

软件项目风险管理研究综述

牟玲玲¹, 陈立文², 张建军²

MU Ling-ling¹, CHEN Li-wen², ZHANG Jian-jun²

1.天津大学 管理学院, 天津 300072

2.河北工业大学 管理学院, 天津 300130

1.School of Management, Tianjin University, Tianjin 300072, China

2.School of Management, Hebei University of Technology, Tianjin 300130, China

MU Ling-ling, CHEN Li-wen, ZHANG Jian-jun. Survey on software project risk management. Computer Engineering and Applications, 2007, 43(10): 202-205.

Abstract: As one of the newly management theory risk management of software project has gained much attention. This paper first reviews the development process of software industry and software project risk management. Then the software project risk basic conception as well as the important software project risk management theory and the method are introduced. Through summarizing the relative research results obtained in the recent 30 years, we summarize three important research tendency of software project risk management. Based on which some future research directions are proposed.

Key words: project management; risk; software project; risk management

摘 要: 近年来, 软件项目风险管理作为新兴的管理理论, 得到了国内外大批学者的青睐, 并取得了一定的研究成果。首先回顾了软件行业及软件项目风险管理的发展历程, 对软件项目风险的基本概念以及重要的软件项目风险管理理论和方法进行了介绍。然后综述了国内外学者近 30 年来在该领域取得的研究成果, 总结出软件项目风险管理的三大研究趋势。并在此基础上提出了未来重点研究方向。

关键词: 项目管理; 风险; 软件项目; 风险管理

文章编号: 1002-8331(2007)10-0202-04 文献标识码: A 中图分类号: F407.6

风险是软件项目开发过程中所固有的、不以人的意志为转移的属性。近年来, 由于缺乏有效的软件风险管理, 导致很多软件开发项目都无法在预定的时间和预算下交付高质量、高性能的软件产品, 软件项目常常以失败告终。因而软件项目风险管理开始受到越来越多的学者地关注并发展成为一门新兴的管理学科, 其中的许多方法和模型已经在实践中得到了应用并取得了良好的效果。本文试图从软件项目风险开始兴起的 20 世纪 80 年代开始对近 30 年的相关理论研究进行简要的综述与评价, 以期总结研究成果, 明确未来研究方向。

1 软件行业及软件项目风险管理发展历程

在 19 世纪 50 年代, 软件伴随着第一台电子计算机的问世诞生了。早期的软件开发并无系统的方法可以遵循, 软件设计是在某个人的头脑中完成的一个隐藏的过程^[1]。大多数软件是由使用该软件的个人或机构研制的, 软件往往带有强烈的个人色彩。而且, 除了源代码往往没有软件说明书等文档。这个时期软件开发基本处于起步阶段, 关于软件项目风险的管理研究工作尚未出现。

20 世纪 60 年代中期到 70 年代中期软件开始作为一种产品被广泛使用。软件的数量急剧膨胀, 软件需求日趋复杂, 维护的难度越来越大, 开发成本令人吃惊地高, 而失败的软件开发项目却屡见不鲜。‘软件危机’就从这时出现了。为了克服这一危机, 在 1968、1969 年连续召开的两次著名的 NATO 会议上提出了软件工程这一术语, 并在以后不断发展、完善。

到了 20 世纪 80 年代, 软件已经历了程序设计、程序系统阶段, 开始向软件工程阶段发展。这一时期的开发技术和手段更加先进, 对管理水平也提出了较高的要求。并出现了软件生存周期(Life Cycle)的概念, 它包括计划、需求分析、设计、程序编写、测试和运行维护 6 个步骤。由于软件危机而引发出来的一些对软件负面的结论有: 软件开发问题是经济增长的主要障碍; 软件失效正在造成几十亿、几百亿的经济损失; 用软件工程技术来控制费用及软件质量也无法尽善尽美。因此就存在软件风险问题。1989 年由 IEEE 出版 Barry W. Boehm 编写的《Software Risk Management》(软件风险管理)是第一本比较深入地专门探讨风险管理的书籍。随后, 越来越多的机构开始研究软件项目风险管理。其中美国卡内基-梅隆大学的软件工程研究 SEI

