

# 一种基于 BP 神经网络的软件需求分析风险评估模型的研究

肖魏娜 张为群 王玲玲

(西南大学计算机与信息科学学院 重庆 400715) (重庆市智能软件与软件工程重点实验室 重庆 400715)

**摘要** 如何确定需求分析阶段潜在风险,并尽可能减少其带来的负面影响,是需求分析中的一个重要问题。提出了一套需求分析风险评估指标体系,在此基础上运用模糊理论与 BP 神经网络,建立了一种软件需求分析风险评估模型,最后通过试验验证了该方法的可行性和有效性。

**关键词** 需求分析,风险,模糊理论,BP 神经网络

## Study on Evaluation Model of Software Requirement Analysis Risk Based on Neural Network

XIAO Wei-na ZHANG Wei-qun WANG Ling-ling

(Faculty of Computer and Information Science, Southwest University, Chongqing 400715, China)

(Chongqing Intelligent Software and Software Engineering Laboratory, Chongqing 400715, China)

**Abstract** It is an important problem that how to ascertain the potential risks of the requirement analysis phases and reduce its opposite forces as low as possible in requirement analysis. We proposed an evaluation model of software requirement analysis risk, which combines Fuzzy Logic with BP neural network based on index system of risk assessment of requirement analysis. At last, the results of experiment demonstrate the feasibility and validity of the approach.

**Keywords** Requirement analysis, Risk, Fuzzy logic, BP neural network

### 1 引言

在软件开发过程中,需求分析是一个非常关键的阶段。其所要做的工作是深入描述软件的功能和性能,确定软件设计的限制和软件同其它系统元素的接口细节,定义软件的其它有效性需求。在这个过程中,只有确定了顾客的需求才能够分析和寻求新系统的解决方法。但是由于软件需求本身的隐含性,用户与开发者之间的沟通障碍,以及需求随着时间、用户的变化而变更等原因,可能使需求分析偏离实际需求而最终导致软件开发的失败,便产生了软件需求分析风险<sup>[1]</sup>。

需求问题导致的主要后果是返工。返工的成本占了总开发成本的 30%~50% (Boehm 和 Papaccio 1988)。而对于返工的情况,70%~80% 是因需求错误引起的 (Leffingwell 1997)。因此,要降低成本、满足用户的实际需求、提高开发进度、减少潜在问题发生的可能性及其带来的负面影响,就必须从软件需求分析开始进行有效的风险评估和管理。文献[1]提出了需求风险的控制方法;文献[2]集中了 7 种风险类型,提出一个定量的三级风险评估模型。但是现有风险评估模型的评估指标体系单一,并且基于经验的定性方法与基于历史数据的统计学方法没有有机地结合起来。本文提出了一种软件需求分析风险评估模型,该模型定义了 17 种风险指标,对各个风险指标采用模糊综合评判方法进行模糊预处理,并将其作为 BP 神经网络的输入节点,通过 BP 网络学习训练得到需求分析风险级别,为合理、准确地评价软件需求风险提供了

有效的辅助决策手段,从而降低了软件开发过程中的风险。

### 2 软件需求分析风险评估指标

指标是评估的工具,是反映评估对象属性的指示标志。软件需求分析风险评估指标体系是一系列风险评估指标的构成体,包括软件需求风险、人员需求风险、环境需求风险和需求本身的风险。本文将风险分为 5 个等级,评语集为  $V = \{v_1, v_2, v_3, v_4, v_5\}$ , 分别代表高、较高、中等、较低、低。17 种风险指标如表 1 所列。

表 1 17 种风险指标

类别	编号	风险名称
软件需求风险 $R_S$	$R_1$	功能需求风险
	$R_2$	性能需求风险
	$R_3$	可用性需求风险
	$R_4$	安全性需求风险
	$R_5$	资源需求风险
	$R_6$	成本及进度需求风险
	$R_7$	可重用性需求风险
	$R_8$	文档资料需求风险
人员需求风险 $R_P$	$R_9$	用户需求风险
	$R_{10}$	开发团队需求风险
环境需求风险 $R_E$	$R_{11}$	外部环境需求风险
	$R_{12}$	完整性需求风险
需求(本身)风险 $R_R$	$R_{13}$	正确性需求风险
	$R_{14}$	一致性需求风险
	$R_{15}$	可修改性需求风险
	$R_{16}$	可跟踪性需求风险
	$R_{17}$	可验证性需求风险

