小型软件项目的风险管理研究①

Risk Management of Small-Scale Software Projects

谢　铭　(广州市信息化办公室 广东 广州 510030)

摘　要： 风险管理是软件项目管理中一个很关键的环节，但通常的风险管理方法研究大都针对于大中型软件项目。然而，对小型软件项目而言，一方面有着与大项目不同的一些特点和管理方式，另一方面很多方法又是共通的。本文针对小型软件项目人员少，周期短的特点，对广为应用的风险表评估方式和管理流程作了裁剪，通过性质分类简化了评估模型，能在一定程度上修正风险评估中的主观偏差，使其分析方法和管理策略更加适用于小型软件项目。

关键词： 风险 风险管理 风险评估 风险分析 小型软件项目

风险管理一直是软件项目管理的重点和难点问题。一项调查表明，大约 70%的软件开发项目超出了估算的时间，90%以上的软件项目开发费用超出预算。因此，软件开发迫切需要进行项目管理尤其是风险管理。然而，现在无论是书籍还是论文，里面阐述的大多数的项目管理方法都是立足于大中型软件项目的，对小型软件项目管理方面的研究相对较少。目前，我国的软件产业仍然以数量占绝大多数的中小型软件企业为主，因此，对与其企业规模相对应的软件项目也实行项目管理，企业才能更好的生存和发展。本文结合小规模软件项目的特点，对其中的软件风险特征、风险识别及分析、风险控制等项目开发过程中常见的风险管理问题及其方法作了一些研究和阐述。

1　基本定义及相关工作

1.1 风险的基本定义

一般认为，任何风险都包括三个主要方面：①发生的有害事件是什么？②发生的可能性有多大？③发生引起的后果如何？这三者构成了评估风险的基础。据此，Kaplan 和 Garrick[1]认为，风险不是一个数字，也不是一条曲线或是一个向量，而应该是一个三元组的完备集，即

$$R_{risk} = \{<S_i , l_i , X_i>\}c$$

其中，R_{risk} 代表风险，S_i 为第 i 个有害事件，l_i 代表第 i 个有害事件发生的几率(likelihood)，X_i 表示第 i 个事件的结果，是一种损失指标；c 脚标表示这个集合是一个完备集。集合中的元素，即三元组$<S_i , l_i , X_i>$只是风险集的一个元素，整个集合才是全部风险。

1.2 风险管理的相关研究

在风险管理模型方面，Boehm 首先正式提出和详细论述了软件开发中的风险问题，并提出了软件风险管理的方法。Boehm 认为，软件风险管理指的是“试图以一种可行的原则和实践，规范化地控制影响项目成功的风险”，其目的是“辨识、描述和消除风险因素，以免他们威胁软件的成功运作”[2]。

在风险管理步骤上，Boehm 也指出风险评估和风险控制是风险管理的两大部分，风险评估又可分为识别、分析和设置优先级 3 个子步骤，风险控制则包括制定管理计划、解决和监督风险 3 步[3]。Boehm 理论对软件项目风险管理做出了理论的描述和总结，指出了风险管理的组成部分和主要步骤，为风险管理方法的研究奠定了基础。但 Boehm 没有清晰地说明风险管理模型所要捕获的软件风险的具体方面，这意味着对具体的软件项目而言，风险管理步骤需要改进和扩展。

卡耐基梅隆大学的软件工程研究所（SEI）也对风险管理进行了大量的研究，提出了持续风险管理模型 CRM（Continuous Risk Management）。CRM 模型将风险管理分为 5 个步骤：风险识别、分析、计划、跟踪、控制，要求在项目的生命周期的所有阶段都关注风险识别和管理[4]。SEI 提出的是一种不断迭代的

① 收稿时间:2008-07-23

风险管理方法，主要思想是：不断地评估可能会造成恶劣后果的因素，决定最迫切需要处理的风险，实施相应的风险控制策略，评测并监控风险策略实施的有效性。然而，CRM 模型层次还包括开发法和用户方双方的组织结构，主要针对于组织严谨，周期长的大中型项目的风险管理。

