IT 工程时间风险的度量方法①
——基于正态假设和蒙特卡罗模拟的度量

王志刚[1,4], 赵俊岚[2], 张丽君[3]

[1](内蒙古财经大学 统计与数学学院, 呼和浩特 010070)
[2](内蒙古财经大学 网络信息中心, 呼和浩特 010070)
[3](内蒙古财经大学 计算机信息管理学院, 呼和浩特 010070)
[4](中国人民大学 统计学院, 北京 100872)

摘 要: IT 工程时间风险是 IT 工程中的最主要的风险之一, 现有的 IT 工程时间风险的度量方法大多是直接基于专家打分的度量结果, 导致结果的精度不够, 给实际应用带来了不便. 为了改善该问题, 文中提出两个不同的分析路径来解决估计结果的离散问题, 一种是特定分布假设条件下 IT 工程时间风险度量方法, 主要步骤是首先设定每一步 IT 时间服从的分布, 然后在理论上推导出特定概率保证下 IT 工程时间风险的计算公式, 最后利用专家打分值计算得到 IT 工程时间风险的度量值; 另一种是利用蒙特卡洛模拟度量 IT 工程时间风险方法, 在设定好分布形态后计算相应的 IT 工程时间风险度量值. 最后文中利用上述两种方法度量了一个简单小型项目在不同概率情况下的时间风险值, 结果表明利用蒙特卡洛模拟度量 IT 工程时间风险方法在满足度量精度的条件下, 具有更广的适用面.

关键词: IT 工程时间风险; 正态分布; 蒙特卡洛模拟

Measuring Risk of the IT Project's Time

WANG Zhi-Gang[1,4], ZHAO Jun-Lan[2], ZHANG Li-Jun[3]

[1](Statistics and Mathematics College, Inner Mongolia University of Finance and Economics, Hohhot 010070, China)
[2](Department of Network Information Center, Inner Mongolia University of Finance and Economics, Hohhot 010070, China)
[3](Computer Information Management College, Inner Mongolia University of Finance and Economics, Hohhot 010070, China)
[4](Statistics College, Renmin University of China, Beijing 100872, China)

Abstract: Risk of the IT Project’s Time is one of the most important risks in IT projects. Most of the existing IT engineering time risk metrics is directly based on Experts Grading Method metric results, so the accuracy of the results is not enough to bring the practical application of the inconveniencing. In order to solve this problem, two different analytical paths are proposed in this paper. The first risk measurement method is to measure the risk in the specific distribution assumptions on IT Project’s time; the second one is using the Monte Carlo simulation to measure IT project time risk. And finally, the risk of a simple small-scale IT projects Time is measured using the two methods. The result shows that the measure IT project time risk method using Monte Carlo simulation can meet the precision, and it is applicable to a wider area.

Key words: risk of the IT Project’s Time; normal distribution; Monte Carlo simulation

1 研究背景

由于 IT 项目的独特性, 几乎不存在完全相同的两个项目, IT 项目风险的可预测性也就差得多, 很难用“先例”来评价项目进度计划是否合理. 同时, 传统网

① 基金项目:内蒙古自然科学基金(2010MS1011)
收稿时间:2013-02-21;收到修改稿时间:2013-03-20

络计划技术都存在一定缺陷, 关键路径法仅适用于各活动历时确定的情况; 计划评审技术(PERT)假定在网络中只有一条占支配地位的路线, 同时还要忽略其它路线重要性提升并变为关键路线的可能性, 忽略了非关键路径上的活动对完工时间的影响. 这两种方法, 在实践中都未能全面有效地考虑项目中的风险. 因此, 在 IT 项目管理, 尤其是技术复杂、活动繁多且各部分联系紧密的项目中, 研究一种能克服传统网络计划技术缺陷并且简单可行的定量风险分析方法, 使其能够科学估计项目完成所需时间, 并找出影响项目进度的关键活动, 将对支持项目管理者做出高质量决策具有重要意义[1].