基金项目: 河北省自然科学基金(the Natural Science Foundation of Hebei Province of China under Grant No.G2005000085); 河北省建设厅研究计划资助项目(No.2004-120)。

作者简介: 牟玲玲(1979-), 女, 博士研究生, 研究方向: 管理科学与工程; 陈立文(1964-), 男, 博士, 教授, 博导, 研究方向: 项目管理与风险控制; 张建军(1972-), 男, 博士研究生, 研究方向: 项目管理与风险控制。

(Software Engineering Institution) 提出了风险管理模型 CRM (Continuous Risk Management), 还多次举行研究软件风险管理的年会, 一些院校也将风险管理纳入软件项目管理中进行讲授。

20 世纪 90 年代至今以网络计算为特征的信息高速公路得到了很快地发展。基于构件的开发方法使得软件开发的规模进一步扩大。软件不仅在规模上快速地发展扩大而且其复杂性也急剧增加, 开发的成本和进度变得更加难以预料。软件开发成功与否已不再仅是技术上的问题。软件项目管理的地位突显出来。在此基础上, 软件项目风险管理研究开始慢慢丰富起来, 理论上对风险进行了一些分类, 提出了风险管理的思路; 实践上也出现了一些定量管理风险的方法和软件工具。是否具有专门的风险管理已经成为判断一个软件开发组织成熟与否的主要指标之一。ISO 提出的 SPICE (Software Process Improvement Capability Determination) 将风险管理作为评估的一部分, 而 SEI 在提出的软件成熟度模型 SW CMM 第 2 版中, 风险管理已经被添加成为第 3 级中集成化软件管理的关键过程领域 (KPA)。国内学者对软件项目风险的研究也日趋丰富起来, 在软件项目管理相关书籍中开始将风险管理作为单独一章, 进行探讨。

2 软件项目风险定义概述

多数风险决策所面临的共同问题是缺乏必要的信息。对于一个充满风险的项目进行管理, 但缺乏必要的手段去检测和描述那些不确定性, 往往会导致项目的失败。根据不同的观点, 不确定性也可以有不同的解释 (见表 1)。

表 1 基于不同角度的不确定性的定义

观点	不确定性的种类	定义
管理者的角度 ^[4] (Gemmer1997)	能否按进度进行	风险事件何时发生, 可以解决到何种程度
	能否有效的控制	不权威且缺乏影响力的决策, 不一致的指挥
	是否有充足的信息	不充足, 欠缺准确的决策信息
分析员的角度 ^[5] (Charette 1990)	是否能结构化	缺乏明确定义系统变量的信息
	如何衡量	缺乏给描述系统变量赋值的信息
	项目结果如何	缺乏项目各种结果发生可能性信息
开发者的角度 ^[6] (Friend and Hickling 1987)	价值观是否相同	没有明确的目标
	工作环境是否确定	没有针对特定环境的准确的决策信息
	意见是否一致	缺乏合作

不确定性导致了风险和潜在损失的发生。风险通常定量的表述为风险发生的概率与其可能引起的损失的乘积。Boehm 称之为风险的表露 (risk exposure)。Charette 认为潜在的损失应该通过一定方法进行推测, 由此提出了分析风险的三要素: 风险事件 S_i (what can go wrong)、风险概率 L_i (likelihood)、风险值 V_i (value):

$$\text{Risk} = \{(S_i, L_i, V_i) | i=1, 2, \dots, n\}$$

这种定义的缺点是将低概率高成本的情况与高概率低成本的情况等同起来。为了避免这种局限性, Kumamoto 和 Henley (1996) 在此基础上增加了一个效用值 (Q_i), 用它来描述结果的严重性, 可用四维标准对其进行衡量: 产品功能 (F_i), 产品缺陷 (D_i), 产品成本 (C_i), 产品使用期限 (T_i)。由此分析风险的三要素变为四要素。

$$\text{Risk} = \{(S_i, L_i, V_i, Q_i) | i=1, 2, \dots, n\}$$

3 软件项目风险管理理论综述

软件项目风险管理开始于 20 世纪 70 年代对软件缺陷预测的研究, 该时期的研究集中于对软件质量的某个属性进行预测的方法。20 世纪 80 年代以来, 在软件项目风险管理研究上出现了很多成果, 本文着重介绍目前有代表性的 8 个软件项目风险管理模型并给出简单评价。

