

# 基于 CMMI 的软件项目风险缓解策略优化模型研究<sup>\*</sup>

潘春光 陈英武

(国防科技大学信息系统与管理学院 长沙 410073)

**摘要** 在基于 CMMI 的软件工程实践过程中,风险管理被集成在 CMMI 框架下的第三级,与过程管理融于同一个框架之中。因此,基于 CMMI 的软件项目风险管理能够得到来自组织的有效信息,使软件开发朝着有规律、可预测的方向发展。本文针对目前基于 CMMI 的软件项目风险管理尚缺乏有效技术支持的现状,初步探讨了风险响应计划的制定问题,并针对实际提出一种风险缓解策略优化模型,为软件项目管理人员的风险决策提供辅助手段。

**关键词** 软件项目,风险管理,能力成熟度模型集成,风险缓解策略

## Research on the Optimization Model of Risk Reduction Actions in the CMMI-based SPRM

PAN Chun-Guang CHEN Ying-Wu

(College of Information System and Management, National University of Defense Technology, Changsha 410073)

**Abstract** In CMMI, risk management is in the third capability maturity level, which is in the same framework with process management. So, the CMMI-based SPRM (software project risk management) can utilize some useful information of software organizations and make the software development in the direction of regularity and prediction. But at present, the risk management methods in the CMMI-based SPRM are still incomplete. In this paper, a simple discussion of risk response planning is made and a optimization model for selecting risk reduction actions is presented, which provides a risk decision method for software project management people.

**Keywords** Software project, Risk management, CMMI, Risk reduction action

软件项目风险管理是软件工程领域中新兴起的一门学科,经过近 30 年的发展,在理论、方法乃至实践上都取得了一定进展。目前,以软件过程改进技术为标志的软件工程实践取得了巨大的成功,使得软件项目风险管理的研究受益匪浅。其中,美国卡内基·梅隆大学软件工程研究所(Software Engineering Institution, SEI)提出的能力成熟度模型集成(Capability Maturity Model Integration, CMMI)是一种普遍认可的过程改进模型,它将风险管理集成于 CMMI 框架的第三能力成熟度等级,使软件风险管理的研究置于过程改进的框架之下。本文在研究基于 CMMI 的软件项目风险管理的过程中,对风险响应计划制定阶段如何实施风险缓解策略的优选问题进行了研究,提出了一种风险缓解策略优化模型,为软件项目管理人员进行风险决策提供了辅助手段。

### 1 基于 CMMI 的软件项目风险管理

CMMI 是 SEI 研究的一个过程改进模型,它来源于 CMM(Capability Maturity Model),后来由于 SEI 开发的多个模型同时使用,限制了组织过程改进的能力,因此暂停了 SW-CMM 2.0 版的推进,开始集中开发 CMMI 项目。

软件过程改进的研究使软件项目风险管理受益匪浅。将风险管理理论与软件过程改进技术相结合,成为目前软件项目风险管理研究的热点和发展趋势<sup>[1]</sup>。其中,以 SEI 开展的基于 CMMI 的软件项目风险管理是其主要研究方向之一。CMMI 在 CMM 基础上进行了扩展,包括了 5 个过程成熟度等级和 25 个关键过程域(KPA)。CMMI 的五级模型有初始级、已管理级、严格定义级、定量管理级和优化级,其中风险管

理作为一个新的、独立的过程域出现在第三个成熟度等级(严格定义级)中,成为衡量软件组织成熟度能力的一项过程标准。

基于 CMMI 的软件项目风险管理在很大程度上克服了以往风险管理框架模式的缺陷。它以 CMMI 框架为依托,提出了基于过程的风险管理,使风险管理与过程改进融入到一个体系之中。它使软件项目风险管理的理论和方法与软件工程实践紧密结合起来,使风险管理更具有针对性和实用性。

目前,基于 CMMI 的软件项目风险管理仍处在框架模型阶段。如何进一步将风险管理的理论和方法融入到 CMMI 过程改进中,具有非常重要的现实意义。本文在对 CMMI 的软件项目风险管理研究过程中,在风险响应计划制定阶段,对如何优选风险缓解策略进行研究,从而为组织节约成本,提高软件开发项目的成功率提供技术支持。

### 2 问题描述

在 CMMI 中,风险响应计划往往是在项目计划制定的基础上开展的。这时,软件需求已经基本分析清楚并明确定义,项目可根据软件需求分析说明书进行工作任务分解,并准备按照工作分解结构(Work Breakdown Structure, WBS)实施软件开发。

