有如下数学形式：

$$Y=F(K，L，N，O，t)$$

式中，Y为产出量，如宏观经济系统中的GDP、工业总产值，微观经济系统中的企业产品的产量、产值、销售收入等；K、L、N、O分别表示生产过程投入的资本、劳动、土地和组织管理要素投入量；t表示时间或工期等。鉴于土地投入量的变化很小，而且在非农业部门中，一般已将土地的价值计入资本之中，而组织管理又难以定量，因此为了简化分析，以后研究的生产函数常记为

$$Y=F(K，L，t) \tag{5.1}$$

1. C-D生产函数

西方学者在采用计量经济学的有关统计法的研究中提出了多种形式的生产函数，如线性生产函数、前沿生产函数、C-D生产函数等，它们分别从不同的侧面反映了西方国家生产过程中的工程经济行为。以下介绍由柯布(C.W.Cobb)和道格拉斯(P.D.Douglas)合作提出的C-D生产函数，其数学表达式为如下形式：

$$Y=AL^{\alpha}K^{\beta} \qquad (5.2)$$

其中，Y为产出量；L为劳动投入量；K为资本投入量；A为除劳动与资本两要素外其他对产出Y的总影响，注意到(5.2)式两边分别对K和L求偏导数有

$$\frac{\partial Y}{\partial K} = \beta A L^{\alpha} K^{\beta-1} = \beta \frac{Y}{K}, \qquad \frac{\partial Y}{\partial L} = \alpha A L^{\alpha-1} K^{\beta} = \beta \frac{Y}{L}$$

或有

$$\alpha = \frac{\partial Y}{\partial L} \cdot \frac{L}{Y} \approx \frac{\dfrac{\Delta Y}{Y}}{\dfrac{\Delta L}{L}}, \qquad \beta = \frac{\partial Y}{\partial K} \cdot \frac{K}{Y} \approx \frac{\dfrac{\Delta Y}{Y}}{\dfrac{\Delta K}{K}} \tag{5.3}$$

式中，α称为劳动力对产出的弹性系数；β称为资本(资金)对产出的弹性系数。由(5.3)式可知，α表述了劳动力增加百分之一会使产出变动的百分比；β表示了资本增加百分之一会使产出变动的百分比，它们分别反映了在其他条件不变的条件下产出对劳动力变化或资本变化的反应程度。生产函数一般都满足如下特性：

(1) 资本与劳动力的边际产出均为正值，即有

$\frac{\partial Y}{\partial L} > 0, \quad \frac{\partial Y}{\partial K} > 0$ ，其经济含义为劳动力(或资本)投入量不变的情况下，资本(或劳动力)投入量的增加，将导致产出量的增加。

(2) 边际产量递减，即有 $\frac{\partial^2 Y}{\partial L^2} < 0, \quad \frac{\partial^2 Y}{\partial K^2} < 0$ ，其经济含义为当其他生产要素固定不变时，随着某一要素投入量的增加，边际产量将逐渐减少。

(3) 生产函数具有非负性，即总产出为正值，且总产量是生产要素组合的结果，单一生产要素的投入不能获得产出。用数学语言来描述，即有

$$Y=F(K，L，t)>0，F(K，0，t)=F(0，L，t)=0$$

2. 规模经济

规模经济(Economics of Scale)或规模报酬是微观经济学中研究的一个重要问题，它表示当生产规模变化时，对产出的影响程度。规模报酬一般有三种情况：当全部投入要素按某种配合方式以相同比例增加时，如果产出的增长比例大于投入要素配合方式增加的比例，则称企业(或厂商)享有递增规模报酬(或规模经济)；如果产出的增长比例小于投入要素配合方式增加的比例，则称为递减规模报酬(或非规模经济Diseconomies of Scale)； 如果产出的增长比例等于投入要素配合方式增加的比例，则称为固定规模报酬。上述经济概念的数学描述如下。

对于任何$\lambda>1$，以生产函数$Y=F(Y，L，t)$描述的生产活动有下述三种情况：

(1) 若$Y=F(\lambda K，\lambda L，t)>\lambda F(K，L，t)$，则称该生产活动呈规模报酬递增或规模经济；

(2) 若$Y=F(\lambda K，\lambda L，t)<\lambda F(K，L，t)$，则称该生产活动呈规模报酬递减或非规模经济；

(3) 若$Y=F(\lambda K，\lambda L，t)=\lambda F(K，L，t)$，则称该生产活动呈规模报酬固定。

注意到将(5.2)式代入生产函数$Y=F(K，L，t)$有

$$F(\lambda K，\lambda L，t)=A(\lambda L)^{\alpha}(\lambda K)^{\beta}=\lambda^{\alpha+\beta}AL^{\alpha}K^{\beta}=\lambda^{\alpha+\beta}Y \qquad (5.4)$$

因此对C-D生产函数容易验证有如下结论：

(1) 若$\alpha+\beta>1$，则该生产活动呈规模报酬递增(规模经济)；

(2) 若$\alpha+\beta<1$，则该生产活动呈规模报酬递减(非规模经济)；

(3) 若$\alpha+\beta=1$，则该生产活动呈规模报酬固定。

3. 弹性系数的求解

形如(5.2)式所示的C-D生产函数是一种较为普遍的生产过程中生产行为之规律性描述，但对于一些不同的企业(部门、地区)而言，由于其外部环境与内部条件的不同，则这些企业(部门、地区)的产出对资本与劳动投入的反应程度应该不同。从数学描述来看，不同企业的C－D生产函数应有不同的α与β。因此对于一个特定的企业(或行业部门)，求解其对应的弹性系数α与β就成为必要。

注意到(5.2)式中实际上Y、K、L、A均为时间t的函数，不妨设为y_t，k_t，l_t，$a(t)$，则有

$$y_t = a(t)\cdot l_t^{\alpha}\cdot k_t^{\beta}$$

对上式两端分别求对数有

$$\ln y_t = \ln a(t) + \alpha \ln l_t + \beta \ln k_t$$

令$Y_t=\ln y_t$，$A(t)=\ln a(t)$，$L(t)=\ln l_t$，$K(t)=\ln k_t$，则有

$$Yt = A(t) + \alpha \cdot L(t) + \beta \cdot K(t) \qquad (5.5)$$

显然，对于一个特定的企业(或产业部门)，若其历史时间序列$\{y_t$，$t=1$，2，…，$n\}$，$\{l_t$，$t=1$，2，…，$n\}$，$\{k_t$，$t=1$，2，…，$n\}$已知，则由(5.5)式知可通过二元线性回归的方法求解$A(t)$、α、β的估计值，从而解决了两个弹性系数α与β的求解问题。有关的多元线性回归方法可详见作者文献[16]。

4. 软件生产函数

运用多元线性回归方法，美国的软件工程专家普特纳姆(L.H.Putnam)根据所掌握的英、美软件项目的时间序列推导出具有如下形式的软件生产函数：

$$S = E \cdot K^{\frac{1}{3}} \cdot t_d^{\frac{4}{3}} \tag{5.6}$$

式中，S表示软件生产规模或软件生产的源代码程序量(单位为非注释性语句数量NCSS)；K为软件项目在生存期内投入的总工作量(单位为人年，Person Year，PY)；t_d为软件项目投入人力的峰值时间(通常为交付期或工期，单位为年)；E称为环境因子，它表示除K与t_d外，软件项目的其他投入要素对产出总量S的影响程度。

由(5.6)式容易验证其满足前述一般生产函数的三个特性中的大部分，即在软件生产过程中，软件工作量或交付工期的边际生产量(交付的源代码程序量)均为正，或有

$$\frac{\partial S}{\partial K} > 0, \quad \frac{\partial S}{\partial t_d} > 0$$

同时满足 $\frac{\partial^2 S}{\partial K^2} < 0$，但却有 $\frac{\partial^2 S}{\partial {t_d}^2} > 0$（即不满足 $\frac{\partial^2 S}{\partial {t_d}^2} < 0$ 之特性），此外还满足

$F(K,0,t) = F(0,t_d,t) = 0$ 之特性。

注意到由（5.6）式可知，软件生产函数中的两个弹性系数有 $\alpha=\frac{1}{3}$，$\beta=\frac{4}{3}$，并有 $\alpha+\beta=\frac{5}{3}>1$，从而可知软件生产活动具有规模报酬递增或规模经济效应。

5.1.2 软件生产率及其影响因素

劳动生产率是任何一个企业在产品生产过程中对生产人员的效率度量指标或整个产品生产效率的度量指标。也是对每个生产人员绩效考核的基础和管理人员跟踪与控制生产进度的主要依据。软件劳动生产率或简称软件生产率(Software Productivity)，被定义为每个人月(Person Months，PM)所交付的源代码程序量(单位：NCSS/PM)。软件生产率的提高对减少成本、增大利润、缩短工期具有重大的作用。因此研究软件生产率的影响因素，寻找提高软件生产率的途径显然具有重要的实用价值。

大量的工作实践与实验研究证实：影响整个产品的软件生产率提高的因素主要有两类: 第一类因素是组织与管理因素，如用人不当、缺乏沟通、管理不善、未发挥团队人员的积极性、缺乏必要的业务规范和科学合理的激励与约束机制等；第二类是技术因素，如产品需求的复杂性和高可靠性、服务器与工作站的存取速度与运算速度、主存储器的约束、虚拟机的易变性、团队人员的能力与经验不足、需求的易变性以及恶劣的工作环境(如闷热、吵闹、过分拥挤的工作间)等都会影响软件生产率的提高。为了消除和降低上述因素的影响必然还涉及到成本增加的代价，因此如何权衡成本增加和提高软件生产率，需要进行更为细微与深入的研究。限于篇幅，本书以下给出提高软件生产率的常用的几个措施：