3.1 Barry Boehm 的模型

Boehm 用公式对风险进行定义, 其中 $RE=P(UO) \times L(UO)$ 表示风险或者风险所造成的影响, $P(UO)$ 表示令人不满意的结果所发生的概率, 表示糟糕的结果会产生的破坏 $L(UO)$ 性的程度^[9]。在风险管理步骤上, Boehm 基本沿袭了传统的项目风险管理理论, 指出风险管理由风险评估和风险控制两大部分组成, 风险评估又可分为识别、分析、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。

Boehm 思想的核心是 10 大风险因素列表。虽然其理论存在一些不足, 但 Boehm 毕竟是软件项目风险管理的开创者。在其之后, 更多的组织和个人开始了对风险管理的研究, 软件项目风险管理的重要性日益得到认同。

3.2 SEI 的 CRM 模型

SEI (Software Engineering Institution) 风险管理原则是^[10]: 不断地评估可能造成恶劣后果的因素; 决定最迫切需要处理的风险; 实现控制风险的策略; 评测并确保风险策略实施的有效性。它将风险管理划分为 5 个步骤: 风险识别、分析、计划、跟踪、控制并强调了这是一个在项目开发过程中反复持续进行的活动序列。

SEI 和 Boehm 的模型都以风险管理的过程为主体, 研究每个步骤所需的参考信息及其操作。

3.3 基于 Leavitt 模型的风险管理

1964 年提出的 Leavitt 模型将形成各种系统的组织划分为 4 个有趣的组成部分: 任务、结构、角色和技术^[7]。Leavitt 模型的关键思路是: 模型的各个组成部分是密切相关的, 一个组成部分的变化会影响其他的组成部分, 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, 并可能降低整个系统的性能。

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。Leavitt 理论的设计方法和实现研究已经广泛应用于信息系统中, 它所考虑的都是软件风险管理中十分重要的环节, 而且简单、定义良好、适用于分析风险管理步骤。

3.4 Riskit 方法

Riskit 是由 Maryland 大学提出的, 旨在对风险的起因、触发事件及其影响等进行完整的体现和管理, 并使用合理的步骤评估风险。该方法使用图形形式化的方法支持在定量分析前进行风险情景的定性分析, 其评估方法可以基于历史数据或者对当前项目的预测^[9]。

Riskit 方法的核心部分是用来描述风险的图形形式化工具分析图 Riskit (Riskit analysis graph), 该分析图可以显式地定义风险的不同特性, 比常规的口头论述要更为形式化。Riskit 提供了多方面多层次的风险管理过程体系。

3.5 JIT 模型

Dale Karolak 提出的 SERIM Model (Software Engineering Risk, 软件工程风险模型) 是 IEEE 推荐的方法^[9]。这种方法的特

点在于从多个角度对软件开发中的风险因素进行了评估,并可在开发周期的任何时候根据评估结果监控风险。SERIM从技术和商业两个角度对软件风险管理进行剖析,考虑的问题涉及开销、进度、技术性能等。JIT软件的3个关键思想是:进行计划、在早期识别风险以及并行开发。由于模型提供了一些指标和模型来估量和预测风险,而且数据来源于大量的实际经验,因此具有很强的说服力。

3.6 EXPERT COCOMO 模型

EXPERT COCOMO是南加州大学 Raymond J.Madachy博士提出的一种运用成本估算因子进行风险评估的模型^[10]。它是对COCOMOII的一个扩展,通过标识、分类、数量化和优先级化项目风险来进行项目计划。它可以检测出输入的异常,以及提供超出COCOMOII估计的成本和进度的风险控制建议。