设按照软件项目需求规格说明书,风险管理人员给出了项目工作分解结构。同时,通过查找软件组织资产库和风险仓库、组织有关专家进行头脑风暴等一系列手段对该项目进行了风险分析并达成一致意见。

通过风险分析,项目管理人员基于工作分解结构,给出了

<sup>\*</sup>国家自然科学基金资助项目(70272002)。潘春光 博士研究生,研究方向:软件项目风险管理、决策分析技术等。

在每个分解元素中可能出现的风险事件及发生概率。同时,对每项风险事件发生时可能对项目各分解元素造成的影响给出了影响值(用经济成本度量)。根据以往历史记录和专家经验,项目管理人员给出了一系列的风险缓解策略。由于每项缓解策略的实施都要花费一定的成本(实施风险缓解策略的花费由专家估算给出),而实施每一项风险缓解策略就可能降低风险事件发生的概率,同时可能减少风险造成的损失。但总体来看,实施风险缓解策略能否确实达到降低项目总成本的目的,从而使实施风险缓解策略有意义,就成为项目管理人员需要着重考虑的问题。而选择哪些风险缓解策略才能最有效地控制风险、降低项目花费,是一个优化问题,需要风险管理人员做出决策。下面就对这一问题建立风险缓解策略优选模型。

### 3 模型的建立

设软件项目分解元素集为  $S$ , 其元素为  $s=1, 2, \dots, s$ ; 风险事件集为  $R$ , 其元素为  $r=1, 2, \dots, R$ 。定义风险概率矩阵为  $P=(P_{r,s})_{R \times S}$ , 其元素  $P_{r,s}$  每个项目分解元素  $s$  中可能发生风险事件  $r$  的概率。再定义风险损失矩阵为  $M=(m_{r,w})_{R \times W}$ , 矩阵元素  $m_{r,w}$  表示风险事件  $r$  对项目工作分解元素  $w$  造成的风险损失。风险损失矩阵度量了风险事件对项目的影 响, 这里的损失也有可能是收益, 表示风险机遇。定义期望风险损失矩阵为  $G$ ,  $G$  用风险概率矩阵的转置和风险损失矩阵的乘积来度量, 即  $G=P' * M$ , 表示每个风险事件造成的风险损失的期望, 则所有期望风险损失矩阵元素之和为整个项目的风险损失。

在风险分析之后, 项目管理人员据实际情况给出风险缓解策略集  $A\{a; a=1, 2, \dots, A\}$ , 而实施风险缓解策略需花费一定成本, 记  $c_a$  为采取风险缓解策略  $a$  所花费的成本。采用风险缓解策略的目的是为了降低风险、减少风险损失。但由于软件项目风险的复杂性, 致使某些实施风险缓解策略未必一定能达到降低风险的目的。而且, 实施风险缓解策略所花费的成本如果大于风险损失的话, 则采用风险缓解策略就没有任何意义。因此, 需要基于项目的成本来考虑如何选择风险缓解策略, 使项目的整个花费最小化。

设给出风险概率矩阵  $P=(p_{r,s})_{R \times S}$ , 风险损失矩阵  $M=(m_{r,w})_{R \times W}$ , 在未采取任何风险缓解策略的条件下, 可以得到期望风险损失矩阵为:

$$G=P' \times M=(g_{s,w})_{S \times W}=(\sum_{r=1}^R(p_{r,s} \times m_{r,w}))_{S \times W} \quad (1)$$

项目总的风险损失为

$$ERL=\sum_{s=1}^S \sum_{w=1}^W (g_{s,w})=\sum_{s=1}^S \sum_{w=1}^W (\sum_{r=1}^R p_{r,s} \times m_{r,w}) \quad (2)$$

项目管理人员经过讨论给出了采用每个风险缓解策略时要花费的成本, 并记第  $a$  个风险缓解策略需要花费的风险缓解成本为  $c_a$ , 则有风险缓解策略成本向量  $C=(c_1, c_2, \dots, c_A)$ 。再令  $X_{a \times a}$  为对角矩阵, 该矩阵的第  $a$  行、第  $a$  列的元素为 1, 其余的元素全为 0。称  $X_{a \times a}$  为策略选择矩阵, 表示项目管理人员选择第  $a$  个风险缓解策略, 并令  $e_a$  为单位列向量, 其中  $e_a$  的第  $a$  个元素为 1, 其余的元素为 0。

采用风险缓解策略的总成本可以用式(3)表示:

$$AAC(X)=C \cdot X_{i,i} \cdot e_i=\sum_{i=1}^A c_i \cdot x_{i,i} \quad (3)$$

现记选择策略  $a$  后对风险概率矩阵  $P$  中的元素  $p_{r,s}$  的影响值为  $v_{r,s,a}$ , 则有向量  $V_{r,s}=(v_{r,s,1}, v_{r,s,2}, \dots, v_{r,s,A})$  为采用所

