

分类号_____

UDC_____

学 号_____

重庆大学

硕士学位论文

论文题目：软件风险管理方法量化研究

论文作者：涂争光

指导教师姓名、
职称、工作单位：曾 一 副教授 重庆大学

申请学位级别：硕士 专业名称：计算机应用技术

论文提交日期：2005 年 4 月 10 日 答辩日期：2005 年 5 月 26 日

学位授予单位：重庆大学 授位日期：2005 年 月 日

答辩委员会主席：朱宗元 教授

论文评阅人：陈 庄 教授 郭 平 副教授

2005 年 5 月 26 日

重庆大学硕士学位论文

软件风险管理方法量化研究



硕士研究生：涂争光

指导教师：曾一 副教授

学科专业：计算机应用技术

重庆大学计算机学院

二〇〇五年五月

Master Degree Dissertation of Chongqing University

**Study on Measurement Methods for
Software Risk Management**



Master Degree Candidate: Tu Zhengguang

Supervisor: Associate Prof. Zeng Yi

Major: Computer Applications Technology

College of computer Science

Chongqing University

May 2005

摘 要

近年来,随着软件理论和技术不断发展,软件风险管理已成为软件工程领域内保证软件质量的必不可少的关键过程之一。如何对影响软件开发风险的因素进行分析和设计软件风险管理分析数据库是进行软件风险管理的主要课题之一。这项工作能够有效地帮助软件开发人员防范、应对和规避软件风险,同时对提高软件产业的投资回报率和成功率具有一定的理论意义和工程参考价值。

软件风险管理就是为解决影响软件项目、过程或产品的风险而制定的准则。逃避风险永远无法获胜。除非在软件的开发过程中,一直处于无风险的状态。只有充分地理解和学习软件风险管理的知识和技能,同时在实践中不断地积累经验才能有效地进行风险防范和控制,达到减少风险的影响程度和实现利益最大化追求的目的。

本文从分析国内外软件风险管理的发展现状入手,系统地介绍了目前国内外现行软件风险分析经典的模型,详细地分析了影响软件开发成功的各种因素,并按照软件生命周期将这些影响因素按照类别、属性进行分类、归纳整理,详细划分了软件风险因素,这样便于软件开发人员能够有针对性地识别、监控风险,也为建立软件风险量化数据库模型打下了坚实的基础;同时介绍了软件风险定性和定量分析方法,分析、比较了定性和定量分析方法的优缺点,在文中着重介绍了软件风险定量分析方法的内容。

本文再结合软件风险定性、定量分析方法的角度,探讨了软件风险定性、定量分析方法的策略,同时结合软件开发过程中风险动态分布变化的特点,借鉴风险决策树方法,准确量化在软件开发过程中的风险因素,从而能够时刻关注、跟踪、监控软件风险,并且通过具体化该模型各功能和模块,能将该模型应用到软件风险管理之中。

本文所有这些工作的作用和目的是:1)较为详尽、系统地介绍了软件风险管理的理论基础,有助于软件开发人员对软件风险管理能有一个全面的认识;2)本文详细地分析了影响软件开发的风险因素(项),并对其进行分类,组织成系统的层次结构,使软件开发人员能够有效地识别和规避风险;3)本文着重介绍了软件风险定量、定性分析方法,并比较其优缺点;4)结合风险决策树,构建了综合风险估测模型,使用了面向对象思想实现了一个软件风险分析模型。

关键词: 软件风险, 软件开发, 风险分析, 风险数据库, 数据分析

ABSTRACT

In recent years, along with the software theories with the technique of continuously develop, software risk the management have become the software engineering realm the inside to guarantee one of the essential to have's key process of the software masses. How factor to develop risk to influence software proceed analysis with design the software risk management to analyze database is one of the main lessons that proceeds software risk's manage. Investment that this work can availably help the software to develop the personnel to against, reply with evade the software risk, at the same time to exaltation software the industry repay the rate to have the certain theories meaning with successful rate with the engineering worth a.

Software risk the management is for solution to affect the software the risk of the item, process or product but the establishment's criterion. Evade the risk the forever can't win. Unless in the software's development process, always be placed in to have no the status of the risk. Purpose for influence degree for knowledge for having adequately comprehending with studying the software risk managing with technical ability, at the same time in the fulfillment talent of steadily backlog experience availably proceeding risk againsting with control, attaining the decrease risk with realizing benefits maximizing pursuing.

Development present condition that this text is from the analysis the domestic and international software risk manage to commence, and systematically introduced now the domestic and international and current software risk analyze the classic's model, and in detail analyzed the every kind of factor that succeed of development of influence software, and affect the factor these to proceeds according to the category, attribute the sort to induce the sorting, and subdivide the software risk the item, like this the development personnel of convenient for software can have to aim at the sex earth to identify, supervise and control the risk, and also for create software risk quantize database the model lay down solid foundation; Introduced the software risk to settle the sex with the quantitative analysis method, analysis at the same time, than settle the sex with the 优 defect of the quantitative analysis method, and put great emphasis oned to introduce the software risk quantitative analysis of the method's contentsed in the text.

This text join together the software risk agains to settle the sex, quantitative analysis the method's angle, and study the software risk to settle the sex, quantitative

analysis the method's strategy, and join together the software to develop the process the inside the risk development to distribute the variety's characteristics at the same time, and draw lessons from the risk decision tree the method, and bring up the software risk to synthesize the development to take the gauge of to predict the model, can time concern, follow, supervise and control the software risk, accurate quantize to develop the risk in the process the factor in the software, and pass to embody each function of that model with mold piece, can apply to the software risk that model to manage in,

This text the function of all these works is with the purpose is:1) than for detailed, systematically introduced the software risk the theories foundation that managed, and is beneficial to the software development the personnel to cognition software risk management can had first completely;2) this text in detail analyzed the influence software the risk factor(item) that development , and combine as to it's proceed sort, organize the system's level construction, make the software development personnel can availably identify with evade the risk;3) this text put great emphasis to introduce the software risk the fixed amount, qualitative analysed the methods , and compare its advantage is with the defect;4) wedge bonding the risk decision tree, set upped to synthesize the risk valuation the estimate model, and use to face to the object thought to realizes a software risk the analysis of model.

Keywords: Software risk, Software development, Risk analysis, Risk database, Data analysis

目 录

中文摘要	I
英文摘要	II
1 绪论	1
1.1 论文背景	1
1.1.1 国外软件风险管理的发展现状	1
1.1.2 国内软件风险管理的发展现状	2
1.2 本文的目的	3
1.3 本文的主要工作	3
1.4 本文的主要结构	3
1.5 本章小结	4
2 软件风险管理的理论基础	5
2.1 风险管理概述	5
2.1.1 软件风险的分类	5
2.1.2 软件风险的特征	6
2.2 软件风险管理经典模型	6
2.2.1 Barry Boehm 的模型	7
2.2.2 SEI 的 CRM 模型	7
2.2.3 基于 Leavitt 模型的风险管理	9
2.2.4 Riskit 风险管理过程	10
2.3 本章小结	12
3 软件风险因素的分析	13
3.1 影响软件风险的因素	13
3.1.1 项目来源阶段的风险因素	14
3.1.2 项目阶段的风险因素	15
3.1.3 项目实施阶段的风险因素	17
3.1.4 后项目阶段的风险因素	21
3.2 软件风险因素的分类	21
3.3 调查研究软件风险的影响	23
3.4 小结	25
4 软件风险分析方法及策略	26

4.1 定性分析方法	26
4.2 定量分析方法	27
4.3 软件风险定量、定性分析的策略研究	31
4.3.1 风险因素发生的概率	32
4.3.2 集合风险	37
4.3.3 评估的策略	38
4.4 小结	38
5 软件风险分析原型系统的设计	39
5.1 功能介绍	39
5.2 系统设计	42
5.3 小结	49
6 结论与展望	50
6.1 本文的总结	50
6.2 进一步的工作	50
致谢	52
参考文献	53
附录：1.在校期间的研究成果	55
2.项目风险变量	56

1 绪 论

近年来,随着软件理论和技术不断发展,软件风险管理已成为软件工程领域内保证软件质量的必不可少的关键过程之一。软件风险管理是一项系统工程,它与软件开发过程有着紧密的联系。一般来讲,软件风险是指软件开发过程中及软件产品本身可能造成的伤害或损失,是导致软件项目延迟、超预算或失败的因素。软件风险管理分为定义、识别风险、分析风险、风险计划、风险跟踪、风险应对几个部分,其中每一个阶段的侧重点有所不同,本文主要针对软件风险分析、评估阶段重点进行考虑和讨论,这对于软件风险管理的风险应对具有一定的针对性,对提高软件产业的投资回报率和成功率具有一定的理论意义和工程价值。