1.1 工程风险

IT 项目具有需求多变的特点, 因此在 IBM 的 RUP 和其他的敏捷方法论中[2], 一直将需求的不确定性列为软件项目的最主要特点, 由于该特点, 进而导致 IT 项目与传统普通项目间存在着很大的差异, 所以 IT 项目的风险管理的难度要高于传统项目[3]. 其风险主要有:

(1) 项目规模估计不准确

相对于普通工程项目中常常出现的后期资金短缺的情况, 在 IT 项目开始后, 很少会发生该类情况, 即便是出现资金短缺情况, 钱也是因为签合同的时候要少了. 所以这类风险的控制通常在放在软件开发的前期.

(2) 人的因素对项目影响

在整个软件项目中, 人可以说是灵魂, 软件项目中不使用钢筋、沙石和水泥, 也不需要施工机械. 软件项目的原材料就是人的思想和智慧, 而计算机和 CASE 软件便是项目中的施工工具. 通过使用键盘和鼠标, 程序员编写出了无数的程序代码. 因此可以看出软件项目最大成本就是人力成本.

(3) 时间风险

每一个工程从着手操作到完成都需要花费一定的时间, 而在 IT 工程中, 能否按照预期时间完成时非常重要的. 但是 IT 项目活动中存在很多不确定的因素, 则使得 IT 工程无法按照预期完成. 本文将主要就 IT 工程时间风险作为研究内容, 进行研究.

1.2 IT 工程时间风险

一个项目能否在预定时间内按工程质量要求完成, 即是项目中重要的问题之一, 也是项目管理的目标之一. [4]但在项目进行过程中, 风险是不可避免的, 正是这些风险因素的存在, 使得 IT 项目各活动历时具有不确定性, 从而影响了项目工期与目标的实现, 加大了项目管理的难度. IT 工程时间风险是对 IT 工程过程中的时间变化存在的风险进行评估, 进而缩小 IT 工程的风险指数. 因此, 在进行 IT 项目进度计划编制时, 应充分考虑被风险因素影响的活动计划完成时间的合理性, 以及各活动历时的不确定性对项目工期带来的影响. 那么, 在 IT 项目管理中对进度风险进行定量分析、合理评估, 对于辅助项目管理者做出高质量的决策显得非常重要. IT 工程时间风险评估方法有多种, 现在较为常用的是运用专家打分的 IT 工程时间风险度量方法.

1.3 基于专家打分 IT 工程时间风险度量的缺陷

专家打分也称为头脑风暴法, 是指按照规定的原则选定特定数量的专家, 按照一定的方式组织专家进行会议讨论, 发挥专家集体的智能结构效应, 对预测对象未来的发展趋势及状况, 依据个人知识和经验做出判断的一种方法[5]. 该方法主要存在以下几点缺陷: (1)由于参加会议的人数有限, 因此代表性不充分; (2)受权威的影响较大, 容易压制不同意见的发表; (3)易受表达能力的影响, 而使一些有价值的意见未得到重视; (4)由于自尊心等因素的影响, 使会议出现僵局; (5)易受潮流思想的影响, (6)给出结果是一个离散型变量, 不利于进一步的风险估计.

本文将针对第 6 点采用特定分布条件下的度量方法和蒙特卡罗模拟方法对 IT 工程的时间风险进行度量, 以期改进以往的时间风险度量结果.

2 基于特定分布假设条件下IT工程时间风险度量

首先在专家打分的基础上, 我们可以获得最有可能的工程时间历时 m 和最小值 a、最大值 b, 一个重要的问题就是如何将这一组离散变量, 变成一个连续分布. 首先根据已有的研究文献和工程时间数据特点, 常将工程的时间历时设定为 β 分布, 正态分布, 三角分布, 梯形分布等分布形态. 本文以正态分布为例对 IT 工程时间进行风险度量, 在理论上其他分布也可以按照这个思路进行分析.