(1) 提高团队工作的业务规范与编程规范；

(2) 采用较为先进的软件工具(如程序库、程序生成器、模型生成器等)；

(3) 部分功能采用商业软件包(如算法软件包、数据库管理系统等)；

(4) 改编现有的已熟悉软件的部分功能；

(5) 采用软件构件技术、多版本技术和软件复用技术；

(6) 建立科学、合理的激励和约束机制；

(7) 对人员的选择采用如下五原则：顶级天才原则、任务匹配原则、职业发展原则、团队平衡原则和逐步淘汰原则。

需要说明的是，顶级天才原则的基本涵义是与其使用更多数量的一般才能人员还不如使用少量的具有更高能力的人员，因为后者对整个软件项目可以贡献更高的软件生产率而成本又不至于会有太大的提高；所谓任务匹配原则的基本内涵为量才而用，根据每个团队成员的能力与素质，分配其合适的任务；所谓职业发展原则的基本内涵为设立多种工作岗位，为工作表现优秀的团队成员职业发展提供了空间；所谓团队平衡原则的基本内涵为团队成员的选择必须要能相互信任、互相合作、彼此取长补短、协调一致地为共同的团队目标努力；逐步淘汰原则的基本内涵是将那些工作不称职(包括能力较差、完不成任务指标或影响团队和谐气氛等)的人员逐步淘汰出团队。

5.2 软件生产过程经济分析

软件系统本质上是一个人—机(软/硬件)系统，从系统的构成与应用来看，硬件是基础，软件是核心(心脏)。因此在重视硬件(计算机、通信设备、传感器等)生产的同时，开展对软件生产过程(开发过程)的技术经济分析是十分必要而有意义的工作。本节主要介绍在不同的软件类型下软件的主要工程经济参数，如生产规模、工作量、投入费用、劳动生产率、环境因子、成本等的相互数量关系，从而为软件的生产过程设计打下基础。同时上述内容的讨论也构成了软件工程经济学的核心内容之一。

5.2.1 软件生产系统动力学方程

软件作为一个特殊产品或系统，其生产过程是由一系列相互关联、相互制约的工程经济要素综合作用的结果。因此，采用系统工程的理论和方法来研究软件的生产过程是十分有益的。根据系统工程的理论，要探索一个目标系统的内在要素关联及其动态发展规律，建立该目标系统对应的系统动力学方程(System Dynamical Equation，SDE)，并以此系统动力学方程为基础来展开研究是一种有效的思路与方法。以下介绍英国软件工程专家诺顿(P.V.Noder)所提供的诺顿—瑞利模型(Noder-Rayleigh Model)及系统动力学方程的求解。N-R模型的有关变量及其经济内涵如表5.1所示。其模型假设如下：

表 5.1 N－R 模型变量表

变量符号	变 量 内 涵	单位
$C(t)$	软件工程在$[0,t)$内投入的累计人力工作量(人力费用)	人年
$m(t)$	$C(t)$的变化率或软件工程在 t 时刻投入的人力密度	人
K	软件工程项目在生存期内投入的总工作量	人年
$p(t)$	软件开发效率函数或学习函数	—
t_d	软件工程项目投入人力的峰值时刻(通常为交货期或工期)	年
D	软件工程项目开发难度系数	人/年
D_0	软件工程项目人力增长率	人/年^2
S	软件生产规模或生产的源代码程序量	NCSS
F_c	软件工程项目生产费用率	万元/人年
F_d	软件工程项目开发劳动生产率	NCSS/人年
E	软件工程环境因子	—

(1) 开发项目中需要投入的总工作量K为有限；

(2) $C(t)$在项目开始时为零，即$C(0)=0$，然后单调增长到K；

(3) 任何时刻开发项目组投入的人力数$m(t)$与尚待解决的问题(或尚需投入)的累计人力工作量成正比；

(4) 在项目生存周期中，项目开发人员由于不断的学习，因而其开发效率可用关于时间 t 的学习函数$p(t)$来描述，其开发人力量$m(t)$与$p(t)$成正比，在多数情况下，可设学习函数有$p(t)=2bt$，$b>0$，$p(t)$是时间 t 的线性增函数。

由上述假设容易建立关于累计人力工作量$C(t)$的如下一阶变系数微分方程及其初值条件:

$$\begin{cases} \dfrac{\mathrm{d}C(t)}{\mathrm{d}t} = p(t)[K - C(t)] \\ p(t) = 2bt \qquad b > 0 \\ C(0) = 0 \end{cases} \tag{5.7}$$

容易求得上述常微分方程的解为

$$\begin{cases} C(t) = K[1 - \mathrm{e}^{-bt^2}] \\ m(t) = \dfrac{\mathrm{d}C(t)}{\mathrm{d}t} = 2Kbt\,\mathrm{e}^{-bt^2} \qquad b > 0 \end{cases} \tag{5.8}$$

注意到(5.8)式的累计人力工作量$C(t)$的变化率函数$m(t)$具有概率论中瑞利(Rayleign)分布函数的形式，故$m(t)$称为诺顿—瑞利(N-R)曲线，(5.7)式又称为人力投入的系统动力学方程。

通过 $\frac{\mathrm{d}m(t)}{\mathrm{d}t}=0$ ，容易求得N-R曲线在$t_o=1/\sqrt{2b}$时取得最大点，并有最大值

$$m(t_0)=2Kb\frac{1}{\sqrt{2b}}\,\mathrm{e}^{-b\frac{1}{2b}}=K\sqrt{2b}\;\mathrm{e}^{-\frac{1}{2}} \tag{5.9}$$

显然，$m(t_o)$即为软件生存周期中的开发人员的峰值。此外还有 $m(0)=0,\lim\limits_{t\to\infty}m(t)=0$ 。

由此可知，对不同的b值(b>0)和K值，N-R曲线均为具有单峰值且自左向右由单调增到单调降的曲线。图5.1画出了当K=10时不同b值的N-R曲线。

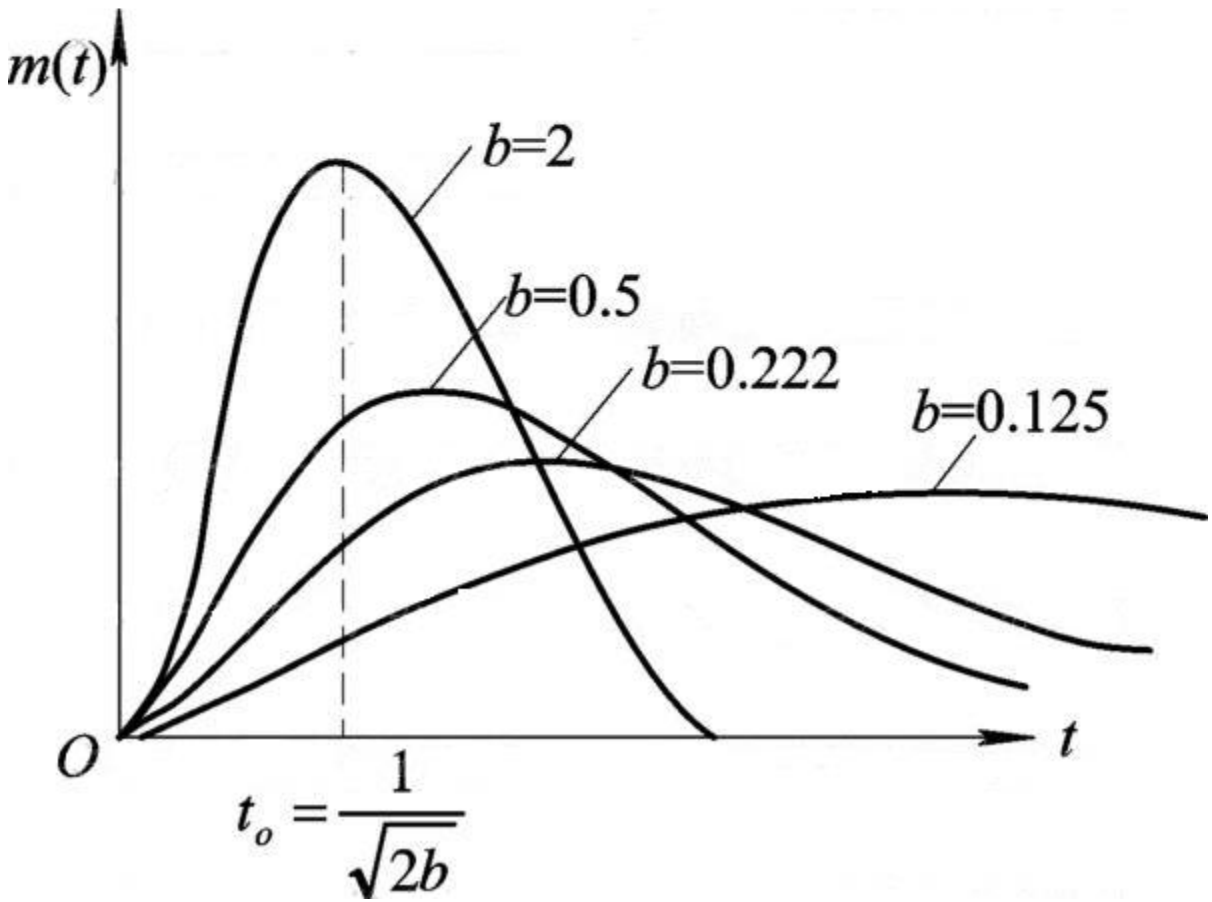

图5.1 K=10时不同b值的N-R曲线

在20世纪70年代，美军陆军中央设计处对所积累的200多个软件开发项目的数据进行了统计分析工作，其中普特纳姆(Putnam.L.H)发现，上述$m(t)$在(0，∞)中的最大点t_o非常接近交货时间t_d，这一结论的经济含义是十分明显的，因为在临近交货期时需要大量的人力资源来编制说明书，进行软件调试与质量检验，并对设计、编码等工作做再修改。注意到此时有 $t_0=\frac{1}{\sqrt{2b}}=t_d$ 或为 $b=\frac{1}{2t_d^2}$，将其代入(5.8)式和(5.9)式，有

$$C(t)=K[1-\mathrm{e}^{-\frac{t^2}{2t_d^2}}],m(t)=\frac{K}{t_d^2}t\mathrm{e}^{-\frac{t^2}{2t_d^2}},t\geq 0 \qquad (5.10)$$

$$C(t_d)=K(1-\mathrm{e}^{-\frac{1}{2}})=0.39K, m_o \overset{\text{def}}{=} m(t_o)=m(t_d)=\frac{K}{t_d}\mathrm{e}^{-\frac{1}{2}} \quad (5.11)$$

(5.11)式的$C(t_d)$说明一个开发好的软件系统在初步运行性能良好并交付给用户时只花费了生存期内投入总人力费用的39%，剩下的61%的人力费用将用于该软件系统在运行维护阶段的质量检验，可靠性增长，维护与修改等工作，而这一结论与国外软件工程的大量实践结果基本符合。