风险管理往往也与所对应系统模型和框架密切相关，在这方面，Leavitt 模型将各种系统的组织划分为 4 个组成部分：任务、结构、角色和技术[5]。根据模型的思想，这 4 个组成部分可以和软件开发的各个因素很好地对应起来，而且这些组成部分是密切相关的，一个组成部分的变化会影响其它的组成部分，如果一个组成部分的状态和其他的状态不一致，就会造成比较严重的后果。Leavitt 模型实际上是提出一个框架，可以更加广泛和系统地将软件风险的相关信息组织起来，其设计方法和思想研究已经广泛应用于信息系统中。

以上是目前软件风险研究领域比较经典的 3 种方法，其都从不同角度或者细化层次对风险管理进行了研究[6]。然而，对具体的软件项目而言，不存在一成不变或者一劳永逸的方法，必须根据项目的实际情况和步骤对其进行调整和裁减。

2 基于性质分类的风险分析方法

2.1 小型软件项目的特点分析

小型项目相对于大中型项目而言，有以下的特点：

(1)项目负责人一般也是中小型软件公司的老板，对软件工程有一定的了解，但不全面；

(2)项目功能相对较简单，涉及面相对较狭窄；

(3)项目开发人员较少，人员组织结构简单；

(4)开发周期往往较短，少则两三个月，多则一到两年。

一般来说，在大中型软件项目中，软件开发主要分为六个阶段：需求分析阶段、概要设计阶段、详细设计阶段、编码阶段、测试阶段、安装及维护阶段。软件公司将软件配置管理、软件质量管理、软件风险管理及开发人员管理四方面内容导入软件开发的整个阶段。小型软件项目的开发同样分为六个阶段，但比较模糊，侧重点也不一样；至于软件配置管理、软件质量管理、软件风险管理及开发人员管理四方面内容则比较少。

因此，考虑到小型软件项目资源少，周期短等特点，小型软件项目风险管理的要求可归纳如下：

①需要建立一个尺度，以反映风险发生的可能性以及描述风险的后果；

②风险预测和评估方法简单直观，要求耗费较少的人力资源和管理成本；

③由于风险种类和形式相少，变动也较少，往往只需关注那些易发生后果严重的风险；

④风险管理可以贯穿于项目开发阶段中，不一定需要独立开来，专人专责。

2.2 小型软件项目的风险描述

小项目看起来比较简单，比较容易成功，因而人们往往忽视了小型软件项目的风险管理。合理的管理模式用一句话形容就是“麻雀虽小，五脏俱全”，即使是小规模的项目开发，仍然应该遵循软件开发的一般规律。但是小项目有它自身的一些特点，具体措施和策略实行起来可以相对灵活些。为了很好地识别和消除软件风险，项目管理者在项目开始时需要标识影响软件风险因素的风险驱动因子，这些因素包括性能、成本、支持和进度等。与此同时，也需标识风险对项目的不同的影响程度。

理论上，可以采用四元组<RS, RT, RR, RP>定义项目的风险描述体系：

RS 代表风险的具体事件集合，如果一个可能发生的事件 s_i 会对项目造成不良的影响，则 $s_i \in RS$，其中下标 i 代表事件序号，用以区分不同的风险事件。

RT 表示风险的影响因素类别，不失一般性地，风险影响因素可以按如下的方式简单分类[8]：

性能风险——项目能够满足客户需求及其使用目的的不确定程度。

成本风险——项目成本预算能够被维持的不确定程度。

支持风险——软件易于纠错、适应和增强的不确定程度。

进度风险——项目进度能够保证产品能按时交付的不确定程度。

如果对以上因素类别分别用 t_1,t_2,t_3,t_4 标识，则 $RT=\{t_1,t_2,t_3,t_4\}$。

RR 表示风险所造成影响的程度，一般都分为可忽略的(r_1)，轻微的(r_2)，严重的(r_3)，灾难的(r_4)四种，即 $RR=\{r_1,r_2,r_3,r_4\}$。下面表 1 就给出了一种风险因素及其影响程度的具体定义：