设软件开发工程的总时间为 T, 该工程一共包含 k 步相互独立的步骤, 流程中的每一细节工作所用的时间为 $t_i(i=1,2,\ldots, k)$, 假设第 i 步时间历时为 t_i 服从正态分布的随机变量, 即 $t_i \sim N(u_i,\sigma_i^2),(i=1,2\ldots, k)$,

其密度函数 $p(t_i)$ 为表达式为：$p(t_i)=(\sqrt{2\pi}\sigma)^{-1}\exp\left\{-(t_i-u)^2/2\sigma^2\right\},-\infty<x<\infty$，该函数形态为一条钟形曲线：中间高、两边低、左右对称，有密度函数表达式可以看出当确定了该函数中的参数 u和σ，此时该函数将被唯一确定，相应的函数图象也随之唯一确定下来[6]. 因此当我们通过专家打分获得最有可能的工程时间历时 m 和最小值 a、最大值 b 后，需要确定相应的参数u和σ. 根据数据分布特点，我们设正态分布条件下的参数值为：$u_i=m_i,\sigma_i=(b_i-a_i)/(2*1.96)$. 此时第 i 步时间历时的分布就唯一确定下来了，相应的基于一组专家打分值，我们就可以唯一确定一组$t_i(i=1,2,\ldots, k)$的分布，此时对于该工程的总时间 $T=t_1+\ldots+t_k$ 的分布为 $T\sim N\left(\sum_{i=1}^{k}u_i,\sum_{i=1}^{k}\sigma_i^2\right)$ (证明相见附录 2)，将对应的 $u_i=m_i,\sigma_i=(b_i-a_i)/(2*1.96)$ 代入得到对应的表达式，相应的对应概率下的分位数为

$$T_q=Z_\alpha\sqrt{\Sigma\sigma_i^2}+\Sigma\mu_i=Z_\alpha\sqrt{\Sigma(b_i\text{-}a_i)/(2*1.96)^2}+\Sigma m_i$$

其中 Z_α 为标准正态分布下概率为∂的上分位数.

3 基于蒙特卡洛模拟IT工程时间风险度量

前文对假设软件开发工程的总时间在服从正态的情况下作了分析，但是由于 IT 工程的特点，其时间分布变量可以是服从不同分布的随机变量，同时不同的步骤之间可以是不同的分布，在这些情况下，要想推导相应的分布就变得比较复杂，有时甚至无法得到显性表达式. 因此这样的分析并不能适应于所有的情况，所以此处提出一个适应更为广泛的分析方法——基于蒙特卡罗模拟的工程时间风险度量方法，该方法可以用于每一步时间历时服从特定分布或者是分布未知以及时间变量不独立情况下的时间风险度量.

对于假定的随机分布来说，蒙特卡洛模拟的中心思想是在给定每一步时间历时服从的特定分布的条件下，通过随机抽样，产生一个随机样本(此处为 k 步的时间模拟值)，然后根据 ，可以得到一个 T 的随机样本，将此过程重复 n 次，可以得到 n 个 T 的值，有这些值可以估计出的分布形态，进而得到特定概率值下的时间风险度量值[7].

具体步骤如下:

a 根据提出的问题构造一个简单、适用的第 i 步时间历时分布函数，所构造分布函数在主要特征参量方面要与实际问题或系统相一致;

b 根据模型中各个随机变量的分布，在计算机上产生随机样本，实现一次模拟过程，产生一个随机样本(此处为 k 步的时间模拟值)，即 ，然后计算 得到一个 T 的样本;

c 重复以上步骤 n 次，可以得到 n 个关于 T 的样本;

d 根据这 n 个值可以绘制出 T 的直方图或者估计出 T 的分布函数;

e 基于第四步的结果计算出特定概率值下的时间风险度量值.

因此，可以通过蒙特卡洛模拟，对 IT 项目进度风险评估中存在的不确定性问题进行有效分析，可以给出任意概率下的时间风险度量值. 虽然此处是对随机变量 T 构建的研究框架，但是整个分析框架和思路可以直接平移到其他随机变量的分析上.