3.7 Soft Risk 风险管理模型

Soft Risk模型是由Keshlaf和Hashim提出的^[11],它基于这样一种观念:记录并将注意力集中在高可能性和高破坏性的风险上是进行风险管理的有效途径。这样可以节省软件开发过程中的时间成本和人力成本,并可以有效地减轻风险的破坏性。此模型确保在软件项目进行中持续地进行风险管理。

3.8 MSF(Microsoft Solutions Framework) 风险管理模型

MSF的风险管理认为:风险管理必须是主动的,它是正式的系统的过程,风险应被持续评估、监控、管理,直到被解决问题或被处理^[12]。该模型最大的特点是将学习活动融入风险管理,强调了学习以前项目经验的重要性。

4 软件项目风险管理研究的三大趋势

Covell和Mumpower曾指出“现代的风险分析方法主要分为两大类,一是运用数学中的概率论方法,再者是用科学的方法识别出各种不同的风险活动与其带来的不利影响之间的关系”。概率论已经有200年的发展历史,因果分析的方法也已经存在相当长的时间。然而,直至19世纪80年代,它们才作为风险分析的理论基础运用到软件工程中,并形成了两种研究趋势。第三种趋势是19世纪80年代中期出现的以实践经验或实际软件开发项目为基础的,用来解决软件开发风险的方法。

按这三种研究趋势可将软件工程中关于风险的文献进行归纳,并大致分为三类:

4.1 第一种研究趋势

这种趋势借鉴并应用了其他领域的方法技术来解决软件开发中的风险问题。在相关领域的方法技术中,应用于软件风险管理的主要是概率论,并且均集中在分析单个风险因素上。例如人工量或代码行的估算。相关文献整理如下:

Boehm提出用统计决策理论来决定在软件项目上进行风险投资的数量问题^[9]。Edgar提出运用概率统计的方法估算由项目风险导致的预算增加的数量^[14]。Garvey和Powell应用三种数学技术(变量的随机模拟、泰勒系列定理、蒙特卡罗随机模拟)来估算工作量及项目规模。Sherer提出用决策树和贝叶斯理论来估算软件失败的可能性^[15]。国内一些学者也纷纷提出相应的风险评估方法,如基于面向对象贝叶斯网络的风险评估方法。该方法通过建立面向对象贝叶斯网络拓扑结构,可定量估计风险的发生概率;再如结合模糊理论、风险权重和专家信任度等概念,对软件项目中的风险进行评估;还有,通过引入模糊

Petir网的有关概念,重点论述应用Petir网对企业信息系统开发的各种风险进行评估的算法以及该算法在信息系统开发风险管理中的应用^[16,17]。

这种研究趋势主要借鉴了统计决策、投资组合、概率统计、失效树、贝叶斯理论、模糊理论等理论来建立模型,从而对软件项目风险进行分析和度量。但是这种研究趋势没有对风险因素与软件项目最终结果的关系进行充分的识别,而仅仅是关注项目进程中的个别风险因素,所以缺乏项目的整体性与系统性。

4.2 第二种研究趋势

这种趋势主要是指开发与设计适合于软件开发过程的风险管理方法。随着现代科学技术的发展,软件开发方法也在得到不断地改进。Boehm在1986年提出的螺旋开发模型(spiral model)可以吸收来自其他开发模型的因素,这也是第一个明确的将风险管理考虑到软件开发中的模型。螺旋开发模型是第二种研究趋势较具有代表性的成果。在该方面做出主要贡献的有:

Charette, Robert N.提出了包含风险识别、估算、评价、计划、控制和监控的软件项目风险管理6步骤^[3]。Boehm提出了风险识别、分析、优先级排序、计划、解决、监控的软件项目风险管理六大过程^[9]。随后Wolff对Boehm在1986年提出的螺旋开发模型的应用与改进。Charette则讨论了风险分析的步骤与策略^[3]。Van Scoy描述了SEI提出的软件项目风险管理6大过程^[18]。Siti等描述了软件项目的范围并在此基础上提出了软件项目风险管理的方法^[19]。Gemmer进行了软件项目风险管理的行为研究^[4]。国内学者在此方面也做出了应有的贡献:例如通过用户需求把度量技术、软件体系结构、软件过程等相关技术融合在一个风险管理框架中,从而保证了风险管理的效果;将项目管理风险理论技术和方法应用于软件开发项目中,有助于管理者规避风险实现高效益。张珞玲,李师贤^[20]将Riskit和Serim方法融合在一起,结合项目实际特点进行了一些修改和调整来进行软件项目风险管理。