表 1　风险影响类别表

RT/RR	性能（t_1）	支持(t_2)	成本(t_3)	进度(t_4)
灾难的(r_1)	根本无法达到要求的技术性能	无法作出响应或无法提供技术支持要求的软件	严重的资金短缺，非常可能超出预算	无法在交付日期内完成
严重的(r_2)	技术性能有所下降	对软件修改会有少量的延误	资金不足，可能会超支	交付日期可能会被延迟
轻微的(r_3)	导致次要任务的执行效果不好		成本、影响和即可恢复的进度上的小问题	
可忽略的(r_4)	导致使用不方便或不易操作		错误对进度及成本的影响很小	

RP 描述风险发生的可能性（概率）大小，这里假设分为几乎不可能(p_1)，很少可能(p_2)，有可能(p_3)，很大可能(p_4)，非常可能(p_5)五种情况，则有 RP={p_1,p_2,p_3,p_4,p_5}。

在大中型项目的风险管理中，影响程度 RR 和风险概率 RP 集合往往是用一定数值范围内的数值集合所标识，例如影响程度 RR 用项目损失时间 x(周)，风险概率 RP 用数值概率 I(0%~100%)标识[7]等。这很大程度是因为大中型项目开发周期长，风险事件数目多，且存在的变数大，需要在估算中增加风险的区分度。况且对大中型项目而言，一般通过 Delphi 法(评估人员首先独立评估，然后一轮轮讨论直至达成共识)和专家咨询法往往可以减少直观量化的偏差，达到一定的数值量化精度。而在小型规模的项目中，一方面风险的事件数量相对较少，另一方面由于参与风险评估的人员少，取评估均值的方法对数值估计的偏差修正有限，所以这里提出的用估计相应类别代替具体数值的方法有以下好处：

(1)使风险估计的内容显得更加直观和清晰；

(2)能减少小项目中评估的主观偏差程度。

需要指出的是，上述对风险描述的定义并不是恒定不变的，可以根据项目自身的特点对其类别数量及定义进行调整。

2.3 小型软件项目的风险预测

风险预测，又称为风险估算，主要是从两个方面评估每一个风险：风险发生的可能性，以及风险发生了所产生的后果。项目的风险预测方法有分解法，故障树法，流程图法，专家调查法等。以上方法相对小项目来说，或者过程太繁复(分解法，流程图法等)或者需要较大的管理和人力成本(专家调查法等)。因此，为了适应小型软件项目开发周期短，人力和其他管理资源相对不充裕的实际情况，可以采取一种相对较简单的预测方法——风险表法。

风险表给项目管理者提供了一种简单的风险预测技术，有着各式各样的描述格式，然而正如前面所述，通常的风险表中风险概率(probability)和影响程度(impact)的估计都是采取具体的数值表征。于是，在文献[8]定义的风险表格式的基础上，结合上述定义的风险描述体系，可以采取以所属类别代替具体数值的估算方式，重新定义一种风险预测表(样本如表 2*)。其中，项目组一开始要在表中的第一列列出所有风险的可能事件，这些可以利用事先定义的风险检查条目来完成。在第二列对风险的影响因素进行分类，风险的发生概率类别放在第三列，第四列是风险的影响程度，第五列者是针对于该风险的专门描述。

若用 RST 表示风险表，则 RST 实质上是一个四元组集合且 $RST \subseteq RS \times RT \times RP \times RR$。

表 2　风险预测样本表[8]

风险事件(RS)	影响因素(RT)	发生概率(RP)	影响程度(RR)	其他描述
规模估算可能非常低	T_3(成本)	P_3(有可能)	R_2(严重)	
用户数量大大超出计划	T_1(性能)	P_2(很少可能)	R_3(轻微)	
复用程度低于计划	T_2(支持)	P_2(很少可能)	R_2(严重)	
最终用户抵制该计划	T_3(成本)	P_2(很少可能)	R_3(轻微)	
交付期限将被紧缩	T_4(进度)	P_3(有可能)	R_2(严重)	
资金将会流失	T_3(成本)	P_2(很少可能)	R_1(灾难)	
用户将改变需求	T_4(进度)	P_5(非常可能)	R_2(严重)	
技术达不到预期的效果	T_2(支持)	P_2(很少可能)	R_1(灾难)	
缺少对工具的培训	T_2(支持)	P_4(很大可能)	R_3(轻微)	
人员缺乏经验	T_4(进度)	P_1(几乎不可能)	R_2(严重)	
人员流动频繁	T_4(进度)	P_4(很大可能)	R_2(严重)	