大量的统计资料表明,目前风险管理被认为是 IT 软件项目中减少失败的一种

重要手段。当不能很确定地预测将来事情的时候,可以采用结构化风险管理等方法和手段来发现计划中的缺陷,并且采取行动来减少潜在问题发生的可能性和影响。风险管理意味着危机还没有发生之前就对它进行处理。这就提高了项目成功的机会和减少了不可避免风险所产生的后果。

1.1 论文背景

1.1.1 国外软件风险管理的发展现状

国际上关于软件风险管理研究大体上是从 20 世纪 80 年代开始将软件风险管理引入软件项目管理之中,美国软件风险管理之父 Barry Boehm 就提出了一个**重复的、由风险驱动的软件生命周期的螺旋式软件开发模型**^[1],1981 年美国防御系统管理学院编写了一本手册,使项目管理人员通晓量化的风险评估的概念和技巧,以协助他们作内部管理。1983 年美国空军系统指挥部出版了几本风险方面的手册,它们包括具有里程碑作用的降低软件风险的 AFSC / AFLC 手册 80045^[2]。美国空军开发了**软件开发能力评估模型**(Software Development Capability Evaluation,简称 SDCE 模型),并将其作为软件开发实践状态的基础,SDCE 的基本目的是减少获得密集型系统的风险^[3]。1984 年,Carnegie Mellon University 组建了软件工程研究所(Software Engineering Institute 简称 SEI),SEI 于 1987 年研究发布了软件过程成熟度框架,并提供了软件过程评估和软件能力评价两种评估方法和软件成熟度提问单。SEI 风险计划在 1990 年获得了通过(SEI 风险计划的两大贡献是风险管理范例和以分类为基础的调查表,风险管理范例是一个模型,它演示了风险管理过程中不同因素是如何相互作用的

^[4]；以分类为基础的风险识别是一种用风险分类和相关调查表确定软件项目风险的可重用方法。），其目的旨在研究风险方法并将其推广到行业应用中去。1991年SEI将软件过程成熟度框架进化为**软件能力成熟度模型**(Capability Maturity Model For Software, 简称SW-CMM)，并发布了最早的SW-CMM 1.0版，并将软件风险管理引入到CMM (Capability Maturity Model) 中去。软件程序经理网 (Software Program Manager's Network, 简称SPMN 成立于1992年，其主要职责是帮助美国国防部软件采办经理解决他们在管理复杂系统时面临的困难) 他们提出了“最佳实践”，发出了“**最佳实践是创造优良业绩的例程活动**”的倡议，希望找出降低成本与风险和提高软件产量与质量之间的平衡点。经过严密的收集和分析过程，SPMN报告称**规范风险管理是最好的实践**。随着这些软件风险管理理论的出现，国外软件产业界纷纷将风险管理纳入项目管理过程，为此一些软件风险管理自动化分析工具开始被研发并且投入使用，一些大的公司和研究机构也建立了风险数据库，为软件的开发和研究提供一些科学化决策，极力降低软件开发风险，提高软件开发的成功率和追求最大的投资回报率。

1.1.2 国内软件风险管理的发展现状

自70年代出现软件危机以来，学术界和企业界对软件工程环境工具和技术的研究都倾注了大量的人力、财力和物力，多年来也取得了许多成果。但一个不争的事实是，仅有这些并没有达到期望的效果。开始意识到，没有良好有序的管理，任何新技术都是无法得以真正的实施的。项目风险管理在国际上正在成为普遍的实践，但是在我国，知道项目管理的人还不多，实行者就更少。这一事实主要是体现在国内专门介绍软件项目风险管理类的书籍的缺少上^[5]。项目管理作为管理科学的一个分支，在国家教委1997年新修订的学科目录上还没有列入。至2000年4月止，我国还没有一个正式发行的项目管理专业刊物^[6]。这也就说明了项目管理这个学科当时的发展和重要性还没有在科技教育界取得共识，同时，项目风险管理也就更没有得到足够多的重视和系统的研究。随着各种外版书籍的引入、与国际接轨思想的促进，风险管理也随着对项目管理规范化的迫切需求而得到了一定的认识。2002年中国科学院计算机技术研究所、北京中科项目管理研究所推出的项目风险管理分析软件PriskA，该软件要求用户先使用微软的Project建立项目计划，然后由用户使用30多种概率分布函数之一对项目中的不确定性进行描述，从而模拟出项目变化的规律。

目前，在我国软件产业界中对风险管理表现出极大的兴趣，近年来我国引进和自行开发的各种软件工具、开发环境种类繁多，数量浩大，花费达到几十个亿元，但是在具体的项目软件风险管理上基本上又是停滞不前。这其中有许多原因，但归结起来主要有以下几点：1) 大部分软件企业对软件风险管理不是特别重视，

其结果也往往是行不通的；2) 软件开发人员普遍存在抵触风险的氛围，一般组织都会奖励挽救局势于危机的人，而对那些指出可能导致软件项目失败的问题的人嗤之以鼻；3) 没有将风险管理作为整个项目管理体系的关键部分，即使是一些已经实行 CMM 的软件企业（包括已经进入 CMM4 级以下的软件企业）对软件风险管理也往往缺乏一套行之有效的软件风险管理的机制和方法；4) 软件开发人员对风险管理的技术和实践缺乏经验，大部分软件开发人员所使用的风险管理技术只是简单的、非系统的数据分析、检查表填写或者依赖于管理者或开发人员的直觉，它们往往停留在一个定性分析的基础上，不能充分地运用定量的方法去分析、评估风险和跟踪风险和应对风险。

1.2 本文的目的

1、建立适用于软件风险管理一个合理化的分析、评估模型。在目前国内外的研究成果的基础上，力图通过一些风险管理过程和方法的改进，改变国内软件产业界在应对软件风险时的一些定性分析、评估软件风险的缺点，使软件风险分析、评估具有更可靠的操作性。

2、软件风险管理的进行定量化分析、评估风险。近年来，随着相应的支撑环境（如：混沌论、决策论、博弈论、概率论、效力论等等理论以及一些辅助决策性的软件和硬件设备等）和工具的出现，以及人工智能等领域研究的突破，这为软件风险管理的定量化分析、评估奠定了一定的基础，使定量化分析和评估具有极高实现性。

1.3 本文的主要工作

- 1、在阅读大量的中外参考文献后，总结了软件风险管理的基本知识；
- 2、详细地分析了影响软件开发的风险因素，并对这些风险因素进行分类归纳整理；
- 3、深入讨论了软件风险定性和定量分析方法，并对这两种方法进行了研究；
- 4、探讨了对软件风险定性、定量分析的策略
- 5、构建了软件风险量化评估的原型系统。

1.4 本文的主要结构

第一章：介绍了进行软件风险管理的必要性，软件风险管理的发展历史和现状；同时介绍了本文的主要目的、主要工作和主要结构；

第二章：总结软件风险管理的理论基础，风险的定义、软件风险的特征，分析了软件风险管理的经典模型；

第三章：详细地分析了影响软件开发的危险因素，并进行了分类归纳整理，同时调查研究部分软件企业的风险管理情况；

第四章：详细地介绍了软件风险定性和定量分析方法，比较了定性和定量分析方法，分析其优缺点，探讨了软件风险定性、定量分析的策略

第五章：构建了软件风险量化评估的原型系统。

本文在这方面的研究将是一项有意义的理论工作，这也有助于软件产业界能够充分地认识和应对软件风险，并且极大地帮助他们全面加强软件产品质量和提高投资回报率（R O I）。

1.5 本章小结

本章以概要的方式介绍了论文背景、目的及本文的主要工作和结构。

2 软件风险管理的理论基础

2.1 风险管理概述

风险关注未来的事情。这意味着,风险涉及选择及选择本身包含的不确定性,在软件开发过程及软件产品都要面临各种决策的选择。风险是介于确定性和不确定性之间的状态,是处于无知和完整知识之间的状态。另一方面,风险将涉及思想、观念、行为、地点等因素的改变^[7]。

2.1.1 软件风险的分类

风险分为以下方式:

(1) 已知风险,是通过仔细评估项目计划、开发项目的商业及技术环境、以及其它可靠的信息来源(如:不现实的交付时间,没有需求或软件范围的文档、恶劣的开发环境)之后可以发现的那些风险。

(2) 可预测风险,能够从过去项目的经验中推测出来(如:人员调整,与客户之间无法沟通,由于需要进行维护而使开发人员精力分散)。

(3) 不可预测风险,它们可能、也会真的出现,但很难事先识别出它们来。

从宏观上看,风险可以分为以下三大类:

1. 项目风险。由于项目在预算、进度、人力、资源、顾客和需求等方面的原因对软件项目产生的不良影响称为项目风险^[8]。(如图 2.1 所示)

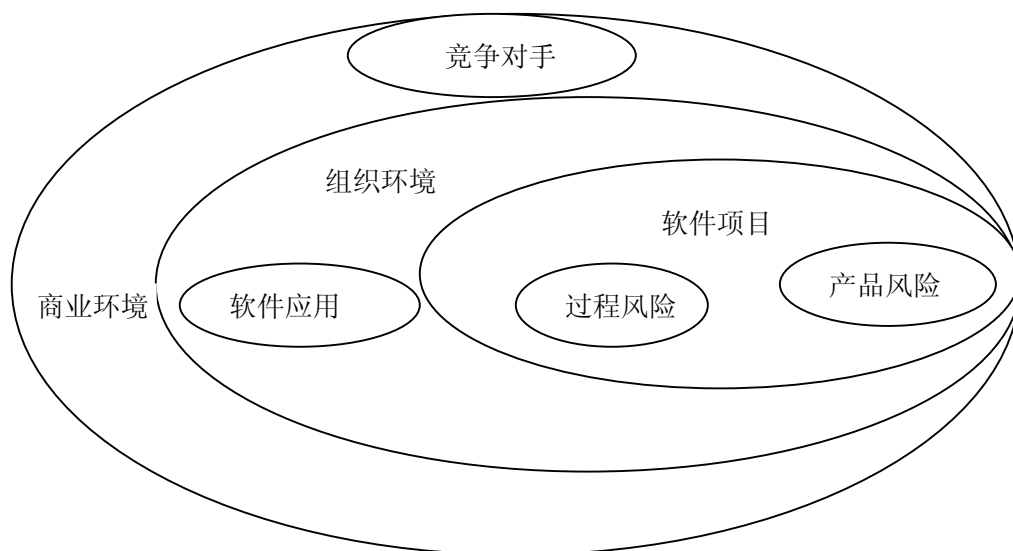


图 2.1 项目风险范围

Fig 2.1 The range of the item risk

2. 技术风险。软件在设计、实现、接口、验证和维护过程中可能发生的潜在问题，如规格说明的二义性、采用陈旧或尚不成熟的技术等等，对软件项目带来的危害称技术风险。

3. 商业风险。开发了一个没人需要的优质软件，或推销部门不知如何销售这一软件产品，或开发的产品不符合公司的产品销售战略，等等，称为商业风险。

一般来讲，软件项目过程是一个不断识别风险、分析风险、计划风险、跟踪风险和应对风险的过程。一个大型软件的开发大约存在 30 至 40 种风险^[9]。如果每种风险都需要 3 至 7 个风险管理步骤，那么风险管理本身就可以构成软件开发过程的一个子项目。一个好的风险管理过程体现了组织在风险管理上的知识和经验，这种知识和经验通过组织制定的过程规范得以确定，组织在风险管理基础结构上的投入，往往也会获得理想的回报。

2.1.2 软件风险的特征

风险与将要发生的事情有关，它涉及诸如思想、观念、行为、地点、时间等多种因素；风险随条件的变化而改变，人们改变、选择、控制与风险密切相关的条件可以减少风险，但改变、选择、控制条件的策略往往是不确定的。通常软件风险有如下特征：

第一，风险存在的客观性和普遍性。作为损失发生的不确定性，风险是不以人的意志为转移并超越人们主观意识的客观存在，而且在项目的全寿命周期内，风险是无处不在、无时没有的。这些说明为什么虽然人类一直希望认识和控制风险，但直到现在也只能在有限的空间和时间内改变风险存在和发生的条件，降低其发生的频率，减少损失程度，而不能也不可能完全消除风险。

第二，某一具体风险发生的偶然性和大量风险发生的必然性。任意具体风险的发生都是诸多风险因素和其他因素共同作用的结果，是一种随机现象。个别风险事故的发生是偶然的、杂乱无章的，但对大量风险事故资料的观察和统计分析，发现其呈现出明显的运动规律，这就使人们有可能用概率统计方法及其他现代风险分析方法去计算风险发生的概率和损失程度，同时也导致风险管理的迅猛发展。

第三，风险的可变性。这是指在项目的整个过程中、各种风险在质和量上的变化，随着项目的进行。有些风险可得到控制，有些风险会发生并得到处理，同时在项目的每一个阶段都可能产生新的风险。

第四，风险的多样性和多层次性。重大工程项目周期长、规模大、涉及范围广、风险因素数量多且种类繁杂致使其在全寿命周期内面临的风险多种多样，而且大量风险因素之间的内在关系错综复杂，各风险因素之间与外界交叉的影响又使风险显示出多层次性，这是重大工程项目中风险的主要特点之一。

2.2 软件风险管理经典模型

目前在国外已经发展了几个经典的软件项目风险管理模型，它们分别是以下几种模型：

2.2.1 Barry Boehm 的模型

在风险管理步骤上，Boehm 基本上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、设置优先级三个子步骤，风险控制则包括制定管理计划、解决和监督三步^[10]。

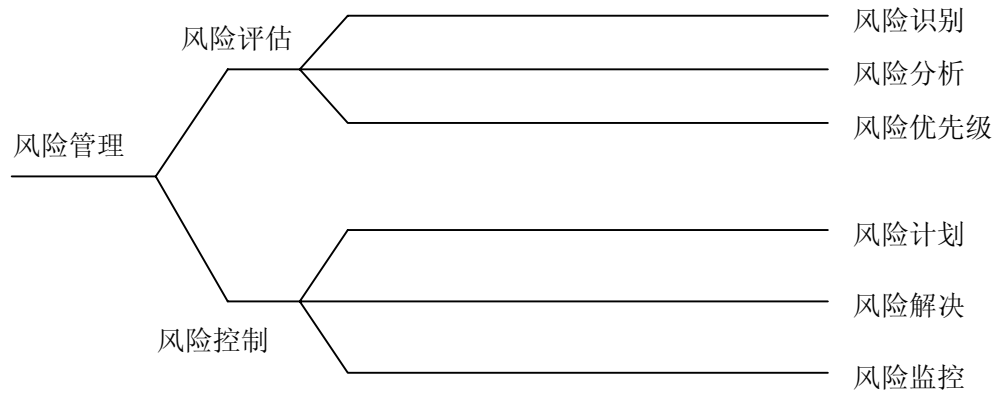


图 2.2 Boehm 风险管理模型

Fig2. 2 Boehm risk management model

Boehm 思想的核心是 10 大风险因素清单，其中包括人员短缺、不合理的进度安排和预算、不断的需求变动等。针对每个风险因素，Boehm 都给出了一系列的风险管理策略例如，对待不现实不合理的进度安排和预算，可以采用增量式开发方法进行软件重用、甚至取消一些需求等。在实际操作时，以 10 大风险清单为依据总结当前项目具体的风险因素，评估后进行计划和实现在下一次定期召开的会议上再对这 10 大风险因素的解决情况进行总结，产生新的 10 大风险因素清单，依此类推。

10 大风险因素清单的思想可以将管理层的注意力有效地集中在高风险、高权重、严重影响项目成功的关键因素上，而不需要考虑众多的低优先级的细节问题。而且，这个清单是通过对美国几个大型航空或国防系统软件项目的深入调查，编辑整理而成的，因此有一定的普遍性和实际性。但是它只是基于风险因素集合的归纳，尚未有文章论述其具体的理论基础、原始数据及其归纳方法。另外，可能由于调查对象的局限性，10 大风险清单并未能全面地覆盖所有的风险因素，而且有些因素实际上说明的是同一个现象。这就意味着风险列表需要改进和扩充，管理步骤也需要优化。