这种研究趋势注重在软件工程中进行风险识别及风险管理方法研究。并以建立软件项目风险管理规范为目的。这些基于问卷调查的方法使得风险管理者有了可以对项目中所存在的风险进行程度描述的工具。反过来,这又促进了风险工作的顺利进行。这些模型方法主要是通过观察方法来识别对项目结果产生影响的因素,缺少可靠的以实际经验为依托的定量化手段去确定这种因果关系。

4.3 第三种研究趋势

这种研究趋势主要以实践活动为基础,通过对实际软件开发项目中的风险因素进行识别和评价极大地促进了软件项目风险管理的发展。该领域的成果可归纳如下:

Barki以实际经验为依据,列出了有效的及无效的项目风险管理方法,识别了软件项目中的一系列风险及变量^[21]。Fairley应用COCOMO模型中的成本估算因子及蒙特卡罗随机模拟方法进行软件项目的风险识别与评价,他将影响进度与成本的风险因素作为软件成本估算的因子,将其应用到成本估算模型中,以衡量这些风险因子对项目进度与成本的影响程度^[22]。Jones制定了软件项目风险管理手册,对项目中出现的风险的原因及解决措施进行了详细的说明,并认为风险是优化软件项目成果的有效手段^[23]。Moynihan依据个人的经验,对相关文献中提出的关键风险因素进行了识别,向众多项目管理者征询项目风险因素,总结了那些失败的软件开发项目所具有的共性,

并在大型控制系统的项目中加以验证^[24]。Mark Keil 等对 top10 进行了重新的调查, 调查对象为芬兰、香港、美国的软件项目经理, 得出新的十大风险因素, 由此对软件项目风险进行了重新分类^[25]。

总之, 这种趋势是研究较多并且非常活跃的领域。它提倡应用科学的方法观察、统计分析、识别风险因素。一些研究者综合运用科学的观察方法, 辅以必要的概率论知识进行软件项目的风险分析, 取得了一定的成果。如 Fairley 应用蒙特卡罗随机模拟来识别与评价风险因子。然而国内在此方面基本没有开展相关的研究和实践活动。

5 结论与展望

通过对相关文献的整理, 可得如下结论:

(1) 从时间上来看国内研究起步较晚, 主要集中在 1998 年以后, 并且还局限于借鉴相关领域的方法技术上。属于第一、二趋势的较多, 对于第三大主流趋势的研究则相对较少, 这也是今后研究的一个重点方向。

(2) 在第三大研究趋势中, COCOMO 模型被不断的应用, 而它本身则是一个软件成本估算模型, 由此可见软件成本估算与软件项目风险管理有着某些必然的联系, 值得展开深入地研究。

(3) 软件项目风险管理发展至今尚未有成熟的理论模型应用于实际, 其主要原因。一是软件开发行业作为一个新兴的行业其发展历史较短, 人们对此所积累的经验比较少, 实行风险管理的观念比较淡薄。二是对软件项目风险管理的研究刚刚起步, 还不能做到理论指导实践。三是即使许多学者已提出各种各样的方法用于软件项目风险管理的不同阶段, 但这些方法之间不具备系统性。很难想象随着风险管理的进行, 管理者要不断的变换各种方法进行风险管理, 同时还要考虑到这些方法的适用范围问题。能否找到一种从风险识别到风险分析再到风险控制都能适用的模型来代替那些纷繁复杂的方法, 以简化风险管理者的工作过程, 这将是今后研究的一个重点。