图5.2画出了$C(t)$，$m(t)$随时间t的变化曲线，由图可知$m(t)$曲线由零递增到m_o，然后再递降到零，而$C(t)$从总体上看是关于t单调增函数，但是$C(t)$在(0，t_d)区间内上升较快，而在t_d以后其增长速度放慢，最后缓慢上升到K。因此一般来说，$C(t)$曲线呈“S”形。

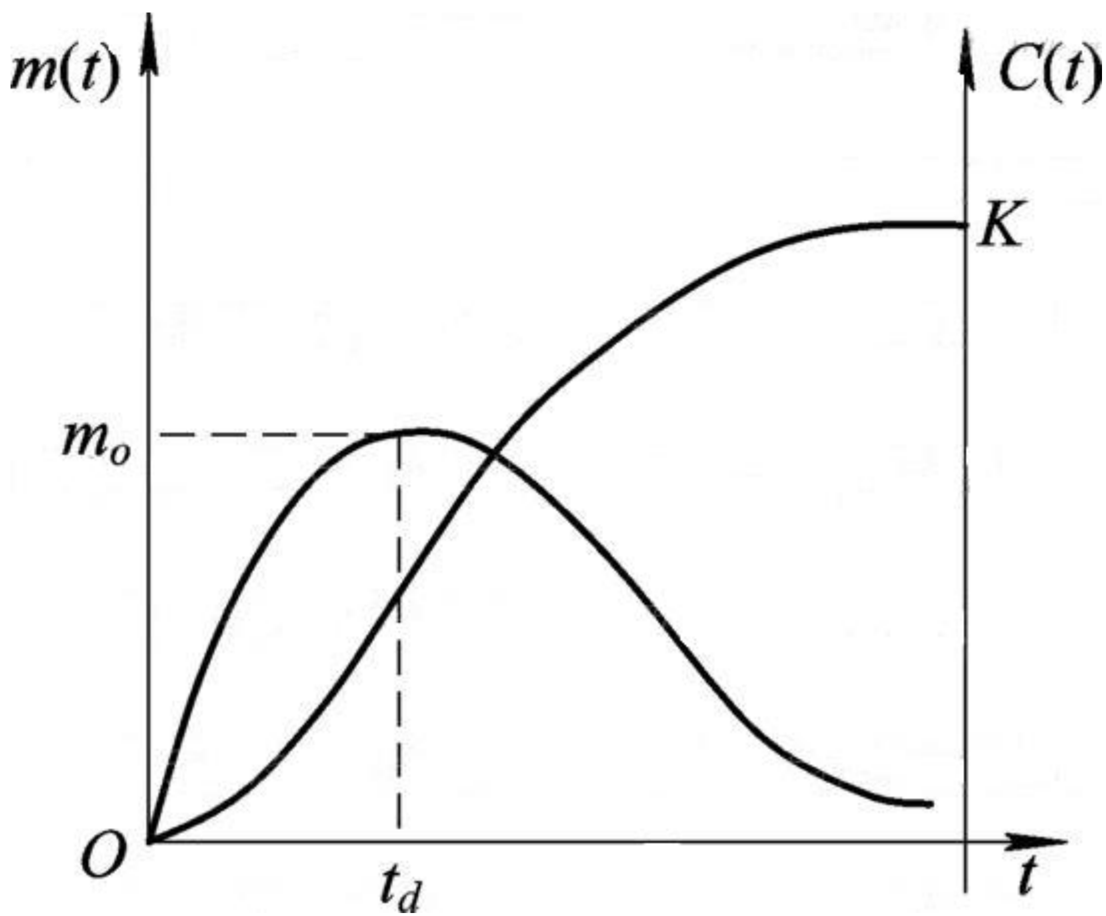

图5.2 $C(t)$及$m(t)$随t变化

5.2.2 软件项目难度系数与人力增长率

软件工程专家普特纳姆(Putnam.L.H)通过对英、美大量软件工程项目资料的研究，得到了一些经验规律性的结论。首先，他发现软件工程的开发难度与生存期内投入的总工作量K成正比，与交付期t^2_d成反比。于是他建议引入一个能用来定量描述项目开发难度的参数D，并称D为软件工程开发难度系数，且有

$$D \stackrel{\text{def}}{=} \frac{K}{t_d^2} = \frac{\mathrm{d}\,m(0)}{\mathrm{d}\,t} \tag{5.12}$$

其次，普特纳姆还发现比值D/t_d在解释软件的开发行为与项目属性方面有重要作用，对于具有同一项目开发特性的软件工程，尽管随着项目规模的增大，K和t_d均将增大。然而，比值$D/t_d=K/t^3_d$则基本上稳定在某一个常数周围，而不同项目开发属性的软件工程这样的稳定常数则会不同。于是普特纳姆据此建议引入一个被称为人力增长率的工程经济参数D_0来描述上述规律性，并给出了不同项目开发属性的稳定常数的具体数值如下：

$$D_0 \overset{\text{def}}{=} \frac{D}{t_d} = \frac{K}{t_d^3}$$

$$D_0 = \begin{cases} 8, & \text{软件是一个与其他系统有多个接口和交互功能的全新软件} \\ 15, & \text{软件是一个新的独立系统} \\ 27, & \text{软件是从其他已开发的软件基础上建立的系统} \end{cases} \tag{5.13}$$

D_0之所以称为人力增长率，是普特纳姆在研究D关于t_d的变化率时得到如下关系：

$$\frac{\frac{\mathrm{d}D}{\mathrm{d}t_d}}{-2}=\frac{\frac{-2K}{t_d^3}}{-2}=\frac{K}{t_d^3}=D_0$$

上式的D_0反映了难度(人力投入)的变化率(增长率)的概念。显然，当一个待开发软件的开发属性确定后，借助于(5.13)式中D_0的经验数据来确定K与t_d的数量关系，并进而可由给定的工期 t_d 来计算未知的K，即有$K=D_0\cdot t^3{}_d$，这对于软件工程的设计是十分有用的。当然，若已知K与t_d，则亦可由(5.13)式来求解D_0，并观察是否与(5.13)式所给出的数值相近。

[例5.1] 某软件项目，其初始人力密度增长率为4人/月，预计1年7个月后交付用户，生产费用率F_c为6万元/人年，试确定项目生存期内投入的总工作量(人力费用)，峰值人数和生存期投入总费用，项目开发难度系数和人力增长率，以及开发阶段投入的累计人力工作量和费用。

解 注意到交付期有

$$t_d = 1\text{年}7\text{个月} = 1\frac{7}{12}\ \text{年} = 1.583\text{年}$$

初始人力密度增长率由(5.12)式知，有

$$D = \frac{\mathrm{d}\,m(0)}{\mathrm{d}\,t} = 4\text{人}/\text{月} = 48\text{人}/\text{年}$$

从而可得

$$K = Dt_d^2 = 48\text{人}/\text{年}\cdot(1.583\text{年})^2 = 48\text{人}/\text{年}\times 2.5\text{年}^2 = 120\text{人年}$$

将K与t_d数值代入(5.11)和(5.13)式，有

$$m_o = m(t_d) = \frac{120}{1.583\sqrt{\mathrm{e}}} = 46\text{人}, D_0 = \frac{D}{t_d} = \frac{48}{1.583} = 30\text{人}/\text{年}^2$$

生存期投入的总费用(平均)为

$$U = F_c \cdot K = 6\text{万元}/\text{人年} \cdot 120\text{人年} = 720\text{万元}$$

此外，由(5.11)式还可得到开发阶段投入的累计人力工作量和费用U_d为

$$C(t_d) = C(1.583) = 0.39K = 0.39 \cdot 120\text{年} = 46.8\text{人年}$$

$$U_d = F_c \cdot C(t_d) = 6\text{万元}/\text{人年} \cdot 46.8\text{人年} = 280.8\text{万元}$$

5.2.3 软件的劳动生产率、生产函数及其关联

由表5.1得知S为软件工程的规模或提交的源代码程序量(单位：NCSS，表示非注释语句的数量)，而$C(t_d)$则表示在软件开发阶段所投入的累计工作量(单位：人年)，因而$S/C(t_d)$表示在软件开发阶段中单位时间所提供的源代码程序量(单位: NCSS/人年)，具有劳动生产率的概念，故人们以符号F_d来表述，并称F_d为软件项目的开发劳动生产率(简称劳动生产率)。普特纳姆通过对大量美国陆军软件工程项目的开发信息的研究发现了又一个经验规律(统计规律)，即有

$$F_d = \frac{S}{C(t_d)} = E_o \cdot D^{-\frac{2}{3}} \tag{5.14}$$

(5.14)式的工程经济意义是明显的，它表示软件工程项目难度越大，劳动生产率越低下，而其中比例系数E则反映了软件项目开发环境的技术状态。显然，在同样的软件工程项目难度下，不同的开发环境技术状态(如开发方法，开发工具，项目管理状况)亦将直接影响软件项目的劳动生产率。以(5.11)和(5.12)式代入(5.14)式，则有

$$S = C(t_d)\cdot E\cdot D^{-\frac{2}{3}} = 0.39K\cdot E_o\cdot D^{-\frac{2}{3}} = 0.39K\cdot E_o\cdot\left(\frac{K}{t_d^2}\right)^{-\frac{2}{3}}$$

$$= 0.39E_o\cdot K^{\frac{1}{3}}\cdot t_d^{\frac{4}{3}} = E\cdot K^{\frac{1}{3}}\cdot t_d^{\frac{4}{3}} \tag{5.15}$$

其中E=0.39E_o。

(5.15)式的工程经济意义亦是明显的，它反映了一个软件工程项目投入要素(投入工作量K和交付期t_d与产出要素(项目提交的生产量或源代码程序量)S的数量关系，与前5.1节一般工程经济学中生产函数的概念相一致。故人们将(5.15)式称为软件工程项目的生产函数，并将其中的系数E称为该项目的环境因子。有关环境因子的测定方法将在本节后面涉及。

以下利用软件项目生产函数做弹性分析，来讨论有关时间(工期t_d)、人力费用总量K与难度系数D的相对变化率的关联和均衡问题。

由(5.15)式显然可得 $K^{\frac{1}{3}} \cdot t_d^{\frac{4}{3}} = S/E$ ，两边取自然对数，有

$$F(K,t_d) \overset{\text{def}}{=} \frac{1}{3}\ln K + \frac{4}{3}\ln t_d = \ln\frac{S}{E}$$