有的风险缓解策略后对元素  $p_{r,s}$  的影响向量; 再记选择策略  $a$  后对风险损失矩阵  $M$  中的元素  $m_{r,w}$  的影响值为  $u_{r,w,a}$ , 则有向量  $U_{r,w}=(u_{r,w,1}, u_{r,w,2}, \dots, u_{r,w,A})$  为采用所有的风险缓解策略后对元素  $m_{r,w}$  的影响向量。

$$\text{令 } v_{r,s}=\prod_{\substack{j=1,2,\dots,A \\ x_{j,j} \neq 0}} v_{r,s,j} \cdot x_{j,j} \text{ 再令对风险概率矩阵 } P \text{ 的影响}$$

函数为  $f(p_{r,s}, v_{r,s})$ , 取  $f(p_{r,s}, v_{r,s})=p_{r,s} \times v_{r,s}$ , 则采用风险缓解策略后的风险概率矩阵可用(4)式表示:

$$F_P(X)=(f(p_{r,s}, v_{r,s}))_{R \times S} \Delta (f_{r,s})_{R \times S} \quad (4)$$

又令  $u_{r,w}=\min_{\substack{j=1,2,\dots,A \\ x_{j,j} \neq 0}} \{u_{r,w,1} \cdot x_{1,1}, u_{r,w,2} \cdot x_{2,2}, \dots, u_{r,w,A} \cdot x_{A,A}\}$ , 令对风险损失矩阵  $M$  的影响函数为  $h(m_{r,w}, u_{r,w})$ , 取  $h(m_{r,w}, u_{r,w})=\min\{m_{r,w}, u_{r,w}\}$ , 则采用风险缓解策略后的风险损失矩阵可用(5)式表示:

$$H_M(X)=(h(m_{r,w}, u_{r,w}))_{R \times W} \Delta (h_{r,w})_{R \times W} \quad (5)$$

则采取风险缓解策略后整个项目的期望风险损失应由公式(6)表示:

$$ERL(X)=\sum_{s=1}^S \sum_{w=1}^W (F_P(X)' \times H_M(X))=\sum_{s=1}^S \sum_{w=1}^W (\sum_{r=1}^R f_{r,s} \times h_{r,w}) \quad (6)$$

一般而言, 风险缓解策略  $i$  和策略  $j$  的选择存在着两种情况: 一种情况是策略  $i$  和策略  $j$  相互排斥, 如果选择风险缓解策略  $i$  则就不能选择策略  $j$ ; 另一种情况是策略  $i$  和策略  $j$  相互包含, 即如果选择策略  $i$  则就意味着选择了策略  $j$ 。定义  $q_{i,j}=1$ , 表示策略  $i$  和策略  $j$  相互排斥, 定义  $b_{i,j}=1$  表示选择策略  $i$  则就选择了策略  $j$ , 这样就可建立如下成本优化模型:

$$\begin{aligned} \text{Minimize } TEC(X) &= AAC(X) + ERL(X) \\ \text{s. t. } &\begin{cases} x_{i,i} + x_{j,j} \leq 1 & \forall q_{i,j} = 1, i, j \in A \\ x_{i,i} \leq x_{j,j} & \forall b_{i,j} = 1, i, j \in A \\ x_{i,i} \in \{0, 1\} & \forall i \in A \end{cases} \end{aligned} \quad (7)$$

其中,  $A$  为风险缓解策略集。

### 4 模型的应用

下面以国内某通过 CMM3 认证的软件公司开发的某市国税电子报税系统为例, 经过改造说明该模型的应用。

该公司开发的某市国税电子报税系统经过几轮与客户方的谈判, 逐步明确了需求。该系统分为企业报税子系统和个体报税子系统, 企业报税子系统的主要功能模块为初始化、数据维护、数据实时处理、划款控制、权限控制、系统登录、安全控制七部分。个体报税子系统主要分两个阶段的工作: 征期前数据准备, 征期内及征期后处理。其中, 企业报税子系统比较重要, 管理人员对其进行了风险分析, 并制定了风险缓解策略。通过风险分析和优先级排序, 项目管理人员认为项目分解元素中可能存在的 12 项风险事件及其概率矩阵  $P$ , 同时, 经过估算, 给出了每项风险事件发生时可能对项目的各分解元素造成的影响矩阵  $M$ 。其中, 12 项风险事件为: 需求定义不清晰、数据结构定义不准确、模块接口设计不合理、图形界面定义太复杂、重要开发人员流失、人员缺乏经验、开发工具不切实际、没有技术上的理解误差、编码中出现逻辑错误、局部测试性能不良、不必要的性能要求过多、训练水平不能满足要求。风险事件对每项分解元素的影响值用金钱表示。