4 案例分析—以某公司财务软件开发的瀑布模型为例

4.1 软件开发的流程

下面本文就某软件公司利用瀑布模型开发财务软件为例，利用上面上文介绍的两种分路径对该项目的 IT 工程时间风险进行度量.

软件生命周期可以划分为计划、开发和运行三个时期，每一时期又细分为若干个子阶段，计划时期包括问题定义和可行性研究两个阶段，开发时期包括需求分析、软件设计、编码测试四个阶段，运行时期主要是系统维护阶段，图 1 列出了一个典型的软件生命周期模型. 由于我们实际应用中主要考虑计划和开发两个时期的时间风险，所以整个流程主要包括：问题定义、可行性研究、需求分析、软件设计、编码、测试等六个大步骤，这些步骤又可细化为若干小步骤，本文所研究的案例就化为了 40 个小步骤，这些步骤的时间历时的专家打分值详见附表 1.

表 1 每一个步骤的参数值表

均值	2.00	3.00	2.00	2.00	2.00	2.00	2.00	3.00	3.00	2.00	2.00	2.00	4.00
标准差	0.51	1.02	0.51	0.51	0.51	0.51	0.51	0.51	1.28	0.51	0.51	0.51	1.02

方差	0.26	1.04	0.26	0.26	0.26	0.26	0.26	0.26	1.63	0.26	0.26	0.26	1.04
均值	4.00	3.00	4.00	4.00	3.00	2.00	8.00	8.00	6.00	6.00	6.00	6.00	6.00
标准差	1.02	0.51	1.02	1.02	0.51	0.51	1.02	1.02	1.02	1.02	1.02	1.02	1.02
方差	1.04	0.26	1.04	1.04	0.26	0.26	1.04	1.04	1.04	1.04	1.04	1.04	1.04

4.2 基于正态分布假设条件下瀑布模型工程时间风险的度量

基于上文在正态分布假设条件下 IT 工程时间风险度量的讨论, 利用附表 1 的专家打分值, 根据公式 $u=m,\sigma=(b-a)/(2*1.96)$ 可以计算出每一步时间分布的均值和标准差. 详见表 1. 根据该表的计算结果以及正态分布的性质计算得到全部步骤时间历时 T 的参数值: $u=97$; $\sigma^2=17.5057$. 同时利用公式(*)可以很快计算出任意概率下的时间风险度量值. 具体计算结果详见表 2.

表 2　正态假设分布条件下的 IT 工程时间风险度量结果汇总表

分位数值	0.001	0.003	0.005	0.010	0.030	0.050	0.100	0.250
时间风险值	84.07	85.50	86.22	87.27	89.13	90.12	91.64	94.18
分位数值	0.500	0.750	0.900	0.950	0.970	0.995	0.997	0.999
时间风险值	97.00	99.82	102.36	103.88	104.87	107.78	108.50	109.93

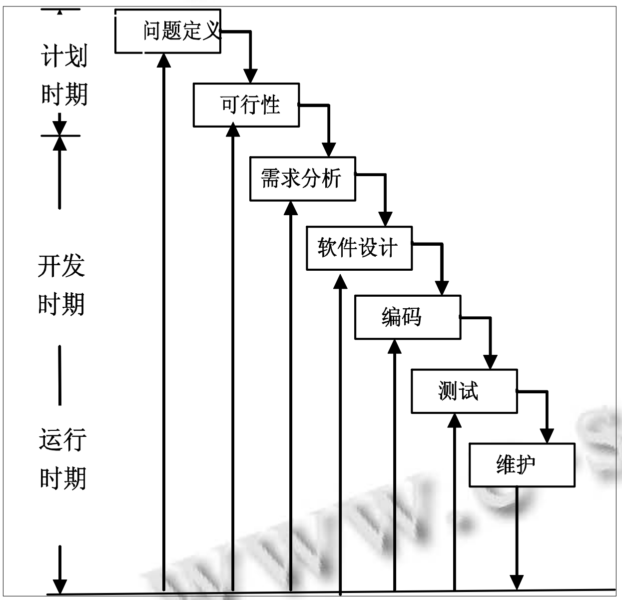

图 1　软件开发的瀑布模型[8]