虽然其理论存在一些不足，但 Boehm 毕竟可以说是软件项目风险管理的开山

鼻祖。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。

2.2.2 SEI 的 CRM 模型

S E I (美国卡内基·梅隆大学软件工程研究所 Software Engineering Institute) 认为，在项目开发过程中，风险描述了对项目千万冲击的损失，它的表现形式是产品质量的下降、开销增多、时间延迟、市场份额减少甚至是完全失败。而风险管理的原则是：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的实施的有效性。S E I 的持续风险管理模型有 7 个原则：全局观点、远视的观点、畅通开放的沟通、集成化的管理、持续的过程、统一的产品观点以及团队合作。

C R M 模型 (Continuous Risk Managements 简称 CRM，持续风险管理) 要求在项目生命周期的所有阶段都要关注风险识别和管理，它将风险管理划分为五个步骤：风险识别、风险分析、风险跟踪、风险控制。软件风险管理过程模型是美国软件工程所 (SEI) 在 1992 年提出的一个风险管理范例 (如图 2.3 所示箭头标识了信息的逻辑流，而沟通则是信息流的核心和手段)，它显示了应用 CRM 的基础活动及其之间的交互关系，强调这是一个在项目开发过程中反复持续进行的活动序列。

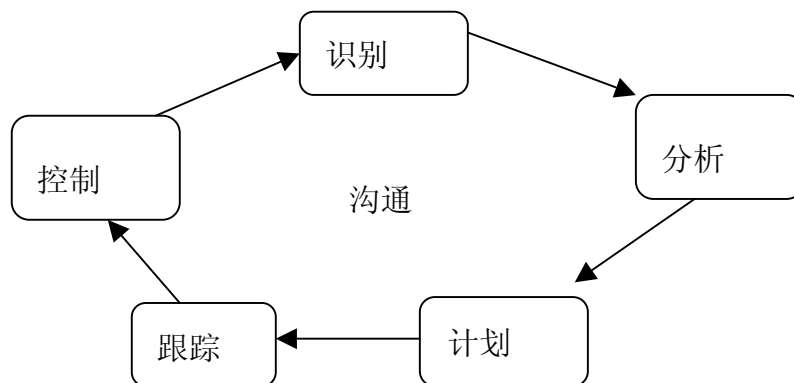


图 2.3 SEI 风险管理范例

Fig 2.3 SEI risk management example

SEI 分别从软件风险管理的风险识别、风险分析、风险计划、风险跟踪和风险应对的各个管理过程用 IDEF0^[11] (Integrated Computer-Aided Manufacturing Definition 简称 IDEF0，一个标准的过程定义) 数据流程图表从两个视角描述了软件风险管理的管理过程 (如图 2.4)；外部视角说明了过程控制、输入、输出和机制，内部视角说明用机制将输入转变为输出的过程活动，且清楚的描述了软件风险管理过程中各个阶段的相互影响、相互作用的关系。软件风险管理过程模型通过控制、输入、输出、和机制描述了顶级过程，控制决定何

时和如何执行，输入就是一个过程转变所需的项，它必须满足过程入口标准，输出是过程转变的结果，这一结果已经通过了过程出口标准的评审，机制决定了过程所用的方法

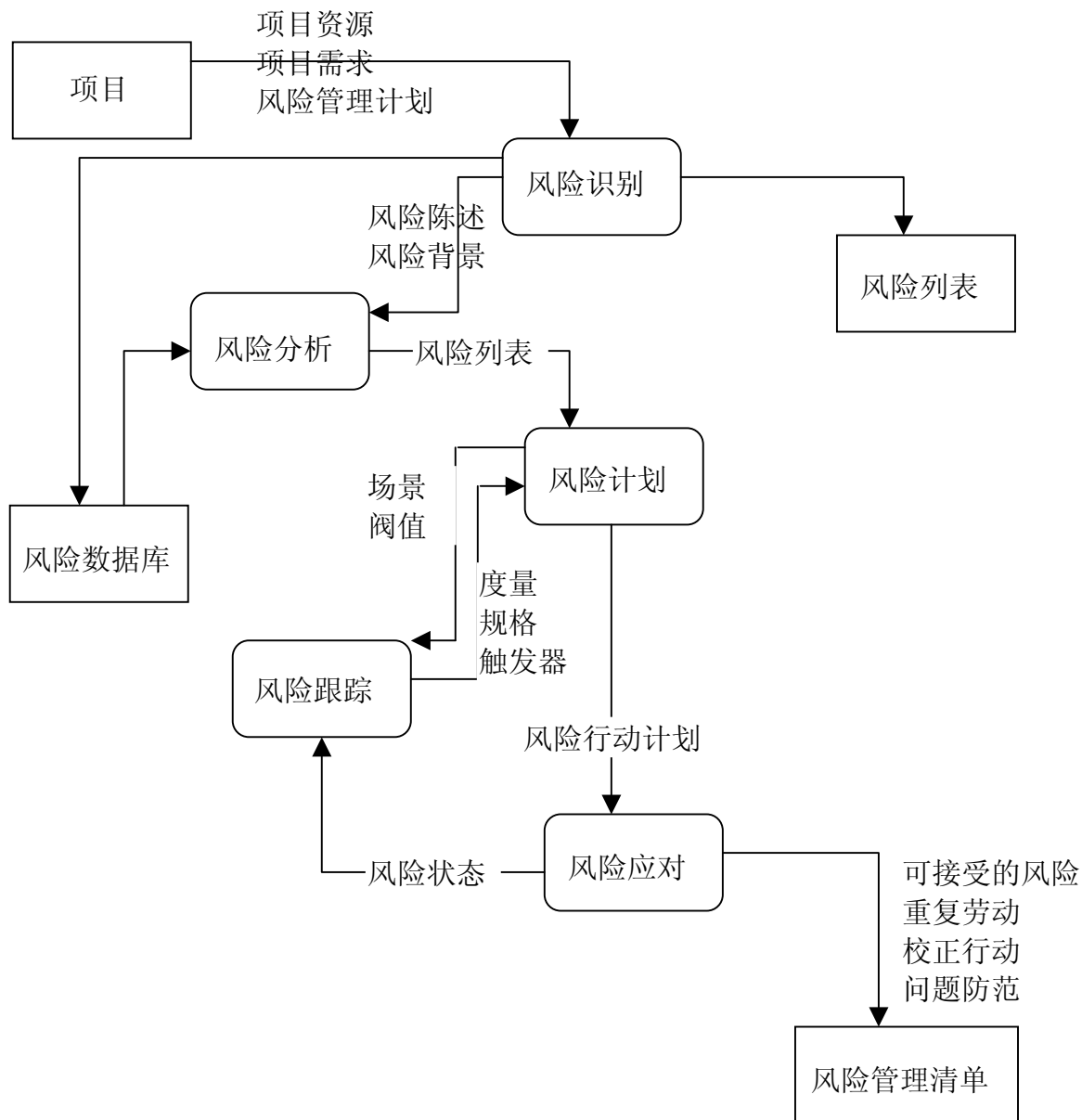


图 2.4 SEI 风险管理过程模型

Fig2.4 SEI risk management process model

2.2.3 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成四个有趣的组成部分（见图 2.5）：任务、结构、角色和技术。这四个组成部分和软件开发的各因素很好地对应起来：角色覆盖了所有的项目参与者，例如软件用户、项目经理和设计人员等；结构表示项目组织和其他制度上的安排；技术则包括开发工具、方

法硬件软件平台；任务描述了项目的目标和预期结果。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部分的不一致，就会造成比较严重的后果，并可能降低整个系统的性能。

将这个模型和软件风险的概念相对应，即：一个系统开发过程中，任何 Leavitt 组成成分的变动都会产生一些问题甚至导致软件的失败。根据 Leavitt 模型，任何导致风险发生的因素都可以归结为模型中的组成部分，例如技术及其可行性；或者归结为组成部分之间的联系，例如程序开发人员使用某一技术的能力。因此，使用 Leavitt 模型从四个方面分别识别和分析软件项目的风险是极其有条理性和比较全面的。例如，在管理任务时，可以采用 BOEH 提出的持续监视风险管理方法；在管理角色时，可以采用 Kell 和 Robert 等的思想，着力提高参与者的风险管理意识，提高使用工具的能力，鼓励揭露存在的问题等。