当在一个特定的机构中开发一个程序量为 S 的软件产品时，$\ln\frac{S}{E}$ 应为常量，故有

$$\mathrm{d}F(K,t_d) = \frac{\partial F}{\partial K}\mathrm{d}K + \frac{\partial F}{\partial t_d}\mathrm{d}t_d = \frac{\mathrm{d}K}{3K} + \frac{4\,\mathrm{d}t_d}{3t_d} = 0$$

$$\frac{\mathrm{d}K}{K} = -4\frac{\mathrm{d}t_d}{t_d} \quad 或 \quad \frac{\Delta K}{K} = -4\frac{\Delta t_d}{t_d} \tag{5.16}$$

(5.16)式表明，若开发时间压缩10%或 $\frac{\Delta t_d}{t_d} = -10\%$ ，由于 $\frac{\Delta K}{K} = -4\frac{\Delta t_d}{t_d} = 40\%$，说明软件相应地应增长人力费用的40%。

同样，注意到有 $D=K/t^2_d$ 两边取自然对数，则有

$\tilde{F}=\ln D=\ln K-2\ln t_d$ ，又对左式两边取微分，则有

$$d\tilde{F}=\frac{\mathrm{d}D}{D}=\frac{\mathrm{d}K}{K}-\frac{2\,\mathrm{d}t_d}{t_d} \quad \text{或} \quad \frac{\Delta D}{D}=\frac{\Delta K}{K}-2\frac{\Delta t_d}{t_d} \tag{5.17}$$

若时间压缩10%，或 $\frac{\Delta t_d}{t_d}=-10\%$ ，由前面计算已知有

$\Delta K/K=40\%$，将此代入(5.17)式，则有

$$\frac{\Delta D}{D}=\frac{\Delta K}{K}-2\frac{\Delta t_d}{t_d}=40\%+2\times10\%=0.60=60\%$$

由此可见，尽管对软件项目管理来说，压缩时间是可以做到的，但同时它是以增加项目的难度和人力费用为代价的，因此时间压缩不能太过分，否则将会导致项目难度及人力费用的极大增加并进而增加项目管理的风险。因此在对K、t_d、D三个变量的一般均衡问题的分析中，可建立如下优化模型，并求解最优技术经济方案。

$$\begin{cases} \min t_d(K,D) \\ K_1 \le K \le K_2 \\ D_1 \le D \le D_2 \end{cases} \qquad \begin{cases} \min D(K,t_d) \\ K_1 \le K \le K_2 \\ t_1 \le t_d \le t_2 \end{cases} \qquad \begin{cases} \min K(t_d,D) \\ t_1 \le t_d \le t_2 \\ D_1 \le D \le D_2 \end{cases}$$

其中，K_1、K_2、D_1、D_2、t_1、t_2分别为软件开发单位根据其现有人力资源总量、技术水平及开发环境条件等来给定的确定值。根据英美使用这些模型的经验，他们认为时间压缩比例一般不能超过原计算值的25%，即有 $\left|\frac{\Delta t_d}{t_d}\right| \leq 25\%$ 。

5.2.4 软件项目开发子周期与生存周期经济要素的关联分析

软件项目各经济要素(人力资源费用、工期、工程难度、生产函数)及其数量关系均是在整个软件项目生存期(又称项目总周期)内获得的。显然，它应该适用于软件项目生存期的各阶段(子周期)如软件设计编码开发阶段、调试与验证阶段、修正维护阶段等。然而实践证明，为了更有效地进行工程项目管理与控制，人们尚需要进一步研究上述生存周期之各阶段，特别是设计编码开发阶段内各经济要素的关联及其与整个生命周期内各经济要素的关联关系。因为这一阶段是编程和分析专业人员和管理人员直接进行软件生产的部分。若我们将软件的设计编码开发阶段称为软件项目的开发子周期，而将软件项目的生存周期称为项目总周期，则以下介绍这二者的工程经济参数之间的关联分析。

若设K_d表示软件开发阶段人力费用总量，$C_d(t)$表示软件开发阶段[0，t)时间段累计人力费用，$m_d(t)$表示软件开发阶段t时刻人力费用，t_{od}表示开发子周期内人力投入的峰值时刻，则与前项目总周期的分析同理，应有

$$\begin{cases} C_d(t) = K_d(1-\mathrm{e}^{-\tilde{b}t^2}), \quad t \geq 0 \\ m_d(t) = 2K_d\tilde{b}t\,\mathrm{e}^{-\tilde{b}t^2}, \quad t \geq 0 \end{cases} \tag{5.18}$$

并有$\tilde{b} = \dfrac{1}{2t_{0d}^2}$，若仍设$t_d$表示软件交付期，则一个实用的假设是：直到$t$=$t_d$时，项目将投入开发阶段人力投入总量$K_d$的95%(其余5%将用于现场安装与有效性测试)，此即为

$$\frac{C_d(t_d)}{K_d} = 1-\mathrm{e}^{-\tilde{b}t_d^2} = 0.95 \text{ 或 } e^{-\tilde{b}t_d^2} = 0.05$$

对上式两边取自然对数，可得$\tilde{b}t_d^2 = -\ln 0.05 \approx 3$或$\tilde{b} = 3/t_d^2$，从而有

$$\tilde{b} = \frac{1}{2t_{0d}^2} = \frac{3}{t_d^2} \quad 或 \quad \sqrt{6}t_{0d} = t_d \tag{5.19}$$

图5.3画出了由(5.10)式与(5.18)式确定的$m(t)$与$m_d(t)$时间曲线。

由图可知，软件项目总周期的$m(t)$与开发子周期的$m_d(t)$在t=0时刻有相同的斜率，或有$\frac{\mathrm{d}m(0)}{\mathrm{d}t} = \frac{K}{t_d^2} = \frac{\mathrm{d}m_d(0)}{\mathrm{d}t} = \frac{K_d}{t_{0d}^2}$，以(5.19)式代入上式，可得

$$K = K_d \frac{t_d^2}{t_{0d}^2} = 6K_d \tag{5.20}$$

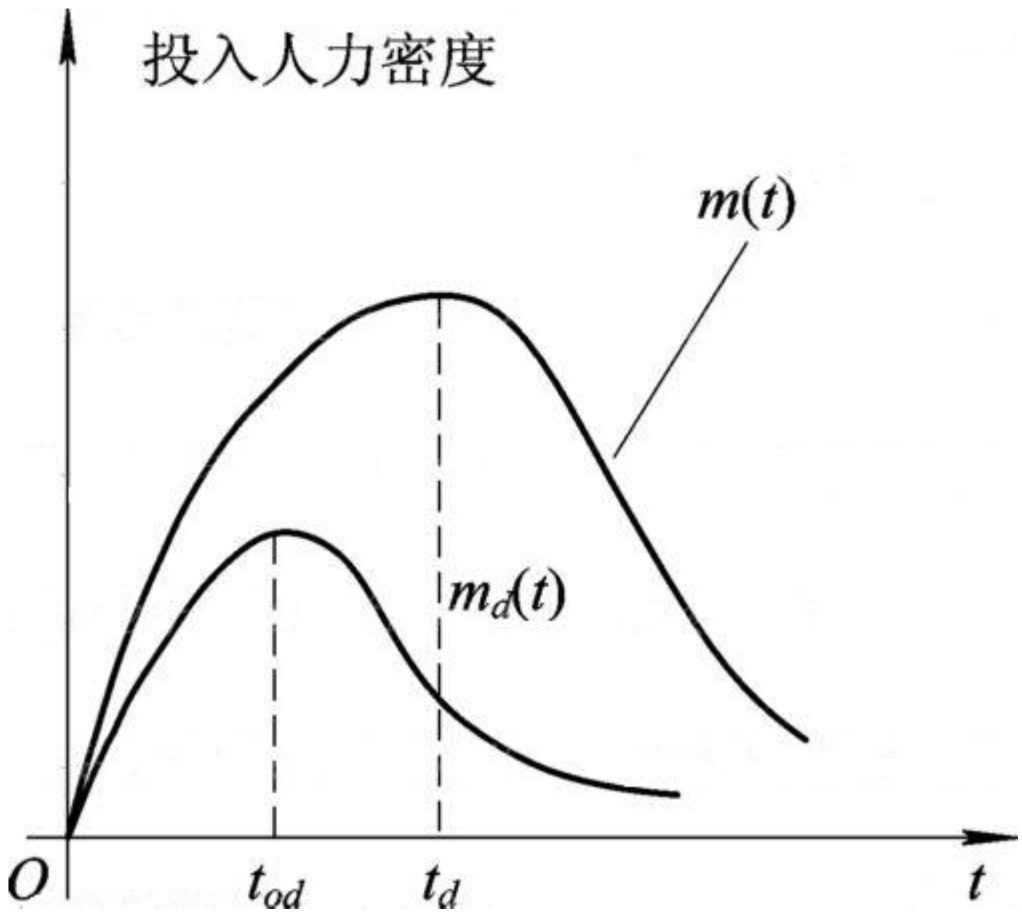

图5.3 $m(t)$与$m_d(t)$时间曲线

再利用(5.18)~(5.20)式有

$$\begin{cases} m_d(t) = \dfrac{K_d}{t_{od}^2} t\,\mathrm{e}^{-\frac{t^2}{2t_{od}^2}} = \dfrac{K}{t_d^2} t\,\mathrm{e}^{-\frac{3t^2}{t_d^2}} \\ C_d(t) = K_d(1-\mathrm{e}^{-\frac{t^2}{2t_{od}^2}}) = \dfrac{K}{6}(1-\mathrm{e}^{-\frac{3t^2}{t_d^2}}) \\ \dfrac{K_d}{t_{od}^2} = \dfrac{K/6}{t_d^2/6} = \dfrac{K}{t_d^2} = D \end{cases} \tag{5.21}$$

上式说明*D*既可以作为整个项目生存周期的难度系数，也可作为开发子周期的难度系数。但对于D_0情况则并非如此，这是由于由(5.13)式有

$$D_0=\frac{K}{t_d^3}=\frac{6K_d}{6\sqrt{6}\cdot t_{0d}^3}=\frac{K_d}{\sqrt{6}\cdot t_{0d}^3}=\frac{\tilde{D}_0}{\sqrt{6}}$$