4.3 基于蒙特卡洛模拟下的瀑布模型 IT 工程时间风险的度量

由于我们并不能保证每个 IT 工程的每一步的分布都服从正态分布, 所以此处本文将讨论每个小步时间历时分布分别服从均匀分布和正态分布两种情常见况下的任意概率下的时间风险度量值度量.

4.3.1 均匀分布的情况

首先假定第 i 步时间 $t_i(i=1,2,\cdots,k)$ 历时分布服从均匀分布, 然后根据 IT 工程工作细分及专家打分表(见附表 1)以及均匀分布的特征. 利用 R 软件分别产生每一步服从均匀分布 $U(a_i,b_i)$ 的 $t_{ij}(i=1,2,\cdots,k)$ 随机样本, 其中 $a_i=a,b_i=b$, 然后根据 $T_i=t_{1j}+\cdots+t_{kj}$ 计算得到一个 T 的样本; 将以上步骤重复 n(本文 n 为 10000 次, 即 $(j=1,2,\cdots,10000)$, 根据这 10000 个值可以绘制出 T 的直方图(详见图 2), 该分布的均值为: 97.4809, 标准差为: 4.7171; 同时基于 10000 个随机样本值可以计算出特定概率值下的时间风险度量值(详见表 3).

表 3　假设每个小步骤均服从均匀分布情况下不同概率下的 IT 工程时间风险汇总表

分位数值	0.001	0.003	0.005	0.010	0.030	0.050	0.100	0.250
时间风险值	83.08	84.58	85.35	86.50	88.64	89.74	91.42	94.28
分位数值	0.500	0.750	0.900	0.950	0.970	0.995	0.997	0.999
时间风险值	97.48	100.69	103.55	105.24	106.34	109.43	110.14	111.63

4.3.2 正态分布的情况

首先假定第 i 步时间 $t_i(i=1,2,\cdots,k)$ 历时分布服从

正态分布，然后根据 IT 工程工作细分及专家打分表(见附表 1)以及正态分布的特征. 利用 R 软件分别产生每一步服从正态分布 t_i 服从 $N\left(u_i,\sigma_i^2\right),\left(i=1,2,\cdots,k\right)$ 的 t_{ij} 随机样本，其中 $U_i=m_i,\sigma_i=\left(b_i-a_i\right)/\left(2\times1.96\right)$，然后根据计算得到一个 T 的样本；将以上步骤重复 n(本文 n 为 10000 次，即)，根据这 10000 个值可以绘制出 T 的直方图(详见图 3)，该分布的均值为：96.9968，标准差为：4.19135，这两个估计结果同均匀分布有一点不同，后面将会看到，该不同将会影响估计结果，同时这个值差别越大，IT 工程时间风险的度量结果相差将会越大；同时基于 10000 个随机样本值可以计算出特定概率值下的时间风险度量值(详见表 4).

表 4 假设每个小步骤均服从正态分布情况下不同概率下的 IT 工程时间风险汇总表

分位数值	0.001	0.003	0.005	0.010	0.030	0.050	0.100	0.250
时间风险值	84.08	85.49	86.16	87.24	89.13	90.11	91.61	94.18
分位数值	0.500	0.750	0.900	0.950	0.970	0.995	0.997	0.999
时间风险值	97.00	99.82	102.36	103.88	104.87	107.78	108.50	109.88