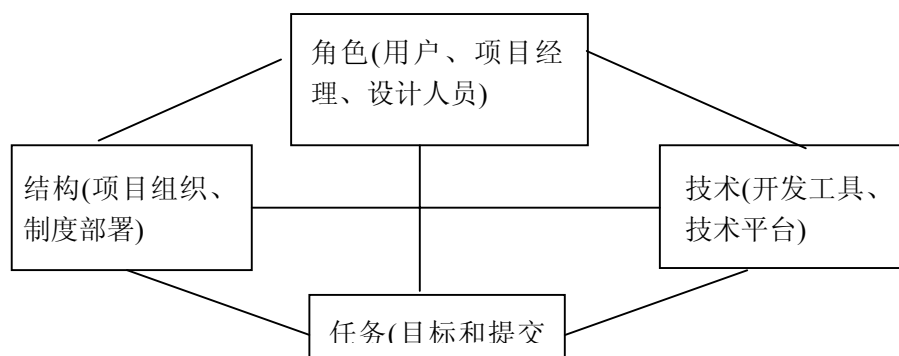


图 2.5 Leavitt 模

Fig2. 5 Leavitt model

Leavitt 模型实际上是提出一个框架，可以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软件中，它所考虑的都不得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。

2.2.4 Riskit 风险管理过程

Riskit 过程是由 MARYLAND 大学提出的，旨在对风险的起因、触发事件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。

Riskit 方法的核心是用来描述风险的图形形式化工具分析图 Riskit 分析图 (Riskit analysis graph)，该分析图可以显示定义风险的不同特性，比常规的口头论述要更为形式化。它是风险管理过程中主要的沟通工具。分析图所描述的元

素及其关系（如图 2.6）所示。

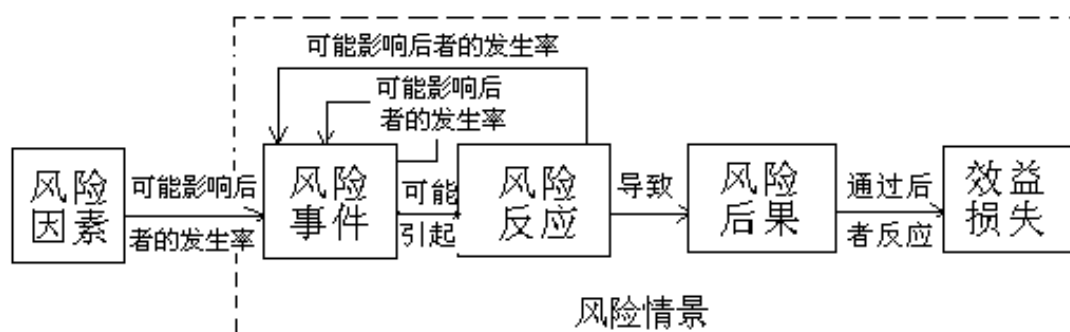


图2.6 Riskit分析图

Fig2.6 Riskit analysis

其中，风险因素是指影响风险事件发生的可能性的特征，例如使用新的开发工具不稳定的需求等，它用来描述主要的项目环境特征。风险事件指的是对项目有负面影响的事情，它可能被多个风险因素触发，也可能性影响其他的事件或者因素。项目关键性成员的退出、一个主要需求的大幅度修改等都属于风险事件。而风险反应描述了对风险事件及其后果所采取的相应措施，它也可能影响风险事件发生的可能性。风险后果表示了风险事件对整个项目的最后影响，例如项目延迟 2 个月提交、被迫放弃实现一些功能。风险后果主要是反应了风险对项目目标的严重性，例如项目开发人员所感知的挫折感、CEO（Chief Executive Officer 执行总裁）等决策层所感到的挫败感等。

在 Riskit 分析图中，使用不同的颜色或者形状表示不同的元素，可以详细而全面地表现与项目相关的风险信息及其制约关系。在此基础上可以 Riskit 以风险管理过程。Riskit 风险管理过程包括 7 个基本活动，在项目生命周期内，这些活动可以重复多次进行，也可以并发地开展。其中，1) 风险管理定义，指的是定义风险管理的作用域、重点、责任和频率等，并辨识所有相关的风险承担人员；2) 目标查阅，指的是查阅已经确立的项目目标，重新定义它们或者显式地补充定义一些原来隐含而未明确表达的目标，并且整理风险承担人员和这些目标的关系；3) 风险识别，指的是使用不同的方法识别对项目成功构成潜在威胁的元素，产生一系列的原始风险列表；4) 风险分析则是对风险元素进行划分和确认，构造完整的 Riskit 分析图，以对每个风险元素有清晰的认识；5) 风险控制计划对最严重的风险元素提出风险控制方案并选择最合适的计划；6) 风险控制则是对风险控制计划的实现；7) 风险监控保证在项目开发全过程中对风险情况进行监督和控制。

对于每个活动，Riskit 都提供了更加详细的定义模板，包括活动的描述、启

动规则、采用的方法和工具、活动负责人、退出活动规则及其具体活动所特有的属性等。而每个活动又可能由更加具体的子活动组成，例如风险分析又可以分为：将原始风险因素筛选、处理、组织以成为可管理的风险因素集合，根据 Riskit 分析图将风险因素的详细信息文档化，对风险因素进行优先级排序等。总而言之，Riskit 提供了多方面多层次的风险管理过程体系。

2.3 本章小结

逃避风险永远无法获胜。除非在软件的开发过程中，一直处于无风险的状态。在进行软件开发的过程中，要正视风险，不是逃避，也不能去追求零风险状态，而是要将风险带来的损失降到可控范围内，同时还要让利益最大化。而能让在蕴含风险和机遇的环境中演绎成功的行为准则就是风险管理，软件风险管理就是为解决影响软件项目、过程或产品的风险而制定的准则。只有充分地理解和学习软件风险管理的知识和技能，同时在实践中不断地积累经验才能有效地进行风险防范和控制，达到减少风险的影响程度和实现利益最大化追求的目的。

3 软件风险因素的分析

从国外成熟的风险管理模型和范例中我们可以看出风险识别是风险管理的第一步，没有有效的可重复的风险识别方法进行成功的风险管理是不可能的，所以我们首先要识别在软件开发过程中的风险因素。

风险因素指可能导致某种风险事件出现的软件系统的环境特征。由于软件项目开发环境有许多相同的特征，如用户需求变化、测试不够完善等。因此，风险因素是指那些与项目特定环境有关的特征，如项目组关键人员流失、不熟悉用户界面开发工具等。

3.1 影响软件风险的因素

风险管理有 10 项原则，分别为：确定的产品需求，并要求文档化；项目应该追求最大的利益，而不是让风险降到最低；有些不确定的风险要在计划过程中忽略，以更多的精力控制那些确定的风险因素；计划时为项目多留余地，比如多准备一套技术解决方案，这对大型项目更为重要；项目过程中要随时掌握风险的真实情况；尽量避免风险因素变成问题；项目团队内部人员应该相互沟通，共同了解风险因素，并齐心协力控制风险因素；充分利用历史项目的经验及风险管理数据；为项目团队成员分配相关的风险责任；在合同中明确说明客户及开发商之间的风险责任。在国内外的一些学者和专家已经尝试总结威胁软件开发成功率的因素，如 ALTER 研究了 56 个决策支持系统的实现后，识别了八大风险因素：不合作的用户、多个用户或实现单位、在多个参与者之间交接、无法指定目标或者用途、无法缓冲对其他人的冲击、缺少支持、缺少经验以及技术或投入的效力问题^[12]。又如 McFarlan 将他所提出的三大角度映射到软件项目开发中，即项目大小、应用技术的经验和项目结构。而 Boehm 则提出 10 大优先级排序的软件风险因素：1.产品定位错误（包括市场定位）2.人员流动 3.项目管理失败 4.开发目标不明确或摇摆不定 5.开发计划执行受到严重影响 6.技术方案有缺陷 7.项目经费超支或不足 8.开发环境及过程管理混乱 9.产品质量低劣 10. 需求发生变化。再如，由 Roger S. Pressman 编写的软件工程经典著作，1999 年出版的《Software Engineering ,A Practitioner`s Approach, Fourth Edition》中，将风险分为产品规模、商业影响、客户特性、过程定义、开发环境、建造技术和人员数目及经验七风险种类，每个风险种类又有 8 到 20 个风险因素。SEI 曾经将软件项目风险分成 13 大类共 200 余种^[13]。SEI 的风险识别方法基于如下假设：

- 一般来说，软件开发风险对项目技术人员而言是已知的，但却未被很好

地表达出来;