到稿日期:2010-05-14 返修日期:2010-08-12 本文受重庆市自然科学基金项目(CSTC,2006BA2003)资助。

肖魏娜 女,硕士生,主要研究方向为软件工程等,E-mail:anna623@swu.edu.cn;张为群 男,教授,主要研究方向为软件工程、形式语言、软件测试等;王玲玲 女,硕士生,主要研究方向为软件工程等。

定义 1(软件需求分析风险评估指标体系  $R_{\text{总}}$ )

$$R_{\text{总}} = \{R_S, R_P, R_E, R_R\}$$

式中,  $R_{\text{总}}$  是风险评估指标的集合,  $R_S$  为软件需求风险指标,  $R_P$  为人员需求风险指标,  $R_E$  为环境需求风险指标,  $R_R$  为需求本身风险指标。

定义 2(软件需求分析风险评估指标权重  $W$ )

$$W = \{w_m | 1 \leq m \leq 17\}$$

式中,  $w_m$  表示在评估过程中第  $m$  个指标在整体评价中的相对重要程度。其中  $\sum_{m=1}^{17} w_m = 1$ 。

定义 3(软件需求风险  $R_S$ )

$$f: X \rightarrow Y, X = \{x | 1 \leq x \leq 8\}, Y = \{R_x | 1 \leq x \leq 8\}$$

式中,  $R_1$  = 功能需求风险,  $R_2$  = 性能需求风险,  $R_3$  = 可用性需求风险,  $R_4$  = 安全性需求风险,  $R_5$  = 资源需求风险,  $R_6$  = 成本及进度需求风险,  $R_7$  = 可重用性需求风险,  $R_8$  = 文档资料需求风险。

定义 4(功能需求风险  $R_1$ )

$$R_1 = \frac{f_r}{f_t}$$

该指标是指在项目开发之初往往不能准确、全面地定义系统的功能需求, 并且在开发过程中, 用户对系统功能的需求发生变更而承担的风险。其中  $f_t$  为系统总功能数,  $f_r$  为具有风险的功能数。

定义 5(性能需求风险  $R_2$ )

$$R_2 = \{T, F, P\}$$

该指标是指由于系统对响应时间、吞吐量、并发用户数等方面要求苛刻, 造成对设计软件策略和选择硬件的严重影响而承担的风险。其中  $T$  为响应时间, 是指系统从请求发出开始到客户端接收到数据所消耗的时间;  $F$  为吞吐量, 是指单位时间内系统处理的客户请求的数量,  $F = \frac{N_{\text{虚}} \times R}{T}$ ,  $N_{\text{虚}}$  表示虚拟用户个数,  $R$  表示每个虚拟用户发出的请求数量;  $T$  表示性能测试所用的时间。  $P$  表示并发用户数, 是指在同一个时间段内访问系统的用户数量,  $P = \frac{nL}{T}$ ,  $n$  是用户登录到退出系统的时间段的数量;  $L$  是  $n$  的平均长度。系统对性能要求越高, 则其风险越高。

定义 6(可用性需求风险  $R_3$ )

$$R_3 = 1 - \frac{MTTF}{MTTF + MTTR}$$

可用性需求是衡量预定的可用时间, 要求系统在这段时间内是真正可用并且是完全可以操作的。可用性需求越高, 潜在风险越大。其中  $MTTF$  为平均无故障时间,  $MTTR$  为故障发生后所用的故障修复时间。