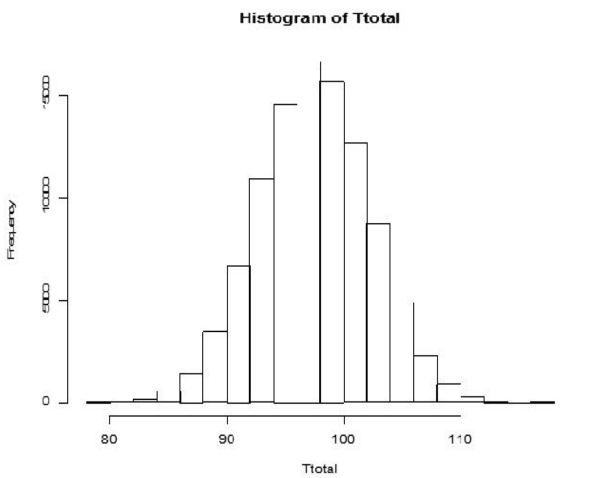

图 2 假设每个小步骤均服从均匀分布情况下的总时间历时分布直方图

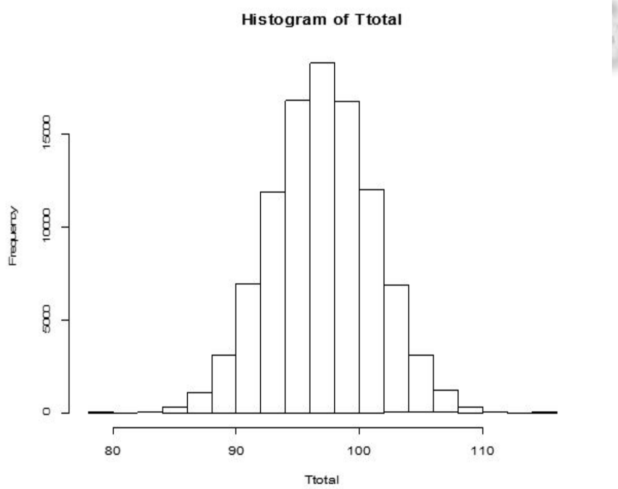

图 3 假设每个小步骤均服从正态分布情况下的总时间历时分布直方图

5 结论

综上所述，可以看出相对于基于专家打分的时间风险度量方法，本文给出的两种时间测度方法具有连续性的特征，这就为我们计算在任意概率下的精确 IT 工程时间风险度量值提供了保障. 就本文所讨论的方法而言，基于正态分布假设条件下的时间风险的度量方法理论上可以推广到任意的分布情况下，这里可以是每个步骤是基于相同的分布(例如文中假设正态分布)，也可以是不同的分布，只要该分布形态能够唯一、有效的确定下来就可以了，当然随着问题分布形式和分布种类变的复杂后，想得到文中类似(*)式的显性表达式就会变得困难，因此该方法虽然理论上适用于任意分布和任意分布组合的情况，但是本文不可能就所有情况一一讨论，并给出函数解析表达式，所以文中给出的第二个方法有效的解决了这个问题，基于蒙特卡洛模拟的时间风险度量方法可以快速而有效的解决上面遇到的问题，在任何情况下都可以很快的通过计算机得到相应概率下的 IT 工程时间风险度量值.

就文中所给的瀑布模型的计算结果可以看出，当分布形态给定为正态时，我们利用了两个分析方法计算相关的时间风险值，而且由于数据做了连续化处理，我们可以计算任意概率值下的 IT 工程时间风险度量值，这相对于以往度量方法有了交大的改善，同时结果显示两种方法在相同的假定分布下计算结果很接近，这个结果与理论研究结果相一致，这也就从另一个方面印证了基于蒙特卡洛模拟的计算结果精确度还是很理想的. 此外当分布形态发生改变时，这个结果会发

生改变，这个变化与不同分布形态差异相关，因为 IT 工程差异性很大，所以这时获得一个准确的时间分布假定就显得很重要.