- 对与风险管理保持一致来说, 一个结构化的可重复的风险识别方法是必要的;
- 风险识别必须覆盖项目相关的所有关键的开发和支持领域;
- 风险识别过程必须创造和保持一个非决断式的风险导出环境, 这样才能获得实验性的或引起争论的观点;

单单依据未被覆盖到的风险数量或本质是不能作出关于项目成功或失败的全面的判断的。

通常, 我们可以根据具体情况并结合软件开发生命周期的各个过程, 对影响软件开发生命周期的风险因素进行以下划分:

3.1.1 项目来源阶段的风险因素

①外部因素

许多风险都和项目之外的因素有关, 而通常是控制不了这些因素的。这就要求更多地协调项目之外的关系。但通常情况下, 也只能尽可能详细、准确地跟踪这些外部因素, 在其发生、变化时及时采取应对措施, 以尽量减少对项目带来的影响。

几种典型的外部因素是: 最终客户的相关信息; 与内部或者外部中间承包商之间的关系; 经过有效培训或者有经验的客户; 从另外一个项目获得的相关经验。

②需求问题

大多数软件项目都面临不确定甚至是相当混乱的需求, 当这些不确定的需求在项目早期出现后, 如果不及时对需求状况进行控制, 那在以后的项目过程中会产生更大的问题, 甚至导致项目失败。

需求问题主要有: 缺乏清晰的产品认识; 缺乏关于产品需求的协议或者相关文档; 在需求分析过程中没有与客户很好沟通或者由于客户的其他原因而导致需求分析没有正确、完整地结束; 没有区分产品需求的优先级; 在项目进行过程中不断有不确定的新的需求产生; 需求改变过快; 无效的需求变更管理过程; 无效的需求变更分析。

③管理问题

项目管理的缺陷也会影响到项目成功。但由于项目经理都是风险管理计划的编写人, 通常不可避免地会忽略或者避开自己管理方面的问题, 而这恰恰是不可忽视的。

管理风险可以分为: 制定了不恰当的任务分工计划; 未了解到实际的项目状态; 项目的所有权和决策权不清晰; 部分不合适的理由导致对客户不切实际的承诺; 管理者或者客户不切实际的期望值; 团队成员之间的矛盾和个性冲突等。

④缺乏项目相关的知识和技术

软件技术日新月异，意味着项目团队必将缺乏足够的技术和知识以保证项目的成功。这些因素可以归纳为：缺乏必要的培训；对相关的方法、工具或者技术有不正确的理解；在具体的项目相关行业领域没有足够的经验；采用了新的技术或者开发方法；无效的、不充分的过程文档，或者忽视了编写文档的过程；采用的方法或者技术没有起到正确的作用，或者没有满足这些方法或者技术的前提条件。

3.1.2 项目阶段的风险因素

由于每个项目有不同的特点，因此具体的风险分析要区别对待。在目前的情况下，可以根据项目阶段来划分不同的风险。以招标项目为例，来看看各个项目阶段的风险是如何构成的。

①决策阶段的风险

项目决策阶段包括竞标、立项，主要风险当然来自于软件项目竞标过程。在竞标之前，要分析该项目是否值得公司去做，是否对公司带来经济价值或者社会价值。确定能够对公司有利后再参与竞标。那么，竞标过程存在哪些风险呢？

竞标阶段是软件开发企业能否成功获得用户的支持并取得项目开发权的重要时期。竞标阶段的风险主要来自软件企业、竞争对手的技术研发能力和市场能力，以及在竞标过程中公司的方案保密能力。商场如战场，知己知彼，百战不殆。分析竞争对手的能力和保护好公司的技术机密是竞标胜利的重要条件。

1) 竞争对手的技术能力

a. 竞争对手的软件开发能力

目前，大多数国内软件厂商并没有通过 CMM 相关认证，因此，考察点主要是竞争对手的项目市场人员关于项目开发的理解以及对方案的影响能力。一般来讲，软件开发能力较强的竞争对手在方案编制时会有很细致的调研工作，这些能力可以通过观察竞争对手在用户中的活动了解到。当然，其他企业具备的资质证书最好自身也有，如 ISO9000 认证。

b. 竞争对手的行业知识能力

如果面对的是某一个行业领域，那么，考察竞争对手在该领域的能力也是风险因素中很重要的一面。由于每个行业都有特殊的专业知识背景，这就要求软件开发商在该领域必须有熟练的专家、工程师或者顾问，否则无法与用户沟通，更不用谈开发软件了。

c. 竞争对手的项目管理能力

如前所述，项目成功率如此之低，那么对于项目管理的能力也会成为分析的重点。现在的用户大多数都已经熟悉了项目管理，因此，软件开发商在方案汇报

过程中对项目管理的认识也是决定能否得到用户支持的重要因素之一。考察对手的项目管理能力可以从方案编制过程、方案影响程度、方案编制人员的构成、项目市场人员的素质等几方面分析,或者从其公司的主页上了解基本信息。

2) 竞争对手的市场能力

竞争对手的市场能力包括公司形象、公司理念、市场影响程度、同类产品或者技术在客户群体中的占有量、对手对特定客户的影响、竞争对手和特定客户的关系等内容。

a. 竞争对手的市场影响

竞争对手如果是大公司,那就存在风险。一般来讲,用户都倾向于使用大公司的产品或技术。例如在调查过程中有一家公司谈论到 2000 年他们在一次招标过程中发现,采用 IBM 的产品报价甚至可以比其他公司的同类产品报价高 8%而得到用户认同。因此,如果竞争对手是大公司,就需要在报价、产品质量等方面多加考虑。对于小公司,如果竞争对手在该产品或者技术领域有很大的影响,这时也需要仔细衡量。

b. 竞争对手对于特定客户的影响

特定客户是指对该项目或产品有绝对决策权或者决定使用权的用户。如果竞争对手和特定客户长期合作并受好评,那就是一种风险。在每个行业,每个单位都有特定客户,应该按照用户单位对特定客户进行分类,并做好各种思想准备。对于招标项目,还要注意另类特定客户,即喜欢提问题并喜欢让汇报人难堪的用户。这些客户都是需要早期进行专门拜访的人。

c. 竞争对手的特定客户关系

竞争对手的特定客户关系是指对手在用户群体中有亲戚、同学、朋友等,并且能够对项目的决策起一定作用的用户。如果竞争对手有这方面的关系,则对软件企业就是一个较大的风险。因为在技术相同、报价差别不大的情况下,用户更倾向于采用自己熟悉的公司的产品,这也是人之常情。

3) 方案机密风险

业界曾经出现过很多这样的案例,现在所谓的商业间谍不又仅在探听对手公司的商业秘密,也采取各种手段获取对手的技术资料。他们采用的渠道也很多,例如,通过对手公司的网站、挖取对手公司的技术骨干员工、伪装对手公司的客户听取公司的汇报,甚至采取卑鄙的偷窃手段从不小心的用户办公室或计算机中获取对手公司的技术资料。用户的计算机、服务器等如果联上 Internet,也会成为部分商业间谍的攻击对象。对于软件开发企业来说,方案机密应该由公司的保密制度控制,并由公司的方案设计人员具体负责。如果方案泄密,会对公司的竞标带来直接的威胁。

在调查时,有位项目经理讲述了他曾经亲身经历过的一次涉及 200 万人民币的项目招标,由于他曾经多次给客户汇报,留在客户那里的文字材料不下 20 份,还有多处客户留有电子文档,在招标完成后他们发现在技术分上和自己相差无几的对手的方案材料几乎和自己的方案一模一样。而对手只用了 3 天时间编写,而他自己却用了 3 个月的调研时间。因此,管理好招标前的技术方案是竞标过程中很重要的环节。

3.1.3 项目实施阶段的风险因素

根据 1999 年 SEI 的统计,商业应用软件的风险主要有 5 类:不充分的用户文档;较低的用户满意度;太多的市场营销时间;有害的竞争活动;诉讼费用^[14]。