定义 7(安全性需求风险  $R_4$ )

$$R_4 = \frac{f_i + f_p}{f_t}$$

该指标是指系统因防止被非法访问以及非法使用其创建或输入的数据, 而进行系统特殊设置所承担的风险。其中  $f_i$  表示防止系统发生损失而设置特殊功能的个数,  $f_p$  表示预防系统潜在危险而设置特殊功能的个数。

定义 8(资源需求风险  $R_5$ )

$$R_5 = \frac{RR + DR}{RT}$$

该指标是指系统对运行时所需的数据、软件、内存空间等各项资源及软件开发时所需的人力、支撑软件、开发设备等资源需求而承担的风险。开发时和运行时资源需求高, 则资源需求风险大。其中  $RR$  表示系统运行时资源需求量,  $DR$  表示系统开发时资源需求量,  $RT$  表示总的资源可用量。

定义 9(成本及进度需求风险  $R_6$ )

$$R_6 = \frac{1}{q_E * EED + q_M * EMD}$$

该指标是指因缺乏估算经验或模型导致系统没有满足成本、进度和质量目标的要求而承担的风险。估算经验和估算模型越不成熟, 风险越高。其中  $EED$  表示估算经验的成熟度,  $EMD$  表示估算模型的成熟度。  $q_E + q_M = 1$ , 分别指成本及进度需求两个方面的权重。

定义 10(可重用性需求风险  $R_7$ )

$$R_7 = \frac{STM - SRM}{STM}$$

该指标是指重用软件系统或其一部分的难度过高而承担的风险。其中  $SRM$  表示系统可重用模块数,  $STM$  系统总模块数。

定义 11(文档资料需求风险  $R_8$ )

$$R_8 = \frac{1}{DA}$$

文档资料需求风险是在需求分析阶段所产生的文档资料, 如需求规格说明书、开发计划和用户手册等文档资料因不齐全、不完备而承担的风险。三分之一(以上)文档资料不齐全, 则风险较高(高)。其中  $DA$  表示文档资料完备程度。

定义 12(人员需求风险  $R_P$ )

$$g: X \rightarrow Y, X = \{x | 9 \leq x \leq 10\}, Y = \{R_x | 9 \leq x \leq 10\}$$

式中,  $R_9$  = 用户需求风险,  $R_{10}$  = 开发团队需求风险。

1. 用户需求风险是指由于省略某类用户、用户参与不足以及用户需求模糊而承担的风险。三分之一(以上)用户没有参与, 则风险较高(高),  $R_9 = q_{U1} MU + q_{U2} UU$ , 其中  $MU$  表示管理者需求,  $UU$  表示使用者需求。  $q_{U1} + q_{U2} = 1$ , 分别指用户需求 2 个方面的权重。

2. 开发团队在整个软件开发过程中起着关键作用。  $R_{10} = q_{C1} PE + q_{C2} GC + q_{C3} DE + q_{C4} PT$ ,  $PE$  表示人员配备,  $GC$  表示团队协作,  $DE$  表示开发经验,  $PT$  表示专业培训。  $\sum_{i=1}^4 q_{Ci} = 1$ , 分别指开发团队需求 4 个方面的权重。

定义 13(环境需求风险  $R_E$ )

$$R_E = q_B * BR + q_L * RL + q_D * RDE$$

该指标是指系统必须遵循业务规则、法律法规或受系统开发环境限制而承担的风险。三分之一(以上)需求受到环境限制, 则风险较高(高)。其中  $BR$  表示业务规则,  $RL$  表示法律法规,  $RDE$  表示开发环境,  $q_B + q_L + q_D = 1$ , 分别指环境需求 3 个方面的权重。

定义 14(需求风险  $R_R$ )

$$R_R = \frac{1}{q_1 * R_i + q_2 * R_c + q_3 * R_u + q_4 * R_m + q_5 * R_a + q_6 * R_v}$$

该指标是指系统需求描述没有满足其完整性、正确性、一致性、可修改性、无歧义性和可验证性而承担的风险。三分之一(以上)需求没有满足这 6 个属性, 则风险较高(高)。其中  $R_i$  表示需求完整性,  $R_c$  表示需求正确性,  $R_u$  表示需求一致性,  $R_m$  表示需求可修改性,  $R_a$  表示需求无歧义性,  $R_v$  表示需