综上所述，蒙特卡洛模拟方法既能达到其他方法的估计精度，同时适用面也更广. 因此，笔者建议在对时间分布未知的情况下，蒙特卡洛模拟方法是一个不错的选择. 此外，虽然本文仅就对 IT 工程时间风险进行了研究，但是该方法并不局限于对工程中时间风险的研究，只需要将文章中第三部分中，蒙特卡洛模拟步骤中的第 a 步中的数值设定为相应问题的参数即可，此后在第二步中就可以产生相应的模拟值，继续执行以下步骤就可以得到相应中随机变量的分布函数和相应概率下的风险值. 此外在计算大型项目多种风险集合时，应该考虑不同部分间的相关关系，由于偏方原因，就不在此处展开了.

参考文献

1 余坚,郑跃斌.信息系统开发过程风险管理的实施模型.计算机工程与应用,2002,12:109–112.

2 Stawicki J,李志民.方法论在标准软件开发与实施中的角色.项目管理技术,2008,6:70–72.

3 方德英,寇纪淞,李萍,秦立栓.IT 项目风险管理保障体系设计.商业研究,2006,10:51–54.

4 蔚林巍.21 世纪的项目管理.企业管理,2001,10:6–22.

5 钱鸿生.基于风险管理的软件生命周期模型研究[博士学位论文].上海:同济大学,2006.

6 茆诗松,周纪芗.概率论与数理统计.北京:中国统计出版社,2006.

7 尹增谦,管景峰,张晓宏,曹春梅.蒙特卡罗方法及应用.物理与工程,2002,12:45–49.

8 钱鸿生.基于风险管理的软件生命周期模型研究[博士学位论文].上海:同济大学,2006.

附录1

附表 1　IT 工程工作细分及专家打分表

开发阶段	开发流程	具体工作步骤	专家打分　（单位：天）		
			m	a	b
计划阶段	可行性研究	1、确定系统规模和目标	2	1	3
		2、研究目前正在使用的系统	3	1	5
		3、导出新系统的高层逻辑模型	2	1	3
		4、重新定义问题	2	1	3
		5、导出和评价供选择的方案	2	1	3
		6、确定一个方案并说明理由	2	1	3
		7、推出行动方案	2	1	3
		8、写计划书并审查	3	2	4
开发时期	需求分析	1、需求获取	3	1	6
		2 分析建模	2	1	3
		3 编写需求规格说明	2	1	3
		4、需求验证	2	1	3
	软件设计	1、模块的算法设计	4	2	6
		2、模块的数据结构设计	4	2	6
		3、数据接口设计	3	2	4
		4、模块测试用例设计	4	2	6
		5、编写详细设计说明书	4	2	6
		6、详细设计评审	3	2	4
	编码	1、编码语言选择	2	1	3
		2、编码风格（符合源程序文档化和数据说明）	8	6	10

		3、语言结构	8	6	10
		4、输入/输出	6	4	8
	测试	1、单元测试	6	4	8
		2、集成测试	6	4	8
		3、确认测试	6	4	8
		4、系统测试	6	4	8
运行时期	维护	1、维护申请报告	-	-	-
		2、维护工作实施	-	-	-
		3、维护文档整理	-	-	-
		4、维护活动评价	-	-	-

注: a 表示专家打分中的最小值, b 表示专家打分中的最大值, M 表示专家打分中的平均值

附录 2

设软件开发工程的总时间为T，流程中的每一细节工作所用的时间为$t_i(i=1,2,\cdots,k)$，同时$t_i \sim N(u_i,\sigma_i^2),(i=1,2\ldots,\ k)$，证明：当$i=k$时，$T=t_1+\ldots+t_i$时的分布为$T \sim N(\sum_{i=1}^{k}u_i,\sum_{i=1}^{k}\sigma_i^2)$.