其中，$\tilde{D}=\dfrac{K_d}{t_{0d}^3}$，为开发子周期的人力增长率。利用(5.21)式容易得到开发子周期峰值人数m_{od}及$C_d(t_{od})$为

$$\begin{cases} m_{od}=m_d(t_{od})=\dfrac{K_d}{t_{od}}\mathrm{e}^{-\frac{1}{2}}=\dfrac{K/6}{\sqrt{\mathrm{e}}t_d/\sqrt{6}}=\dfrac{K}{t_d/\sqrt{6\mathrm{e}}} \\ C_d(t_{od})=K_d\left(1-\dfrac{1}{\sqrt{\mathrm{e}}}\right),C_d(t_d)=K_d(1-\mathrm{e}^{-3}) \end{cases} \tag{5.22}$$

由此可得

$$\frac{C_d(t_{od})}{K_d}=1-\frac{1}{\sqrt{\mathrm{e}}}=39.3\%,\frac{C_d(t_d)}{K_d}=1-\mathrm{e}^{-3}\approx 95.1\%$$

(5.22)式可以作为一个标尺来控制软件设计编码开发阶段的开发进度。若在某个时间上项目已经消耗了其全部人力费用的39%，而此时计划中的任务也已经得到及时正确的完成且无需增派某些人员时，该项目经理即可确认此项目运行轨迹是正确的，而项目结束时的总人力费用可能就是预计的K_d人年，且不会拖延交货时间。而当交货时刻$t=t_d$，且已消耗95%的总开发人力费用时，则尚有5%的K_d还可用于现场安装与有效性测试，从而保证项目开发工作的顺利完成。

[例5.2] 某软件开发项目待开发的程序量S已经测算为9000NCSS，其开发将在环境因子确定为1200的环境中进行，并注意到该软件项目是一个独立的数据处理类型程序，其人力增长率选定为D_0=15。

试求：(1) 开发子周期t_d与开发峰值人数出现时刻t_{od};

(2) 开发子周期投入人力费用K_d，总投放人力费用K与项目难度系数D;

(3) 峰值人数m_{od}、$C_d(t_{od})$与$C_d(t_d)$;

(4) 该软件项目的生产率F_d。

解 (1) 利用(5.15)式所示的软件项目生产函数，可得

$$(\frac{S}{E})^3 = Kt_d^4 = D_0 t_d^7$$

由此可得

$$t_d = \left[\frac{1}{D_0}(\frac{S}{E})^3\right]^{1/7} = \left[\frac{1}{15}(\frac{9000}{1200})^3\right]^{1/7} = 1.16\text{年}，\quad t_{od} = \frac{t_d}{\sqrt{6}} = 0.66\text{年}$$

(2) 由上述生产函数可得

$$K = \frac{\left(\frac{S}{E}\right)^3}{t_d^4} = \frac{\left(\frac{9000}{1200}\right)^3}{1.61^4} = 62.7885\text{人年}$$

$$K_d = \frac{K}{6} = 10.465\text{人年}，D = \frac{K}{t_d^2} = 24.22\text{人/年}$$

(3) 由(5.22)式有

$$m_{od}=\frac{K}{t_d\sqrt{6\mathrm{e}}}=\frac{62.7885}{1.61\times\sqrt{6\mathrm{e}}}=9.66\text{ 人}$$

$$C_d(t_{od})=K_d\left(1-\mathrm{e}^{-\frac{1}{2}}\right)=10.465\times0.393=4.11\text{人年}$$

$$C_d(t_d)=K_d(1-\mathrm{e}^{-\frac{t_d^2}{2t_{od}^2}})=K_d(1-\mathrm{e}^{-3})=10.465\times0.951=9.95\text{人年}$$

(4) 由(5.14)式有

$$F_d=\frac{S}{C(t_d)}=\frac{S}{0.93K}=\frac{9000}{0.93\times62.7885}=367.53\,\mathrm{NCSS}/\text{人年}$$

5.2.5 环境因子的测定

由于环境因子E是由软件工程项目的开发环境、技术状态(如开发方法、开发工具和设备项目管理状况等)所决定的，而这些因素往往取决于软件工程的开发机构，由于不同的开发机构往往有其不同的开发风格与习惯，从而有其不同的通常使用的开发方法、开发工具与设备和管理风格，因此因子E是一个与具有共性的特征量D_0完全不同的反映个性的工程经济参数，它无法像人力增长率D_0那样可以给出一些有广泛性的数值，而应由各开发机构根据反映本部门的项目开发特色的经验数据来确定。例如，某开发机构欲开发一目标软件A，欲确定其环境因子E_A，此时可采用类比法来求解E，亦即该开发机构可从其所拥有的历史资料信息库中选择一个与目标软件A有相似功能，采用相似开发方法与设备的软件B(已运行)，根据该软件B在开发期内所投入的累计人力费用$C_d(t_d)$，交付期t_d和所提交的源代码程序量S，通过下式来计算E：

$$\begin{cases} E = \dfrac{S}{K^{\frac{1}{3}} t_d^{\frac{4}{3}}} \\ K = 6K_d = 6\dfrac{C_d(t_d)}{0.95} = 6.3 \cdot C_d(t_d) \end{cases}$$

如果开发机构所拥有的历史资料信息库中与目标软件A有相似功能，采用相似开发方法与设备的软件有多个，例如有B_1，B_2，…，B_l等软件l个，此时可调用这l个软件的数据序列 $\{(C_{d_j}(t_{d_j}), t_{dj}, S_j),\quad j=1,2,\cdots,l\}$并采用最小二乘法来计算$E$。其中，$C_{d_j}(t_{d_j})$表示第$j$个软件项目在开发子周期[0，$t_{d_j}$)时间段累计投入的人力费用(单位：人年)；$t_{d_j}$表示第$j$个软件项目在开发子周期内人力投入的峰值时刻(单位：年)；S_j表示第j个软件项目在开发子周期内所交付的源代码程序量(单位：NCSS)。注意到有

$$H \overset{\text{def}}{=} \frac{S}{C_d(t_d)} = \frac{EK^{\frac{1}{3}}t_d^{\frac{4}{3}}}{K/6.3} = \frac{6.3\cdot E\cdot t_d^{\frac{4}{3}}}{K^{\frac{2}{3}}} = \frac{6.3\cdot E}{\left(\frac{K}{t_d^2}\right)^{\frac{2}{3}}} = \frac{6.3\cdot E}{D^{\frac{2}{3}}} \qquad (5.23)$$

对(5.23)式两边取常用对数有

$$\lg H = \lg(6.3\cdot E) - \frac{2}{3}\lg D \qquad (5.24)$$

注意到l个软件B_1，B_2，…，B_l有不同的难度系数D_j，不同的源代码程序量S_j和累计投入的人力费用 $C_{d_j}(t_{d_j})$，j=1, 2，…，l，但却有相同的环境因子(此中设该开发机构人员变动不大，故开发人员的开发风格与习惯、开发方法、开发工具和管理常保持前后的一致性)。因而令

$$y_j=\lg H_j,\quad A=\lg(6.3\cdot E),\quad x_j=\lg D_j$$

则由(5.24)式可得

$$y_j = A - \frac{2}{3}x_j \quad j=1, 2, \cdots, l$$

则由最小二乘法拟合有

$$L \overset{\text{def}}{=} \frac{1}{l}\sum_{j=1}^{l}(y_j - A + \frac{2}{3}x_j)^2$$

达最小。

注意到有

$$\frac{\partial L}{\partial A} \overset{\text{def}}{=} \frac{-2}{l}\sum_{j=1}^{l}(y_j - A + \frac{2}{3}x_j) = 0$$

故有

$$\hat{A}=\frac{1}{l}\sum_{j=1}^{l}(y_j+\frac{2}{3}x_j) \quad 或 \quad \lg(6.3\cdot\hat{E})=\frac{1}{l}\sum_{j=1}^{l}(y_j+\frac{2}{3}x_j)$$

由此可得

$$\hat{E}=\frac{1}{6.3}10^{G},\quad G=\frac{1}{l}\sum_{j=1}^{l}(\lg H_j+\frac{2}{3}\lg D_j)$$

5.3 不同规模软件的生产过程经济分析

人们在对不同规模的软件工程经济分析时发现这样一个事实：随着规模的不同，软件工程项目的人力资源组织及其管理有较大的区别，对于一些小型软件工程项目，由于功能需求简单，项目难度低，因而少数几个软件工程师即可完成其规划、分析、设计、编码、测试等全部任务而无需其他的支持人员。然而在大、中型软件工程项目建设中，情况就会有所不同。由于这些软件工程往往是具有不同的应用背景(如交通工程、水电工程、宇航工程、军事作战工程等)的嵌入式软件，因而在软件规划，分析与设计中不仅需要大量的应用工程专业知识和系统硬件(计算机网络与通信设备)的理论方法与操作经验知识，而且由于投入了大量人力资源而使工程的计划与组织的协调显得十分重要。因此为了使这样的大、中型软件工程能快速、高效且高质量地完成建设，开发机构将投入的人力资源分成项目开发任务组和项目支持任务组是必要的，其中项目开发任务组负责软件工程开发所必须完成的基本任务如规划、分析、设计、编码及其审查与测试等，而项目支持任务组则完成如下的支持任务：