一般来讲,实施阶段的风险主要有 6 点:进度风险;技术设计风险;客户满意度风险;协作关系风险;人力资源风险;项目验收结算风险。

①进度风险

进度风险常常是软件开发项目的第一个风险因素。大部分项目都不能按期完成的主要原因在于计划和实际情况的差别。

a. 多数项目是在一定的“政策”前提下提出来的,因此一般都要求赶工期,这在传统项目中很常见。对于软件开发项目,工期一般是一定的,按照软件工程理论和实践经验,一般中、小型系统能够准确地估计时间。但是,大多数用户都要求开发商在某某工期内完成,远远短于实际开发需要的时间。这时候要求讲信誉的软件开发商就不能完全同意客户的意见,双方要协商提出一个合理的开发周期。

b. 每个人都希望尽早地看到成果,这是人之常情。大多数用户都希望开发商在工作 1—2 个月的时候就能够有产品可供用户使用,叫做阶段产品。一方面是用户需要让领导看到该项目的进展,另一方面用户也希望准确了解项目进度和问题。这种希望是好的,可以理解。但是,软件开发项目常常需要很长时间的调研过程,尤其是需求分析阶段,通常都在 2—3 个月时间。设计时间通常都是需求分析时间的 1—1.5 倍^[15] (PSP Technical Report. Humphrey, 2000)。这样,用户要真正使用软件最少也应该在 4 个月以后。

c. 技术难度系数估计不足的主要原因是开发的软件项目中有以前没有开发过、目前也没有试验过的内容。对于陌生的领域,无论软件开发商怎样承诺,都会有风险的。这种风险表现为根本实现不了或者实现起来时间很长。如果遇到这方面的风险因素,应该及早地进行试验,并和用户协商以取得有效的方法来保证将风险的影响降到最低。

d. 特殊原因很多,包括国家政策、经济形势、交通状况、行业政策、领导干预等社会、政治经济因素,也包括用户单位制度改革、地域发生变化等客观因素。

②技术设计风险

a. 软件体系结构设计思路

目前的软件体系结构很多也很杂。在大多数用户还没有规范管理的情况下, 由于应用的复杂性、用户需求的变化性、规范的不标准性等因素, 影响了软件开发项目的结构思路。如果在体系结构方面设计错误, 带来的结果是项目失败, 而不仅仅是功能不全等可以弥补的过失。另外, 软件体系结构中还有一个重要的内容就是目前的 B / S(Business/Serve) 结构和 C / S(Customer/Serve) 结构的合理使用。对于大多数应用来讲, 如果是信息查询、报表系统等信息检索软件内容, 则可以使用 B / S 结构, 借助浏览器用户可以很方便地获得所需数据。但是, 对于需要较强的用户交互界面(如图形处理软件)、业务逻辑规则复杂的计算或者需要单机独立运行的程序(如多媒体汇报系统或者图形管理系统中的数据库管理工具), 还是强烈建议采用传统的 C / S 结构。

b. 平台选用原则

大多数客户都有自己熟悉的数据库平台或者应用服务器平台, 有的选用了 Oracle 数据库, 有的选用 Sybase5 有的选用 WebLogic 或者 IBM 的 WebSphere, 在软件设计方面就不得不考虑他们选用的系统。

又如办公自动化平台, 目前较为成熟的方案有微软的 Office + Exchange Server 和 IBM Domino + Lotus Notes 两种方案。如何选择平台就得看用户习惯或者他们如何做决策, 最后才能看你如何推荐了。

c. 接口设计

由于目前的应用服务器(无论是 IBM 的 WebSphere、Oracle 的 IAS、SliVerStream 的 E—Port 还是国内开发的应用服务器)还不能真正解决遗留系统的数据共享和无缝连接问题, 在给企业客户做软件开发项目当中必然会遇到许多接口设计的问题。另一方面, 设计的系统如果比较复杂, 涉及到几个系统间的通信就是接口设计问题。再者, 如果软件是采用组件方法开发的, 的接口标准也是应该重点考虑的方面。

d. 数据库设计

目前所有软件几乎都要涉及到数据库。小的数据库可能是几张表, 关系很简单。大的数据库系统则是几百张表, 关系也非常复杂。由于历史原因, 国内很多企业的数据库在关系方面都没有设计得很好, 虽然很多客户宣称自己的数据库符合第三范式, 但你却能看到他们自己经常在表中添加诸如字段 10、字段 11 之类的东西。也有很多企业有数据库标准, 但这些标准在企业内部并没有很好执行, 这就要求你要采取更加复杂的手段来应付这种“既要符合标准, 又要先进”的问题。

e. 专业设计

客户需要的软件专业性都很强，因此对于用户群体专业的理解在很大程度上决定了项目的效率和最终用户满意度。专业性很强的客户要求项目团队不断和用户交流，掌握用户需要的每一个细节。因为，真正的专家在现场。

③客户满意度风险

客户满意度是项目开发中极为重要的一环，通常包含四个方面的问题：

a. 功能满意度

软件开发完成后，最主要的目的就是达到用户要求的功能。大多数合作开发的软件都能够达到用户要求的功能，因为这是和用户共同开发出来的，用户在软件开发过程中充分地参与了项目，并在项目实施过程中不断地对软件开发提出建议。但是，如果在前期没有进行详细地分析和设计，也会给最终产品带来意想不到的结果。

b. 用户界面

良好的用户界面是获得那些可能被新系统改变了工作方式的客户认可的重要条件之一，设计良好的用户界面，就像给客户一个良好的工作环境一样重要；而界面枯燥、繁琐会直接将产品推向用户的垃圾堆。

c. 协作关系

如果软件项目是由甲、乙方共同开发完成，那么协作关系就成了客户满意度很重要的一个环节。在项目过程中，项目团队成员中每一个人都负有责任和义务同用户保持良好的沟通渠道，并及时获得用户的意见和建议。只要有一次对用户建议不理睬的行为，就足以使你的项目走向死亡。

d. 客户关系

软件项目周期长，在合作过程中双方的关系会随着时间的推移而发生变化。尤其代表公司的项目小组成员，必须和用户建立良好的合作关系。这样才能保证项目顺利完成。

④协作关系风险

协作关系是指项目可能由几家开发商合作开发。

a. 合作开发的协作关系

指和用户合作开发的协作关系。由于用户大部分时间都有自己的本职工作，不一定能够有完整的时间真正参与到项目中来，这就带来一定的人力资源风险。

b. 合作招标的协作关系

由于项目过于复杂，或者公司技术力量不能达到要求，或者合作方在用户群体中有特殊的客户关系，就会促成这种合作招标由几家单位合作开发的情况，这时候，竞争对手变成合作伙伴。

在国内这种合作方式对项目很不利，尤其合作方是由于特定的客户关系取得胜利的时候，这种项目通常都不会成功。要想合作成功，最重要的就是将双方的职责划分清楚，并确认双方的系统间要有最小的幅合度，如果交叉太多，失败的可能性也就越大。这种风险在规模较小的软件企业里更应该注意。

c. 委托开发的协作关系

即公司目前的技术或者人力资源不足时，将项目的一部分或者全部委托给第三方开发。一般情况下，都应该将这种关系报告给用户的相关领导，并征得用户的同意。委托开发最主要的问题就是时间和质量。受托方在不一定非常了解该项目的需求的情况下就开始编制软件，这种情况在国内是常见的。

d. 协作方的地域问题

由于项目周期一般都不足，因此地域位置距离较远的项目要在很大程度上考虑距离带来的项目周期延长。

⑤人力资源风险

软件开发是知识性的人为劳动，而目前世界上所有国家的软件开发人员流动性都很大。因此，人力资源风险也是项目风险中的重要因素。主要有：

a. 项目组成员流动

即组成项目组的开发人员在项目过程中发生变化。这种流动可能会导致项目小组成员之间的关系不协调，例如团队中加入一位个性和其他人冲突很大的成员等。

b. 项目经理意外事故

项目经理是项目的核心成员，直接掌握项目的计划与控制，故项目经理的稳定性很大程度上决定了项目的成败。

c. 开发人员的技术能力

组成好的项目组对软件开发项目的成败有很大的关系。开发人员必须是能够完成软件开发项目中所有需要的技术能力的人员。如果团队中有培养的对象，就必须考虑人员冗余，即多配置一名熟练的技术人员。