求可验证性。 $\sum_{i=1}^6 q_i = 1$ , 分别指系统需求 6 个方面的权重。

### 3 软件需求分析风险评估模型

本文提出的风险评估模型包括两个部分:一是历史风险数据库。它不仅记录了当前项目风险信息,还存储了领域专家风险知识及以往对需求分析的风险状态观测的信息。二是采用模糊理论与神经网络相结合对风险等级进行评估。由于各个风险评价指标与风险级别之间存在非线性关系和动态变化规律,模糊系统具有容易被人理解的表达能力,神经网络可以实现从输入到输出的任意复杂的非线性映射关系,因此两者结合可以提高整个系统的学习能力和表达能力,从而为需求分析风险进行有效预测。

#### 3.1 软件需求分析风险识别

风险识别是整个风险评估过程的第一步,它是对软件需求分析阶段进行考察研究,从而识别并记录有关风险。其基本任务就是识别出需求分析阶段存在的潜在风险。通过对比风险评估指标,搜索历史风险数据库历史案例,同时结合专家经验知识,确定需求分析阶段存在的风险,生成风险列表。具体流程如图 1 所示。

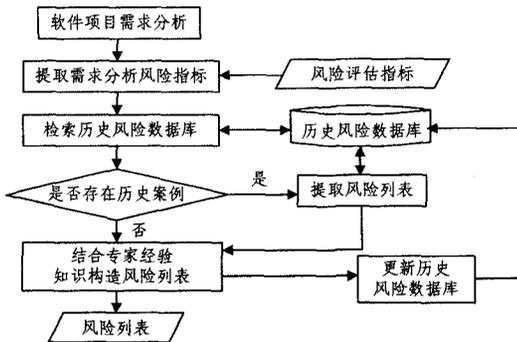


图 1 软件需求分析风险识别流程

其中,历史风险数据库用来跟踪风险、记录风险并对风险进行简单查询和统计,同时它还存储了两部分知识:一是已经被实践所证实是正确的领域专家知识;二是针对数据库中已有项目风险数据进行归纳的能够解决特定领域问题的知识。

#### 3.2 软件需求分析风险分析

软件需求分析风险分析是指人们应用风险分析工具,对已识别出的各类风险进一步细化风险描述。其目的是将风险及风险背景明晰化,从而可以根据软件需求分析风险评估模型自身的准则来确定风险等级及其影响。

由于 BP 神经网络适合对定量输入数据进行分析,定性的风险指标不确定性很大,因此在该阶段使用模糊理论对各风险指标进行量化,经过模糊预处理后其值在  $(0, 1]$ , 从而作为 BP 神经网络的输入。令  $r_{ij}$  表示风险因素  $R_i$  的隶属度,  $R$  表示综合评判矩阵,  $B_i$  表示  $R_i$  的模糊综合评判集,  $X = \{b_1, b_2, \dots, b_{17}\}$  表示经过预处理的 17 个指标值。具体算法如下:

输入:  $R_{\text{总}}, V = \{v_1, v_2, v_3, v_4, v_5\}$

输出:  $X = \{b_1, b_2, \dots, b_{17}\}$

Step1 根据  $V = \{v_1, v_2, \dots, v_5\}$ , 求  $r_{ij}$ 。  $f: R_{\text{总}} \rightarrow F(V), R_i \mapsto f(R_i)$ 。其中  $f(R_i) = r_i = (r_{i1}, r_{i2}, \dots, r_{im}), \forall i = 1, 2, \dots, 17$ 。

Step2 若  $i = 17$ , 由  $R_f \in F(R_{\text{总}} \times V)$ , 构造  $R, R = R_f = (r_{ij})_{17 \times n}$ ; 否则转 Step1。