(1) 应用学科领域知识的支持。

(2) 计算机网络与通信设备的使用与维护支持。

(3) 工程计划网络(PERT)的设计、跟踪与控制。

(4) 文本提供、质量保证与配置管理。

(5) 资源控制、任务跟踪协调与进程监控。

显然，上述的项目支持任务组的工作是十分重要的，而且软件项目的规模越大，所需要的支持任务量也越大。下面介绍有关上述内容的定量分析内容。

5.3.1 不同规模软件的人力投入属性及其比较

为研究涉及项目开发任务组及项目支持任务组的有关工程经济分析，我们首先给出了有关工程经济参数的变量表5.2。若设Δ_1表示开发阶段时间区间(子周期)，p表示开发与支持，d表示开发，S表示支持，则显然有

$$\frac{\mathrm{d}C_i(t)}{\mathrm{d}t} = m_i(t), \quad C_i(t) = \int_0^t m_i(s)\,\mathrm{d}s, \quad i \in I = \{p, d, s\}$$

$$m_p(t) = m_d(t) + m_s(t), t \in \Delta_1$$

$$C_p(t) = C_d(t) + C_s(t), t \in \Delta_1$$

表 5.2 有关变量经济内涵表

变量	经 济 内 涵	单位
$m(t)$	在项目生存周期内 t 时刻的人力投入量	人
$m_p(t)$	在项目开发阶段 t 时刻的人力投入总量，它包括开发人力投入量与支持人力投入量两部分	人
$m_d(t)$	在项目开发阶段 t 时刻的开发人力投入量	人
$m_s(t)$	在项目开发阶段 t 时刻的支持人力投入量	人
$C(t)$	在项目生存周期内$[0，t)$区间内累计投入人力总量	人年
$C_p(t)$	在项目开发阶段$[0，t)$时间段内累计投入开发人力与支持人力总量	人年
$C_d(t)$	在项目开发阶段$[0，t)$时间段内累计开发人力投入总量	人年
$C_s(t)$	在项目开发阶段$[0，t)$时间段内累计支持人力投入总量	人年
K	在项目生存周期内为完成所有任务投入的总工作量(人力量)	人年
K_p	在项目开发阶段内为完成所有(包括开发任务与支持任务)任务投入的人员总量(人力量)	人年
K_d	在项目开发阶段内为完成开发任务投入的开发人员总量(人力量)	人年
K_s	在项目开发阶段内为完成支持任务投入的支持人员总量(人力量)	人年
t_d	在项目生存周期内投入人力峰值的时刻或项目交付时间或工期	年(或月)
t_{op}	在项目开发阶段内投入人力峰值的时刻	年(或月)
t_{od}	在项目开发阶段内开发人力峰值的时刻	年(或月)
t_{os}	在项目开发阶段内支持人力峰值的时刻	年(或月)

国外很多软件工程学者在经过对以往已完成的软件工程项目的各工程经济变量数据进行研究后得到了一些有益的结论，这些结论列于表5.3。由表得知任何一个软件项目开发子周期内的开发人力投入量m_d(t)，项目任务人力投入总量$m_p(t)$及总周期(生存周期)内的人力投入量$m(t)$三者的分离与重合程度与软件规模(程序量)S有很大的关联。我们将软件规模(非注释性源代码程序量)$S \leq 18$ kNCSS的软件称为小型软件，将$S \in (18$ kNCSS，70 kNCSS)的软件称为中型软件，而将$S \geq 70$ kNCSS的软件称为大型软件。则由表5.3得知在小型软件中有$m_d(t)=m_p(t)$，这是由于投入人力少，因此即使有一些支持任务，通常也由开发人员兼顾；而在大型软件中，由于所投入的支持任务人力量远远大于开发任务人力量，从而使$m_p(t)$与$m(t)$非常接近或基本重合；至于中型软件则呈现出$m_d(t)$、$mp(t)$、$m(t)$三者分离的现象，而且随着S的增大，$m_p(t)$与$m_d(t)$分离度越大，而$m_p(t)$与$m(t)$重合度越大。上述这种人力投入的规律性可详见图5.4(a)、(b)、(c)、(d)。其中图5.4(a)为小型软件项目，图(b)与(c)为中型软件项目，图(d)为大型软件项目。下面我们分别对大、中、小型软件工程分别作有关的工程经济分析。

表 5.3 规模属性表

程度量 S	$S \leqslant 18$kNCSS	$S \in (18$kNCSS,70kNCSS$)$	$S \geqslant 70$kNCSS
$m_d(t)$与 $m_p(t)$的关系	$m_d(t)$与 $m_p(t)$重合	$m_d(t)$与 $m_p(t)$分离，S 愈大，两者的分离度愈大	$m_d(t)$与 $m_p(t)$分离
$m(t)$与 $m_p(t)$的关系	$m(t)$与 $m_p(t)$分离	$m(t)$与 $m_p(t)$分离，S 愈大，两者的重合度愈大	$m(t)$与 $m_p(t)$基本重合

注：1kNCSS＝1000NCSS。

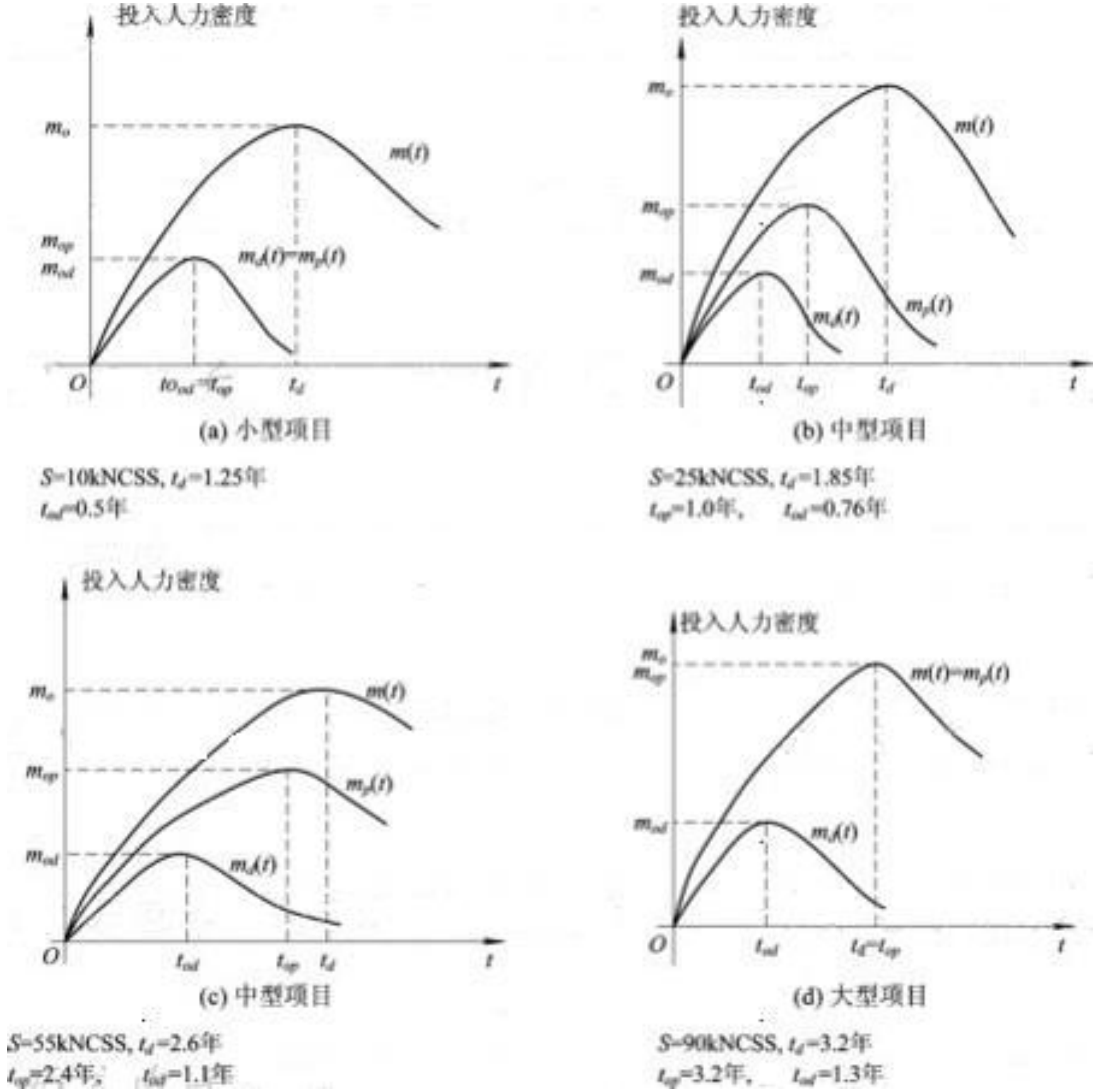

图5.4 不同规模软件开发属性图

5.3.2 不同规模软件的生产过程经济分析

以下介绍小型、中型、大型软件生产过程的工程经济分析。

1. 小型软件工程经济分析

由表5.3得知：在小型软件工程项目中有$m_d(t)=m_p(t)$，从而也有$C_d(t)=C_p(t)$，$K_d=K_p$，$m_{od}=m_{op}$，$t_{od}=t_{op}$，而且有关$m_d(t)$和$m_p(t)$的相关工程经济参数$C_d(t)$、$C(t)$、K_d、K、m_{od}、m_0、t_{od}、t_d、D、D_0、$C_d(t_{od})$、$C_d(t_d)$之间的数量关系(5.10)～(5.15)式和(5.18)～(5.22)式对于小型软件工程项目仍然适用，于是人们也可利用上述各工程经济参数间的数量关系式来进行小型软件工程的工程经济分析与设计。

2. 大型软件工程经济分析

由表5.3得知：在大型软件工程项目中有$m(t)=m_p(t)$，从而也有$C(t)=C_p(t)$，$K=K_p$，$m_0=m_{op}$，$t_d=t_{0p}$，而且各工程经济参数间的数量关系(5.10)～(5.15)式和(5.18)～(5.22)式对于大型软件工程项目仍然适用，于是人们也可利用上述各工程经济参数间的数量关系式来进行大型软件工程的工程经济分析与设计。

3. 中型软件工程经济分析

由表5.3得知：在中型软件工程项目中，由于$m_d(t)$、$m_p(t)$、$m(t)$三者分离，虽然有(5.10)～(5.15)式和(5.18)～(5.22)式对中型软件工程仍然适用，但$m_p(t)$仍需求解，$m_p(t)$、$C_p(t)$、K_p、t_{op}、m_{op}彼此之间的关联及其与其他工程经济参数之间的关联仍待研究。为此，以下首先讨论$m_p(t)$的求解。考虑到$m_p(t)$仍可用诺顿/瑞利函数来描述，即与前同理推导有

$$m_p(t) = 2K_p ct\,\mathrm{e}^{-ct^2} \tag{5.26}$$

注意到项目峰值人数在t_{op}时刻出现，故在(5.26)式中两边对t求导数并令其为零，即可解得$C = \dfrac{1}{2t_{op}^2}$，再将其代入(5.26)式有

$$m_p(t)=\frac{K_p}{t_{op}^2}t\,\mathrm{e}^{-\frac{t^2}{2t_{op}^2}},\quad C_p(t)=\int_0^t m_p(s)\,\mathrm{d}s=K_p(1-\mathrm{e}^{-\frac{t^2}{2t_{op}^2}})\quad t\ge 0 \tag{5.27}$$