证明：由于t_1和t_2在整个数轴上取值，故知其和T_1也在整个数轴上取值，利用卷积公式可得T_1的密度函数. 按卷积公式应先把t_1的密度函数$p_{t_1}(t_1)$中的t_1用T_1-t_2代替，而t_2的密度函数$p_{t_2}(t_2)$不变，带入卷积公式后，即得

$$\begin{aligned}p_{T_1}(T_1)&=\int_{-\infty}^{\infty}\left(\frac{1}{\sqrt{2\pi}\sigma_1}\exp\left\{-\frac{(T_1-t_2-u_1)^2}{2\sigma_1^2}\right\}\right)\times\left(\frac{1}{\sqrt{2\pi}\sigma_2}\exp\left\{-\frac{(t_2-u_2)^2}{2\sigma_2^2}\right\}\right)dy\\&=\frac{1}{2\pi\sigma_1\sigma_2}\int_{-\infty}^{\infty}\exp\left\{-\frac{1}{2}\left[\frac{(T_1-t_1-u_1)^2}{\sigma_1^2}+\frac{(t_2-u_2)^2}{\sigma_2^2}\right]\right\}dy\end{aligned}$$

经过一些代数运算，可以得到

$$\frac{(T_1-t_2-u_1)^2}{\sigma_1^2}+\frac{(t_2-u_2)^2}{\sigma_2^2}=\frac{(T_1-u_1-u_2)^2}{\sigma_1^2+\sigma_2^2}+A\left(T_1-\frac{B}{A}\right)$$

$$A=\frac{1}{\sigma_1^2}+\frac{1}{\sigma_2^2};\quad B=\frac{T_1-u_1}{\sigma_1^2}+\frac{u_2}{\sigma_2^2}$$

代回原式，可得

$$p_{T_1}(T_1)=\frac{1}{2\pi\sigma_1\sigma_2}\exp\left\{-\frac{1}{2}\frac{(T_1-u_1-u_2)^2}{\sigma_1^2+\sigma_2^2}\right\}$$

$$\int_{-\infty}^{\infty}\exp\left\{-\frac{A}{2}\left(t_2-\frac{B}{A}\right)^2\right\}dy$$

利用正态分布性质，上式中的积分等于$(2\pi/A)^{\frac{1}{2}}$. 于是

$$p_{T_1}(T_1)=\frac{1}{\sqrt{2\pi}\sqrt{\sigma_1^2+\sigma_2^2}}$$

$$\exp\left\{-\frac{1}{2}\frac{(T_1-u_1-u_2)^2}{\sigma_1^2+\sigma_2^2}\right\},(-\infty<T_1<\infty)$$

这正是均值为u_1+u_2，方差为$\sigma_1^2+\sigma_2^2$的正态分布. 这表明：两个独立的正态变量之和仍然是正态变量，其参数对应相加，即：

$$N(u_1,\sigma_1^2)+N(u_2,\sigma_2^2)=N(u_1+u_2,\sigma_1^2+\sigma_2^2)$$

由此我们得出当$i=3$时:

$$\begin{aligned}&N(u_1,\sigma_1^2)+N(u_2,\sigma_2^2)+N(u_3,\sigma_3^2)\\&=N(u_1+u_2+u_3,\sigma_1^2+\sigma_2^2+\sigma_3^2)\end{aligned}$$

有数学归纳法得到当$i=k$时：$T=t_1+\ldots+t_i$时T的分布为 $T \sim N(\sum_{i=1}^{k}u_i,\sum_{i=1}^{k}\sigma_i^2)$ 成立.

(全文完)

(上接第 188 页)

术,2008,6:39–41.

4 Claessens J. On the security of today's online electronic banking systems. Computer & Security, 2002,21(3):257–269.

5 Lee K, Kim J. DDoS attack detection method using cluster analysis.Science Direct, 2008,(34):1659–1665.

6 邱鹏,吴绍永.基于控制台的中间件负载均衡研究.计算机与应用,2005,4(2):32–35.

7 李云鹤,武善玉.基于 DDoS 防范的负载均衡集群设计与实现.网络通讯与安全,2007,(18):1522–1526.

8 邓全亮,薛霄.基于中间件技术的系统集成模型及应用.制造业自动化,2005,(27):76–80.