(4) 虽然缺少成熟理论但是从以上文献中不难发现, 对项目风险的识别与评价已成为进行软件项目风险管理不可或缺的两大步骤。这些文献中研究如何识别风险的较多, 且可采用的方法也相对较多, 但是如何评价和量化风险因素则研究的相对较少, 这也是由软件项目风险难以量化的特性决定的。但为了进行有效的软件项目风险管理, 量化工作必不可少, 如何进行有效地量化也将是这一领域较有价值的研究方向。

(收稿日期: 2006 年 11 月)

参考文献:

- [1] 郑人杰. 实用软件工程[M]. 北京: 清华大学出版社, 1991, 2: 1-20.
- [2] Gemmer A. Risk management: moving beyond process[J]. IEEE Computer, 1997, 30(5): 33-43.
- [3] Charette, Robert N. Applications strategies for risk analysis[M]. New York: McGraw-Hill, 1990.
- [4] Friend J K, Hickling A. Planning under pressure: the strategic choice approach[M]. Oxford: Butterworth-Heinemann, 1987.
- [5] Boehm B. Tutorial: software risk management[M]. Los Alamitos, California: IEEE Computer Society Press, 1989.
- [6] Clyde C, Haimes Y Y. Risk associated with software development: a holistic framework for assessment and management[J]. IEEE Transactions on Software Engineering, 1993, 23(3May/June): 710-723.
- [7] Lyytinen K, Mathiassen L, Ropponen J. Attention shaping and software risk: a categorical analysis of four classical approaches[J]. Information Systems Research, 1998, 9(3): 233-255.
- [8] Kontion J, Basili V R. Risk knowledge capture in the riskIt[C]// Proceedings of the 21st Annual Software Engineering Workshop, 1996.
- [9] Karoak D W. Software engineering risk management[M]. Los Alamitos, California: IEEE Computer Society Press, 1996.
- [10] Madachy R J. Heuristic risk assessment using cost factors[J]. IEEE Software, 1997, 14(3): 51-59.
- [11] Keshlaf A A, Hashim K A. A model and prototype tool to manage software risks[C]// Proceedings of the First Asia-Pacific Conference on Quality Software, IEEE, 2000: 297-305.
- [12] Microsoft Corp. MSF risk management process[EB/OL]. (1999-06). <http://www.microsoft.com>.
- [13] Edgar J D. Controlling Murphy: how to budget for project risk[J]. Concepts, 1982(Summer): 60-73.
- [14] Sherer S A. Software failure risk: measurement and management[M]. New York: Plenum Press, 1992.
- [15] 蒋国萍, 陈英武. 基于面向对象贝叶斯网络的软件项目风险评估[J]. 系统工程与电子技术, 2005, 27(2): 353-356.
- [16] 李美华, 付宏. 软件项目风险评估模型的建立[J]. 吉林大学学报: 信息科学版, 2005, 23(6): 696-701.
- [17] Boehm B. A spiral model of software development and enhancement[J]. ACM SIGSOFT Software Engineering Notes, 1986, 11(4): 22-42.
- [18] Scoy V, Roger L. Software development risk: opportunity, not problem[M]. [SI.]: Software Engineering Institute Carnegie Mellon University, 1992.
- [19] Sisti F J, Sujoe J. Software risk evaluation method, Version 1.0, CMU/SEI-94-TR-19, ADA2 90697[R]. Pittsburgh, Pa: Software Engineering Institute, Carnegie Mellon University, 1994.
- [20] 张珞玲, 李师贤. 软件项目风险管理方法比较和研究[J]. 计算机工程, 2003(3): 91-94.
- [21] Henri B, Rivard S, Talbot J. Toward an assessment of software development risk[J]. Journal of Management Information Systems, 1993, 10(2): 203-225.
- [22] Fairley R. Risk management for software project[J]. IEEE Software, 1994, 11(3): 57-67.
- [23] Capers J. Patterns of software system failure and success[M]. Boston: International Thomson Computer Press, 1996.
- [24] Moynihan T. How experienced project managers assess risk[J]. IEEE Software, 1997, 14(3): 35-41.
- [25] Keil M A. A framework for identifying software project risks[J]. Communications of the ACM, 1998, 41(11): 76-83.