为进一步研究开发投入人力，支持投入人力和项目总人力投入间的彼此关联关系，可设

$$K_p=\frac{K}{a^2}\qquad a>0 \tag{5.28}$$

对于$m(t)$与$m_p(t)$在一般情况下仍应有

$$\frac{\mathrm{d}m(0)}{\mathrm{d}t}=\frac{\mathrm{d}m_p(0)}{\mathrm{d}t}\text{或有}D=\frac{K}{t_d^2}=\frac{K_p}{t_{op}^2} \tag{5.29}$$

将(5.28)式代入(5.29)式和(5.27)式，可得

$$t_{op}=\frac{t_d}{a},m_p(t_{op})=m_{op}=\frac{K_p}{t_{op}}\mathrm{e}^{-\frac{1}{2}} \tag{5.30}$$

利用(5.28)、(5.11)式及上述两式容易得到

$$m_{op}=\frac{K_p}{t_{op}}\mathrm{e}^{-\frac{1}{2}}=\frac{K/a^2}{t_d/a}\mathrm{e}^{-\frac{1}{2}}=\frac{K}{a\cdot t_d}\mathrm{e}^{-\frac{1}{2}}=\frac{m_0}{a} \tag{5.31}$$

再利用(5.31)、(5.22)和(5.11)式，有

$$\frac{m_{op}}{m_{od}}=\frac{m_o/a}{K/(t_d\sqrt{6\mathrm{e}})}=\frac{\sqrt{6}}{a} \tag{5.32}$$

将(5.30)式代入(5.27)式，有

$$C_p(t_d)=\int_0^{t_d}m_p(s)\,\mathrm{d}s=K_p[1-\mathrm{e}^{-\frac{t_d^2}{2t_{op}^2}}]=K_p(1-\mathrm{e}^{-\frac{a^2}{2}})$$

$$Y(a)\overset{\text{def}}{=}\frac{C_p(t_d)}{K}=\frac{C_p(t_d)}{a^2K_p}=\frac{1}{a^2}(1-\mathrm{e}^{-\frac{a^2}{2}}) \tag{5.33}$$

注意到一个中型软件项目在项目子周期内各经济量间的相关关系(5.28)～(5.33)式均与参数a有关，我们称a为规模参数。以下来讨论参量a的确定。

5.3.3 规模参数a的确定

普特纳姆在对以往的信息系统数据资料的研究中发现，软件项目的程序量S与参量a值有极强的负相关关系，并根据普特纳姆数据库中的数据计算得到S与a的样本相关系数达-0.998。上述经验结论说明可以建立S与a的经验公式。为此，我们首先将普特纳姆数据中的一组样本序列$\{(S_l，a_l)，l=1，2，\ldots，n\}$在$SOa$平面上标点并连接成曲线，此中$S_l$为第$l$个软件项目的程序量，$a_l$为由该项目的$t_d$与$t_{op}$相除所得到的比值。我们发现此关联曲线具有分段负指数曲线形状特征(详见图5.5)，为此可采用函数$a(S)=a_0+be^{-eS}$来作曲线拟合。运用典型的非线性回归拟合(或其他非线性曲线拟合方法)容易求得$a_0=1, b=6.23, e=0.079$，从而获得了拟合曲线

$$a(S)=1+6.23e^{-0.079S} \tag{5.34}$$

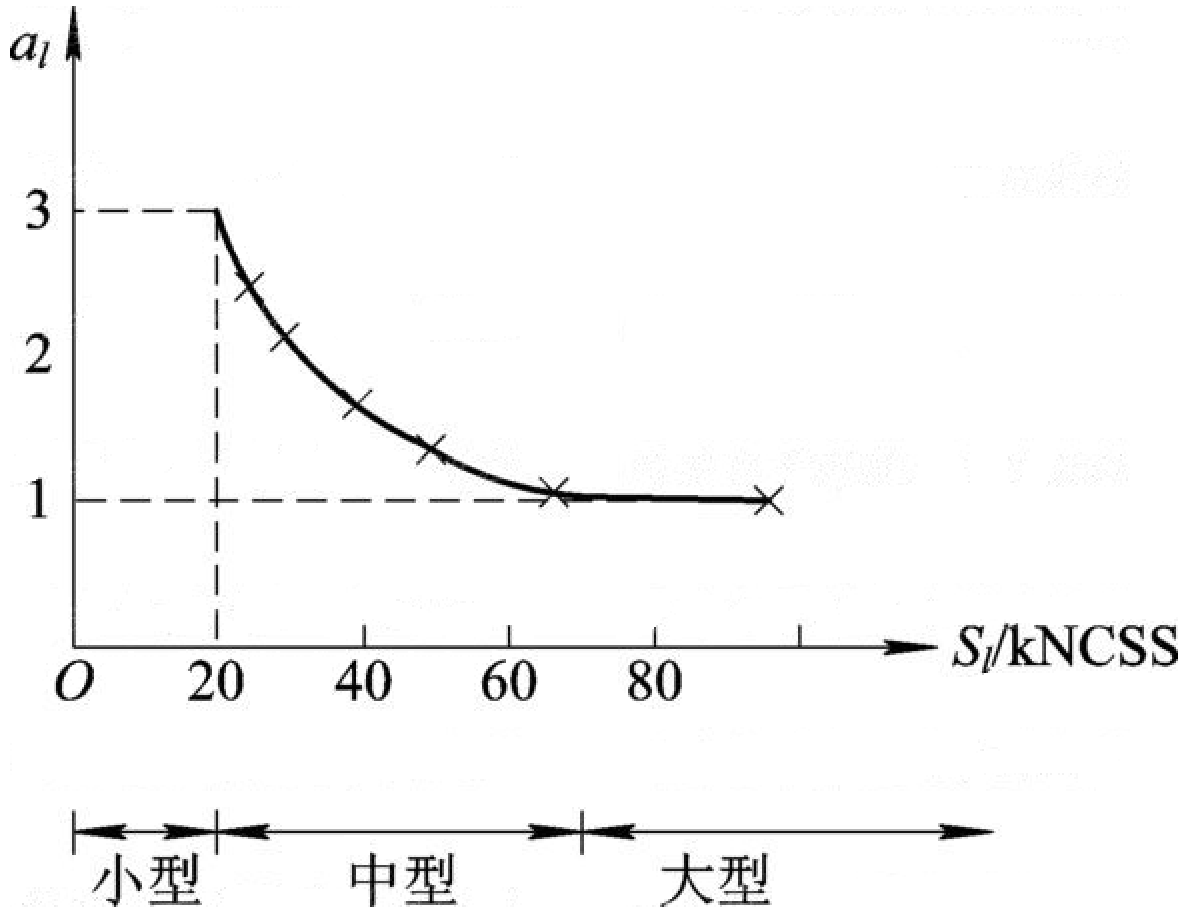

图5.5 *a*-*S*曲线图

普特纳姆还对此拟合曲线的有效性问题做了研究，并列出了表5.4所示的对比，表中S_n列及a_n列(第二列与第三列)分别为不同软件规模的程序量及运用该软件项目实际数据t_d与t_{op}相除算得的真实a_n值，而该表之第四列显示出了当$S=S_t$时代入拟合算法(5.34)式所算得的对应拟合值 $\hat{a}_n = a(S_n)$ ，容易计算该拟合的均方误差有

$$\varepsilon = \frac{1}{7}\sqrt{\sum_{n=1}^{7}(a_n - a(S_n))^2} < 1.2\%$$

表 5.4 拟 合 误 差 表

n	S_n	a_n	$a(S_n)$
1	5～15	2.44	2.51
2	20	—	2.29
3	25	1.85	1.87
4	30	1.61	1.59
5	40	1.30	1.27
6	50	1.15	1.12
7	70	1.04	1.02
8	100	1.00	1.002

注意到在表5.4中，除S_8=100 kNCSS>70 kNCSS为大型软件项目外，其他均为中型规模软件，因此，人们可根据(5.34)式由中型软件规模S来确定其对应的规模参数a。

对于小型软件，由于有$t_{od}=t_{op}$，则利用(5.30)式和(5.19)式的结果可得$a=t_d/t_{op}=t_d/t_{od}=\sqrt{6}$。对于大型软件，由于有$t_{op}=t_d$，因而有$a=t_d/t_{op}=1$。

综合上述三种不同规模的结果，可得规模参数a的基本算法如下：

$$a(s)=\begin{cases}\sqrt{6} & S\le 18\,\mathrm{kNCSS}\\ 1+6.23\mathrm{e}^{-0.079\,s} & 18\,\mathrm{kNCSS}<S<70\,\mathrm{kNCSS}\\ 1 & S\ge 70\,\mathrm{kNCSS}\end{cases}$$

[例5.3] 欲开发一程序量S=45 000 NCSS的中型嵌入式软件项目，根据该软件的开发属性知人力增长率可取推荐值D_0=8，环境因子经考察定为E=2400，试计算:

(1) 该软件项目工期t_d，生存周期内人力总费用K，难度系数D;

(2) 项目开发子周期内峰值人数m_{od}及其出现时间t_{od};

(3) 项目开发子周期内人力总费用K_P、峰值人数m_{oP}及其出现时刻t_{oP}，t_d时刻的投入累计人力费用$C_P(t_d)$。

解　注意到S=45 000 NCSS，故为中型软件项目，因此对项目完成的研究应深入到开发子周期、项目子周期及总周期(生存周期)及其关联中去。

(1) 由(5.15)式与(5.13)式可得项目生存周期内各参量有

$$\left(\frac{S}{E}\right)^3 = Kt_d^4 = \frac{K}{t_d^3} \cdot t_d^7 = D_0 t_d^7$$

$$t_d = \sqrt[7]{\frac{\left(\frac{S}{E}\right)^3}{D_0}} = \sqrt[7]{\frac{\left(\frac{45\,000}{2400}\right)^3}{8}} = \sqrt[7]{\frac{(18.75)^3}{8}} = 2.6\text{年}$$

故有 $K = D_0 t_d^3 = 8 \times (2.6)^3 = 141\text{人年}$，$D = \frac{K}{t_d^2} = \frac{141}{2.6^2} = 21\text{人}/\text{年}$