Step3 根据  $W$ , 由模糊线性变换计算出  $B_i = \{b_1, b_2, \dots, b_n\}, \forall i =$

$1, 2, \dots, 17$ 。即

$$b_j = (w_{11} \wedge r_{1j}) + (w_{21} \wedge r_{2j}) + \dots + (w_{171} \wedge r_{17j}), j = 1, 2, \dots, n$$

得到  $b_m, b_m = \max\{b_i | b_i \in B_i \wedge 1 \leq i \leq n\}$ 。

Step4 若  $i = 17$ , 则得到  $X = \{b_1, b_2, \dots, b_{17}\}$ , 输出  $X$ 。算法结束, 否则转 Step3。

#### 3.3 BP 神经网络模型设计

BP 算法是一种用于前向多层神经网络的反传学习算法, 由鲁梅尔哈特(D. Rumelhart)等于 1986 年提出。采用 BP 算法的多层感知器神经网络模型一般称为 BP 神经网络, 它由输入层、隐层、输出层组成。

A) BP 神经网络输入层、输出层

BP 神经网络输入层节点为 17 个经过模糊预处理的需求分析风险评估指标。分别以  $b_i$  表示, 其中  $i = 1, 2, \dots, 17$ 。

BP 神经网络输出层节点为需求分析风险级别, 因此输出个数为 1, 以  $y$  表示。

B) BP 神经网络隐含层数

隐含层具有抽象作用, 它从输入中提取特征。增加隐含层有利于提高 BP 神经网络的性能、降低误差, 但同时也会使网络复杂化。1988 年 Cybenko 指出各节点采用 S 型函数, 一个隐含层就足以实现任意判决分类的问题, 所以评估模型可采用一个隐含层<sup>[8,9]</sup>。

C) BP 神经网络隐含层节点数

隐节点的作用是从样本中提取并存储内在规律, 每个隐节点都有若干个权值, 而每个都是增强网络映射能力的一个参数。确定最佳隐节点数可以参考一些经验公式<sup>[10]</sup>:

$$m = \sqrt{n+l} + \alpha \quad m = \log_2 n \quad m = \sqrt{nl}$$

式中,  $m$  为隐层节点数;  $n$  为输入层节点数;  $l$  为输出层节点数;  $\alpha$  为 1~10 之间的常数。根据试验, 模型隐层节点数为 15 个。

#### 3.4 基于 BP 神经网络软件需求分析风险评估模型算法流程

本文采用累计误差的批处理方式的 BP 算法进行训练, 由于遵循了减小全局误差为目标的“集体主义”原则, 因此批训练可以保证总误差向减小方向变化。令  $V$  表示输入层到隐含层之间的权值矩阵,  $W$  表示隐含层到输出层之间的权值矩阵,  $p$  表示样本模式计数器,  $q$  表示训练次数计数器,  $E$  表示误差,  $\eta$  表示学习率,  $E_{\text{min}}$  表示网络训练后达到的精度,  $X_p$  表示当前样本,  $X$  表示输入向量数组,  $d$  表示期望输出向量数组,  $P$  为训练样本总数,  $l$  为输出层节点数。具体步骤如下:

输入:  $X = \{b_1, b_2, \dots, b_{17}\}$

输出:  $y$

Step1 初始化。对  $V, W$  赋随机值, 将  $p = 1, q = 1, E = 0, \eta = 0.01, E_{\text{min}} = 0.001$ 。

Step2 计算各层输出。用  $X_p$  对  $X, d$  赋值, 通过输出层和隐含层公式:

$$\delta_k^o = -\frac{\partial E}{\partial \text{net}_k} \quad \delta_j^h = -\frac{\partial E}{\partial \text{net}_j}$$

计算出隐含层输出向量  $Y$ 、输出层输出向量  $O$  中各分量。

Step3 根据  $E_{\text{总}} = \sqrt{\frac{1}{2} \sum_{p=1}^P \sum_{k=1}^l (d_k^p - o_k^p)^2}$  计算网络总误差  $E_{\text{总}}$ 。