$P$  和  $M$  矩阵如下, 矩阵中空余的部分数值为 0。其中  $M$  中的数据以 100 元为数量级, 负值表示影响值不是损失值, 而是收益值。



```

    {
        fanhui=nn["userid"]. ToString();
    }
    else
    {
        return "该用户不存在,请重新登录";
    }
}
}
}

```

从上述代码可以看出,组件 logic 负责登录逻辑的处理,比如判断用户是否合法以及合法性判断后相应的处理逻辑。该组件对应图 2 中的 Business-Layer,它通过向低层组件 dbsql 发送各种数据库存储要求,实现业务逻辑的所要求的具体数据库存取操作。

理论上 sqldb 和 logic 两个组件在任何机器上单独编译成.dll 文件即可直接使用,充分实现了代码分离。

### 4.3 表示层的实现

新创建网站时系统会有一个默认的页面文件 default.aspx。在该页面文件中,通过放置一些 textbox 文本框,button 按钮,lable 控件实现登录表单。在后台 default.aspx.cs 中写入如下代码:

```
protected void Button1_Click(object sender, EventArgs e)
```

```

    {
        sqllogic sqllogic xx=new sqllogic. sqllogic();
        Label1. Text=xx. xianshi(this. TextBox1. Text);
    }
}

```

从上述代码可以看出,表示层只是调用组件 logic 提供的方法 xianshi(string xx),通过页面控件传递参数。具体的逻辑处理完全由逻辑层的组件负责。理论上,表示层可以通过各种页面编辑工具制作,逻辑层可以在任何机器上存在,二者互不干扰,表示层的修改不影响逻辑层,反之亦然。这就充分方便了代码的复用以及应用程序的扩展。

**小结** 在 ASP.NET 中实现三层结构还有很多其他的方法,本文只是通过一个例子,对基于组件的开发方法做了一个简要的探讨。Asp. Net 天然的结构特性决定了它在 B/S 结构的软件开发中必将占有重要的地位。

### 参 考 文 献

- 1 Chris ULLman Chris Goode Asp. Net 入门经典. 北京:清华大学出版社,2002
- 2 廖信彦. Asp. NET 技术参考. 北京:中国铁道出版社,2001
- 3 石志国. Asp. Net 程序设计实用教程. 北京:电子工业出版社,2006
- 4 启明工作室. Asp. Net 网络应用系统开发与实例. 2005

(上接第 281 页)

表 1 各风险缓解策略对概率矩阵 P 和影响矩阵 M 的影响情况

风险缓解策略序号	影响类型	影响编号	工作分解元素	风险项序号														
				1	2	3	4	5	6	7	8	9	10	11	12			
1	P	1a	1.2.1	0.60														
	P	1b	1.2.2								1.50							
	P	1c	1.2.3			0.80							0.80					
	P	1d	1.2.2.3				0.70											
	M	1g	1.2.3								20						0.80	
2	P	2a	1.2.1	0.20														
	P	2b	1.2.1.1							0.80								
	P	2c	1.2.2								1.50							
	P	2d	1.2.2.3				0.60											
3	P	3a	1.2.3										0.20					
	P	3b	1.2.4											0.30				
4	P	4a	1.2.2									2.00						
	P	4b	1.2.2.1			0.60												
	P	4c	1.2.2.3		0.70													
	M	4e	1.2.3.2							20								
5	P	5a	1.2.1	0.50														
	P	5b	1.2.2.3				0.90											
6	P	6a	1.2.1	0.60														
	P	6b	1.2.2.1			0.90												
	P	6c	1.2.2.3				0.90											
7	P	7a	1.2.3									1.20						
	P	7b	1.2.4										1.10					
8	P	8a	1.2.3									0.80						
	P	8b	1.2.3.1					0.8										
9	P	9a	1.2.2									1.20						
	P	9b	1.2.2.1			0.90												
10	P	10a	1.2.4											0.80				
	P	10b	1.2.6.1												0.90			
11	P	11a	1.2.3									0.90						
	M	11b	1.2.4									20						
	M	11c	1.2.6.3										25					
	M	11d	1.2.6.1												10			
12	P	12a	1.2.6.2														0.30	
13	M	13a	1.2.6.3															5