(2) 在开发子周期内有

$$t_{od} = \frac{t_d}{\sqrt{6}} = \frac{2.6}{\sqrt{6}} = 1.1\text{年}$$

$$m_{od}=\frac{K}{t_d\sqrt{6\mathrm{e}}}=\frac{141}{10.5}\approx 13\text{人}$$

(3) 在项目子周期内，由(5.35)式可得

$$a(45)=1+6.23\exp\{-0.079\times 45\text{ k}\}=1.18$$

从而由(5.28)～(5.33)式可得

$$K_P=\frac{K}{a^2}=101\text{人年},t_{op}=\frac{t_d}{a}=2.2\text{年}$$

$$m_{op}=\frac{K_P}{t_{oP}\sqrt{\mathrm{e}}}=\frac{m_o}{a}=\frac{K}{at_d}\mathrm{e}^{-\frac{1}{2}}=28\text{人}$$

$$C_P(t_d)=K_P(1-\mathrm{e}^{-\frac{a^2}{2}})=50.8\ \text{人年}$$

注意到在$t=t_d$时已消耗了开发人力费用K_d的95%，从而还剩K_S用于管理支持、质量检验，现场测试等，利用(5.33)式有

$$K_s = C_P(t_d) - K_d \cdot 95\% = K_P(1 - \mathrm{e}^{-\frac{a^2}{2}}) - K_d \cdot 0.95$$

$$= 101 \cdot (1 - \mathrm{e}^{-\frac{(1.18)^2}{2}}) - 22.2 = 28.6\text{人年}$$

5.4 软件项目理论生存周期长度及其关联分析

对于大型软件项目，人们除关心开发子周期与项目任务子周期内的工程经济分析外，更应当关心在软件交付用户使用后的经济活动及其经济分析。为此，我们设该软件项目的理论生存周期长度为 t_f，亦即当 $t \geq t_f$ 时该软件将“报废”，但在实际使用时软件的报废将视多种情况而定，而并非为 t_f，以下来寻求 t_f 与 t_d、K 等主要经济量的关系。注意到 K 为[0，t_f]期间投入的累计人力资源总量，而 $C(t_f)$ 为[0，t_f)内投入的累计人力资源总量，故可认为有 $C(t_f)=K-1$。

此外，通过大量观察得知在大型软件项目中，项目任务子周期与总周期(生存周期)基本接近，亦即有$C(t)\approx C_P(t)$，从而有

$$K-1=C(t_f)\approx C_P(t_f) \qquad (5.36)$$

由(5.36)式与(5.10)式还有

$$C(t_f)\approx C_P(t_f)=K_P(1-\mathrm{e}^{-\frac{t_f^2}{2t_d^2}})=K(1-\mathrm{e}^{-\frac{t_f^2}{2t_d^2}}) \qquad (5.37)$$

综合(5.36)式与(5.37)式有

$$K-1=K-K\,\mathrm{e}^{-\frac{t_f^2}{2t_d^2}} \text{ 或者 } K=\exp\left(\frac{t_f^2}{2t_d^2}\right)$$

两边取对数及移项得

$$t_f = t_d \sqrt{2 \ln K}$$

(5.38)式给出了该软件项目生命周期的“报废”时刻t_f与交付工期t_d、投入人力费用总量K之间的数量关系。据此关系可进一步研究在时间区间(t_d，t_f)间的经济活动及其经济分析。

[例5.4] 某欧洲国家的国际长途电话中心已经开发一通信控制软件，该软件用高级语言编写，程序量S=245 kNCSS, 开发工作投入的人力总费用K_d=196人年，自开发到交付的时间间隔t_d=3.66年，为研究该软件交付用户后的有关经济活动。

(1) 试求该软件项目的K_P、$C(t_d)$、t_{oP}、t_f、D_0、D;

(2) 试求该软件项目的E、t_{od}、m_{od}、m_{oP};

(3) 对上述各经济变量作经济分析。

解 (1) 注意到S=245 kNCSS为大型软件，此时有$a(245)=1$，于是由(5.20)、(5.11)、(5.38)式得到

$$K_P=K=6K_d=6\times196=1176\text{人年}$$

$$C(t_d)=C_P(t_d)=0.39K=0.39\times1176=459\text{人年}$$

$$t_{oP}=t_d=3.66\text{年}$$

$$D=\frac{K}{t_d^2}=\frac{1176}{(3.66)^2}=88\text{人/年}$$

$$D_0=\frac{K}{t_d^3}=\frac{1176}{(3.66)^3}=24\text{人/年}^2$$

(2) 利用(5.15)、(5.19)、(5.22)、(5.32)式得

$$E = \frac{S}{(Kt_d^2)} = \frac{245}{1176^{\frac{1}{3}} \times (3.66)^{\frac{4}{3}}} = 4115$$

$$t_{od} = \frac{t_d}{\sqrt{6}} = 1.5\text{年}$$

$$m_{od} = \frac{K_d}{t_{od}\sqrt{\mathrm{e}}} = 79\text{人}$$

$$\frac{m_{oP}}{m_{od}} = \frac{\sqrt{6}}{a} = \sqrt{6}\text{或}m_{oP} = \sqrt{6}m_{od} = 193\text{人}$$

(3) 由$t_f - t_d$=13.8－3.66=10.14年可知，该软件交付使用后理论上尚需运行10多年，因而必须投入相当一批人力费用来作软件维护、有效性测试、可靠性增长试验等其他任务，而这部分投入的总人力费用可计算有

$$K - C(t_d) = 1176 - 459 = 717 \text{ 人年}$$

为保证上述任务完成所需要的技术环境因子，由上计算为E=4115，这是一个较高的环境因子值，因此必须创造条件来满足此环境要求。

在整个生存周期[0，t_f)中开发人员高峰、项目任务人力高峰分别出现在t_{od}=1.5年，t_{oP}=3.66年，并显然有$0<t_{od}<t_{oP}=t_d<t_f$。而相应的峰值人数依次为m_{od}=79人，$m_o=m_{oP}$=193人，由此可知：对大型软件项目人力费用的峰值不在开发阶段[0，t_{od})，而在交付软件时刻$t=t_d=t_{oP}$。

由该软件项目难度系数D=88人/年=1.83人/周，说明该软件开发组人数初期基本上是按照平均每周1.83人的速率在增加，这样的高速率正好解释了前面计算出的人力增长率为24人/年^{2}这一事实。而事实上由于该软件项目是一个大型项目，组建这样一个大型项目任务组也确实需要这样的速度，因为这样的大项目组除开发人力外，还需要资源控制、计划支持、任务定义、任务跟踪和进程控制等人力投入。

作业

习题6、9、10

习 题 五

1. 什么是生产函数? 它反映了经济学中的什么问题? 什么是规模报酬递增、规模报酬递减、规模固定? 如何进行判定?

2. 如何确定软件的生产函数? 软件生产函数反映了软件生产过程中的何种工程经济特性?

3. 什么是软件生产率? 影响软件生产率的主要影响因素有哪些? 为提高软件生产率，常用的应对措施有哪些?

4. 普特纳姆(Putnam)所建立的数量关系式(5.10)~(5.15)式反映了软件生产过程中的哪些工程经济参数间的关联? 能解决软件分析与设计中的哪些问题?

5. 对于一个给定工期t_d、生存期投入总费用K的软件项目，若用户要求工期提前5%，试问此时相应的生存期投入总费用K、项目难度系数D和开发阶段峰值人数m_{od}将会有什么样的变化。

6. 某软件工程是一个与其他系统有多个接口和交互功能的全新软件，系统规模S=7500 NCSS，根据该软件开发机构过去项目开发的经验，环境因子E可取1500。试利用Putnam等建立的生产过程分析理论，求解该软件的开发工期t_d、生存期投入总费用K、开发阶段的峰值人数m_{od}。

7. 某嵌入型软件，其程序测算值S=18 000 NCSS，根据该软件开发机构过去项目开发的经验，环境因子E可取1200，完成该项目的人力增长率取D_0=27。试利用Putnam等建立的生产过程分析理论，求解该软件的开发工期t_d、生存期投入总费用K、生存期峰值人数m_o和项目难度D。

8. 由Putnam所建立的生产函数推导软件规模(源代码程序量)S、人力增长率D_0、工期t_d与环境因子E的数量关系式，并计算当S=4500 NCSS，t_d=2年，D_0=7.5人/年2时对应的环境因子E、软件开发难度D、项目周期内累计投入的人数资源量K、人力投入的最大值m_o。

9. 某实时处理软件属小型软件，其开发环境因子估算为E=2200，经参照同类软件的统计资料，该软件的人力增长率定为D_0=8人/年2，软件程序量测算值S=5500 NCSS。

(1) 利用Putnam模型计算该软件开发时间t_d、项目总周期人力总费用K、开发子周期人力费用K_d、项目难度系数D、开发阶段峰值人力数$m_{od}=m_d(t_{od})$。

(2) 用户对(1)中计算之开发时间t_d不满意，希望在保持原有S、E条件下，在此t_d基础上压缩工期两个月，试问相应的D_0及K将会有何种变化。

10. 某编译程序属中型软件，开发工作始于1988年3月，1991年9月交付使用，截止1991年9月共耗费人力费用$C_d(t_d)$=14.8人年，开发程序工作量S=47000 NCSS。

(1) 利用Putnam模型计算规模参数a、项目总周期人力总费用K、开发子周期的人力总费用K_P、开发环境因子E、项目难度系数D和人力增长率D_0以及开发峰值时间t_{od}和人数$m_{od}=m_d(t_{od})$、项目峰值时间t_{oP}和人数$moP=m_P(t_{oP})$。

(2) 根据(1)中计算的D_0值，你认为在保持原有的人力费用投入水平下，此软件能否在更短的时间内开发出来? 理由何在?

11. 某大型商业系统软件有规模S=150 000 NCSS，交付期(工期)t_d=3.2年，截止交付期时为止已累计投入人力资源数$C(t_d)$=400人年，软件单位平均成本K_c=6万元/人年。

试求该软件:

(1) 环境因子E、理论生存周期长度t_f、总成本U;

(2) 开发周期内最大峰值人数m_{od}、项目生存周期内最大峰值人数m_o。

12. 软件开发机构如何来确定自身的软件环境因子E? 我国软件行业欲建立类似于(5.35)式的规模参数$a(s)$的具体函数表达式，以适应我国国情，你认为需采取哪些措施? 应采集哪些数据序列? 解决上述问题的步骤有哪些?