一旦完成风险表的内容，就需要根据概率及影响类别等对风险表进行风险评估，然后根据评估结果来进行排序。高概率、高影响的风险放在表的上方。这就完成了一次的风险排序。项目管理者根据排序后的风险表制定相应的风险管理文档，并且可以定义一条终止线(表中的某一行)，只有在那些终止线上的风险才会得到进一步的关注，终止线之下的风险则根据需要可再评估以完成第二次排序。

2.4 小型软件项目的风险评估

风险评估是对已识别的风险要进行估计和评价，主要任务是确定风险发生的概率与后果，确定该风险的经济意义及处理的费用及效率分析。在风险评估过程中，需要进一步审查在风险预测阶段所做的估算的精确度，试图为所发现的风险排出优先次序，并开始考虑如何控制或避免可能发生的风险。

要使评估发生作用，必须定义一个可客观量化的风险水平值， 风险水平值代表处理该风险的重要程度，越需要优先处理的风险越重要，风险水平值也越高。定义函数映射 u：RP-->R^+，v：RT--> R^+，w：RR--> R^+， 这里 R^+为非负实数集，RP,RT,RR 分为前面定义的发生概率、影响因素、影响程度的类别集合。其中 $u(p_i) \mid p_i \in RP$，$v(t_i) \mid t_i \in RT$，$w(r_i) \mid r_i \in RR$ 分别代表着相应性质类别（p_i, t_i, r_i）在项目风险评估中的重要程度，数值越大，所代表性质的重要性越高。

由于 RP，RT，RR 都是离散元素的有限集合，若 $(s_i, p_i, t_i, r_i) \in RST$，则 $u(p_i)$，$v(t_i)$，$w(r_i)$实质上就等价于有关风险事件 s_i 的不同性质的重要性权值。则事件 s_i 的风险水平值可定义为 $f(s_i)=g(u(p_i),v(t_i),w(r_i))$，实数域函数 g 反映了 RP,RT,RR 的权值组合关系。为求简单计，通常情况下可令 $g(x,y,z)=xyz$，即

$$f(s_i)=u(p_i)*v(t_i)*w(r_i)$$

一个不同的函数组{u,v,w,g}就描述了一种不同的风险评估的方式，可以根据实际项目的具体情况定义不同的函数 u,v,w 甚至 g。例如，若项目需要优先确保进度，则可将权值 $v(t_4)$提高，若需要优先确保对成本控制，同样可将 $v(t_3)$提高…这样可以根据不同的具体要求在项目不同阶段实行相应的风险处理策略。

同样，可以用 $s=\sum f(s_i)$，$s_i \in RS$ 表示整个项目的风险水平，有需要的话还可定义一个风险水平阈值 s_m，当整体风险水平超过它既 $s>s_m$ 时就可考虑中止项目。但实际上，风险水平很少能表示成光滑曲线。在大多数情况下，它只是一个数值区域，其中存在很多不确定性。另外，函数 u,v,w 甚至 g 的设定都有一定的经验性和针对性要求，否则难以准确反映项目的实际风险情况。但对小型项目而言，由于风险事件及其组合情况还不是很复杂，这种量化的风险评估对具有一定经验的系统分析员来说，还是可以把握的。

3 小型软件项目的风险管理策略

3.1 项目的风险管理文档

所有的风险分析活动都只有一个目的，那就是辅助项目组建立处理风险的策略和流程。一个有效的策略必须考虑三个问题： 风险避免，风险监控，风险管理及意外事件计划，这些都需要在风险管理文档中阐述清楚。同时，小型项目的风险管理可以包含在软件项目计划中，将其风险分析结果及其应对措施文档化，并由项目管理者作为整个项目计划中的一部分来使用。