Step4 若  $p < P$ , 则  $p = p + 1$ , 转 Step2。

Step5 计算各层误差信号。

$$\delta_k^o = (d_k - o_k)(1 - o_k)o_k$$

$$\delta_j^h = \left( \sum_{k=1}^l \delta_k^o w_{jk} \right) (1 - y_j)y_j$$

Step6 根据公式:

$$w_{jk} \leftarrow w_{jk} + \eta \delta_j^o y_j; \quad v_{ij} \leftarrow v_{ij} + \eta \delta_j^h x_i$$

计算出  $W, V$  中各分量, 调整各层权值。

Step7  $q=q+1$ 。

Step8 若  $E_{RME} < E_{min}$ , 算法结束; 否则  $E=0, p=1$ , 转 Step2。

图 2 给出基于 BP 神经网络软件需求分析风险评估模型流程。

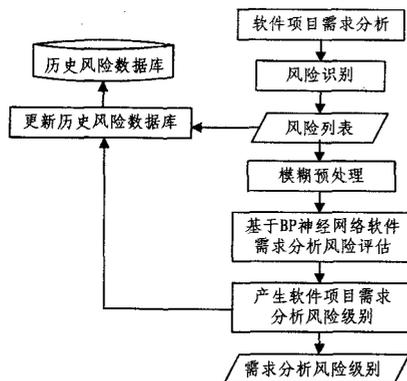


图 2 基于 BP 神经网络软件需求分析风险评估模型流程

#### 4 仿真试验

为验证上述方法的可行性与有效性, 我们用仿真实验预测软件需求风险。仿真 10 个已完成需求分析的项目作为样本, 经过训练实验, BP 神经网络模型达到预设精度。现以某

档案管理系统的需求分析作为测试样本。该系统需要将档案管理规范化、集成化, 并有效共享文件资源, 同时要防止档案信息泄露或被篡改数据; 系统档案资料基本齐全。根据本文提出的模型, 进行风险识别, 由于没有历史案例, 则直接根据风险评估指标, 结合 10 位专家经验知识, 产生风险列表, 如表 2 所列。

表 2 档案管理系统需求分析风险列表

专家	1	2	3	4	5	6	7	8	9	10
R <sub>1</sub>	v <sub>4</sub>									
R <sub>2</sub>	v <sub>4</sub>									
R <sub>3</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>3</sub>
R <sub>4</sub>	v <sub>2</sub>									
R <sub>5</sub>	v <sub>4</sub>									
R <sub>6</sub>	v <sub>2</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>2</sub>	v <sub>2</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>2</sub>	v <sub>3</sub>
R <sub>7</sub>	v <sub>2</sub>									
R <sub>8</sub>	v <sub>5</sub>									
R <sub>9</sub>	v <sub>2</sub>	v <sub>2</sub>	v <sub>3</sub>	v <sub>2</sub>	v <sub>3</sub>	v <sub>2</sub>	v <sub>3</sub>	v <sub>2</sub>	v <sub>2</sub>	v <sub>2</sub>
R <sub>10</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>2</sub>	v <sub>3</sub>	v <sub>2</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>3</sub>
R <sub>11</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>2</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>2</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>3</sub>
R <sub>12</sub>	v <sub>3</sub>	v <sub>4</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>4</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>4</sub>	v <sub>4</sub>
R <sub>13</sub>	v <sub>3</sub>	v <sub>4</sub>	v <sub>3</sub>	v <sub>4</sub>	v <sub>4</sub>	v <sub>3</sub>				
R <sub>14</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>4</sub>	v <sub>3</sub>	v <sub>4</sub>	v <sub>4</sub>	v <sub>4</sub>	v <sub>3</sub>
R <sub>15</sub>	v <sub>4</sub>	v <sub>4</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>4</sub>	v <sub>4</sub>	v <sub>4</sub>	v <sub>3</sub>	v <sub>3</sub>	v <sub>4</sub>
R <sub>16</sub>	v <sub>3</sub>	v <sub>2</sub>	v <sub>2</sub>	v <sub>2</sub>	v <sub>3</sub>	v <sub>2</sub>				
R <sub>17</sub>	v <sub>5</sub>	v <sub>4</sub>	v <sub>5</sub>	v <sub>5</sub>	v <sub>5</sub>	v <sub>4</sub>	v <sub>4</sub>	v <sub>5</sub>	v <sub>4</sub>	v <sub>5</sub>