总而言之，项目风险管理文档应当包含风险的描述、预测及其评估结果，对风险的预防或缓解措施，以及措施实施和效果的跟踪记录等，是项目风险管理活动的主要依据。这对通常缺乏软件质量保证人员的小型软件项目来说，风险管理文档的制定和维护显得更为重要，下面就给出了一种风险管理文档的大纲(参考 RIMM 计划[8])：

一、引言
- (1)文档的范围和目的
- (2)主要风险综述
- (3)风险的责任承担
  - a.管理者
  - b.技术人员

二、项目风险表
- (1)风险事件及其影响因素
- (2)风险的预测及其评估结果
- (3)终止线之上所有风险的分析

三、风险的缓解、监控和管理
- (1)缓解措施
  - a.一般策略

b.缓解风险的特定步骤

(2)监控措施

a.被监控的因素

b.监控办法

(3)管理措施

a.意外事件计划

b.特殊的考虑

四、风险监控的迭代时间安排表

五、总结

3.2 项目的风险管理流程

可以这么说，项目风险管理文档是项目风险管理策略的核心，一切的风险管理流程将围绕着文档进行。与此同时，对小型软件项目而言，由于风险变动情况不大，往往主要关注那些易发生，后果严重的高危，其所对应的风险管理流程也可以相对简单些。图 1 就根据前面所述的风险分析方法，结合 Boehm 的十大风险列表思想[2]和 CRM 模型对风险管理步骤的划分[4]，列出了一种适用于小型软件项目的，可迭代的风险管理流程：

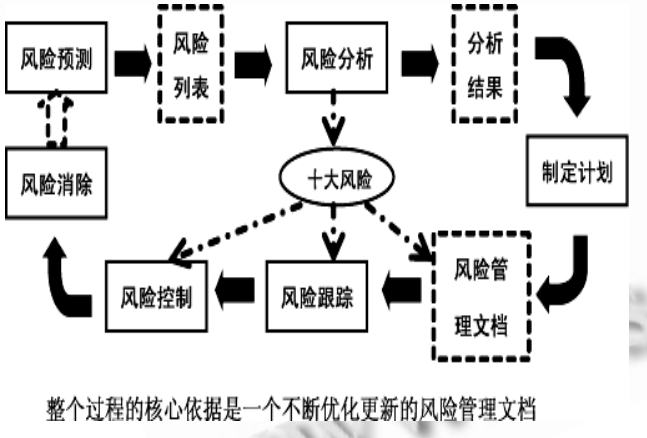

图 1　一种小型软件项目的风险管理流程

4　总结

风险管理是一种特殊的规划方式，被认为是减少项目失败可能性的一种重要手段，虽然目前风险管理的实际应用还不够广泛或不够有效，但这并不意味着风险管理可以被忽略。因此本文借鉴现有的研究成果，将风险管理模型与目前较广泛的小型软件项目的特点进行融合，针对风险表在小项目的风险预测中数值主观偏差较大的问题，引入了性质分类的预测及评估方法，并在此基础上探讨了小型软件项目的管理策略。我们下一步的工作是继续研究风险评估模型中的权值设定等函数定义，优化、实现并测试这个风险管理策略。

参考文献

1 Kaplan S, Garrick BJ. On the Quantitative Definition of Risk, Risk Analysis,1981,1(1):11－27.

2 Boehm B. Software Risk Management: Principles and Practice. IEEE Software, 1991.

3 Boehm BW. Software Engineering Economics. Englewood Cliffs, NJ: Prentice Hall,1981.

4 Chittister,Clyde, Kirkpartick R, et al. Risk Management in Practic. SEI Technical Review,1993.

5 Lyytinen K, Mathiassen L, et al. Attention Shaping and Software Risk: A Categorial Analysis of Four Classical Approaches. Info System Research, 1998.

6 张珞玲,李师贤.软件项目风险管理方法比较和研究.计算机工程, 2003,29(3):91－94.

7 曹汉平,贾素玲,王强,等.软件项目风险管理.中国项目管理网:http://www.opentest.net/quality/itemrisk.htm. 2002.6.

8 Roger SP. 软件工程--实践着的研究方法.北京.机械工业出版社.1999:132－150.