各项风险指标值经过模糊预处理后, 作为神经网络输入值, 如表 3 所列。

表 3 预处理结果

指标	b <sub>1</sub>	b <sub>2</sub>	b <sub>3</sub>	b <sub>4</sub>	b <sub>5</sub>	b <sub>6</sub>	b <sub>7</sub>	b <sub>8</sub>	b <sub>9</sub>	b <sub>10</sub>	b <sub>11</sub>	b <sub>12</sub>	b <sub>13</sub>	b <sub>14</sub>	b <sub>15</sub>	b <sub>16</sub>	b <sub>17</sub>
结果	0.3	0.3	0.5	0.7	0.3	0.5	0.7	0.1	0.7	0.5	0.5	0.5	0.5	0.5	0.3	0.5	0.1

经过本文提出的模型进行风险预测, 该系统需求分析风险值为 0.72, 风险级别较高。对比文献[3]提出的软件需求风险评估模型, 其风险指标值由专家打分取平均分获得。而本文提出的模型, 对定性的风险指标值使用模糊理论进行量化。将模糊理论与 BP 神经网络结合预测软件需求分析风险等级, 提高了整个系统的预测精确度, 为合理、准确地评价软件需求风险提供了有效的辅助决策手段, 从而降低了软件开发过程中的风险。

**结束语** 本文构建了软件需求风险评估指标, 并建立了软件需求分析风险评估模型。与传统软件需求分析风险评估模型相比, 该模型不仅评价指标全面, 而且将基于经验的定性方法与基于历史数据的统计学方法有机地结合起来。利用此模型对软件需求风险进行预测, 可以更科学地进行风险应对和管理, 从而降低软件开发过程中的风险。

#### 参考文献

[1] 付玉, 邱冠周, 冯其明, 等. 应用软件开发的需求风险及控制[J]. 计算机工程与应用, 2005(13): 91-93  
 [2] 青琳. 软件项目的风险评价模型[J]. 计算机工程与应用, 2004(13)  
 [3] 潘梅森, 熊齐. 基于 SVM 的软件需求分析风险评估模型[J]. 计算机工程, 2008(9)

[4] 周彦晖, 张为群. 软件形式化与可视化软件模型的转换[J]. 计算机科学, 2003, 30(9)  
 [5] Boehm B W. Software risk management: principles and practices [J]. IEEE Software, 2007, 8(1): 32-41  
 [6] Cortellessa V, Goseva-Popstojanova K, Guedem A R, et al. Model-based Performance Risk Analysis [J]. IEEE Transactions on Software Engineering, 2005, 31(1): 3-20  
 [7] Zhao Dong-mei, Wang Jing-hong, Ma Jian-Feng. Fuzzy risk assessment of network security[C]//Proceedings of the 4th International Conference on Machine Learning and Cybernetics. Dalian, August 2006: 4400-4405  
 [8] 葛哲学, 孙志强. 神经网络理论与 MATLAB R2007[M]. 北京: 电子工业出版社, 2007: 108-116  
 [9] 高隽. 神经网络原理及仿真实例[M]. 北京: 机械工业出版社, 2003: 151-153  
 [10] 韩力群. 神经网络理论、设计及应用[M]. 北京: 化学工业出版社, 2009: 59-61  
 [11] 田雨波. 混合神经网络技术[M]. 北京: 科学出版社, 2009: 33  
 [12] Charette R. Software engineering risk analysis and management [M]. New York: Mc Graw Hill, 2006: 178-189  
 [13] 周成容. BP 神经网络的模糊改进及应用[J]. 重庆工学院学报: 自然科学版, 2008, 22(6): 153-156