⑥项目验收结算风险

项目后期，最重要的工作就是维护和验收。一般情况下，这时的项目都具备了验收条件。但由于还有维护时间，还需要公司继续投资。因此，在验收完成后，用户能否及时支付项目尾款和公司的投资能力在很大程度上决定了项目是否能够真正完成。如果用户暂时不能支付项目尾款，则公司就需要投资继续完成维护工作。这种情况在一些不景气的企业客户中常有发生。结算风险一般包括：

a. 用户的支付能力

如用户在前期的投资预算超支或者项目资金被挪作他用，用户就不能及时支

付项目款项，这时，开发商并不能完全放弃，应该从公司的流动资金中拨出该款项继续开展项目后续工作。在公司支付能力有限的情况下，就应该和用户协商，尽量督促客户支付项目款项或者采取其他措施。

b. 结算周期长

用户具备支付能力，但由于其他原因使得用户不能够及时支付现金。这种情况常有发生。原因包括人事关系、签字领导出差、资金暂时冻结等等。

c. 客户关系

在项目进行过程中，如果公司和客户之间由于某些原因关系紧张也会影响到结算。

3.1.4 后项目阶段的风险因素

后项目风险是指项目验收完成后遇到的风险。之所以分析这些风险，是因为必须考虑项目完成后继续对用户的技术支持，以建立长期的合作关系。后项目风险主要包括：

①灾难救援风险

灾难通常来自于用户。例如，用户在使用系统过程中，由于错误操作导致数据库服务器崩溃、数据损失，或由于用户的服务器连接了 Internet，导致病毒破坏或数据泄密等。灾难救援工作通常是问题发生后才启动，这时最重要的就是帮助用户解决问题，并帮助用户制订以后防范的措施和政策。

②客户需求变更

软件开发完成后，在使用软件的过程中，用户又提出新想法或改变了想法。并请求软件开发商帮助解决该问题。通常做法是，在一定的工作量范围内尽量帮助用户解决问题，尽量满足用户的需要。但应该有一定的限制条件，否则会大大增加项目成本。一般来说，如果超过 1 人月工作量，就需要和用户协商是否重新立项或者采用其他解决办法。

③其他不可预计风险

例如，对某企业客户，用户在项目完成后，正准备推广该系统，但由于领导换届，新任领导对该系统并不认同，那么系统的推广工作就会受到很大阻力。

3.2 软件风险因素的分类

由于一个软件开发中的风险可以是已知的、未知的或不可知的。没有有效的可重复的风险鉴别方法想去进行风险管理根本上是实现不了的。所以我们要识别风险，同时对影响软件的风险因素进行分类。我们可以结合软件生命开发周期将软件开发过程中的风险因素组织成二个层次：类别和属性。

经常对风险评估结果进行分类可进一步看清风险的本质，在进行风险分类时，

我们通常可以按照以下 5 个步骤：

1、阐明。根据标准符号写出风险陈述。用最简单的术语描述风险，提炼对问题的看法；

2、取得一致。就风险陈述的含义取得一致认识风险描述的含义。一旦每个人都理解了问题就应该就书面描述的所写的问题范围取得一致。取得一致不是将风险验证为风险，而是文档化问题的含义和意图；

3、分类。从风险分类系统中选出风险类别，如 SEI 软件风险分类法[CARR93]。如果两个或者更多类别可用于描述同一个风险，这个风险就可能是复合型风险；同时尽可能将风险细化，并且注意将相关风险联系在一起的依赖关系；

4、联合。按风险类别为风险陈述分类。积累并评审一个风险类别的所有问题通过来自不同角度的风险文档，获得对这一类风险的整体认识；

5、精简。去掉重复的风险并总结相关的风险。总结了一个逻辑领域的问题后，即可轻松找出并去掉重复的风险。然后用风险类别将相关的风险与其他风险分开。

风险分类法是一种有效的工具。通过它我们可以获得广泛的系统水平的风险^[16]。这些风险通常由程序员鉴别，并且按照分类法的层次结构分类。此外，分类法识别各种领域风险从而可以对它们进行更详尽的调查，而且它是通过同等的管理者、工程师和技术支持人员间的交互来使用。按照风险因素的类别、属性二个层次来描述了软件风险因素分类的层次结构(如图 3.1)。

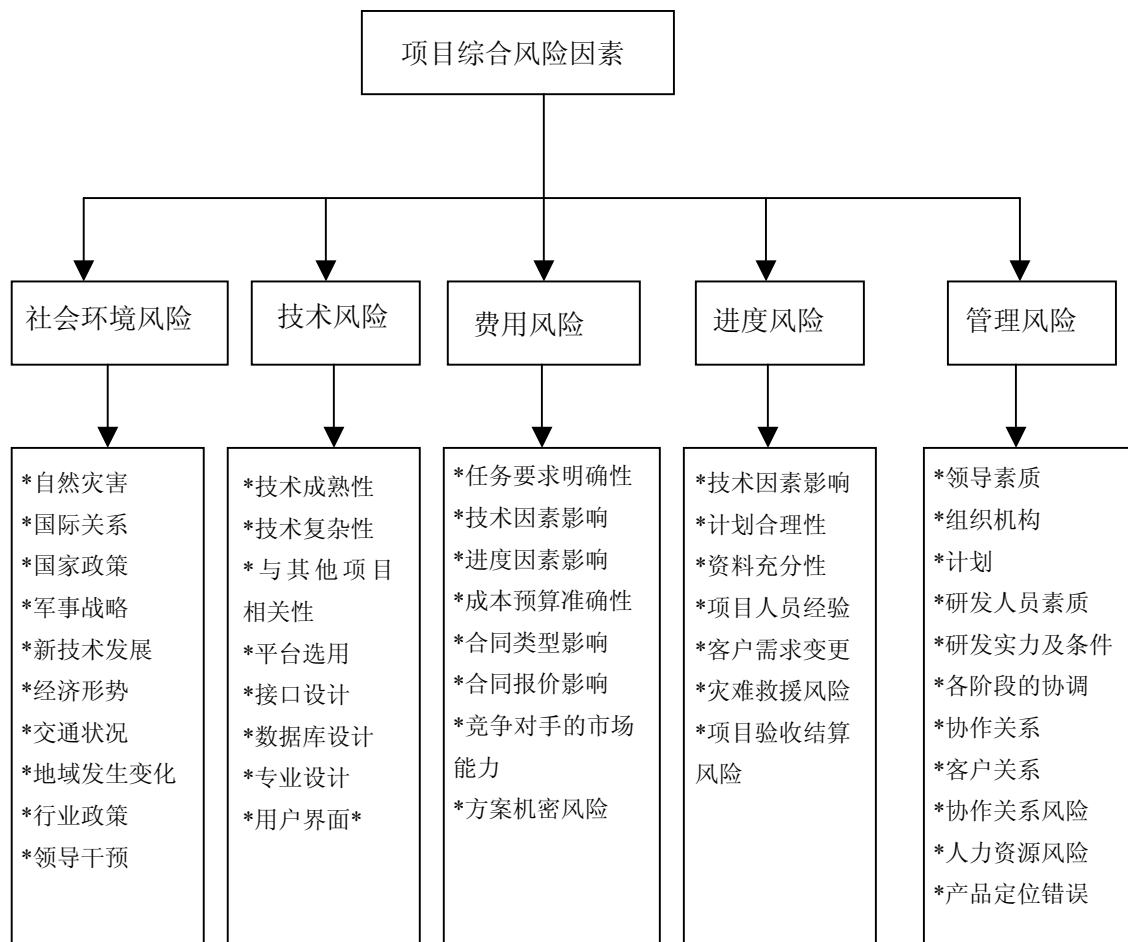


图 3.1 软件风险因素分类的层次结构

Fig 3.1 Arrangement framework of software risk element class

3.3 调查研究软件风险的影响

前面结合软件开发生命周期总结出一些影响软件项目开发的风险因素和分类，但正确与否还必须通过实践的检验。在调查过程中，借鉴和采用了“问卷调查+数据分析”的验证方法。问卷问题包括是非题、定量题和程度选择题等。

在调查开始时，根据每个风险因素对软件开发的综合影响力的大小进行排序，找出影响力最高的若干个风险因素加以特别的关注，使用 Delphi 法或少数服从多数的方法，根据 Delphi 流程^[17]（如图 3.2）挑选出需要的风险因素项。并且结合专家的知识库衡量出风险因素的出现概率和综合影响力大小，确定出 21 个风险变量指标 41 个问题，采用当面访问讨论为主、邮寄信函和 E-mail 咨询为辅的方式，对 10 名在软件企业兼职的在校学生、5 名软件公司项目负责人、3 家外包开发软件的单位进行了调查。问卷采取区间单位法，分为：可以忽略、不太重要、

不确定、比较重要、十分重要五大区间。（这是参考了 Borda 的方法。Borda 将风险等级划分为五级：十分严重、严重、中等、微小和可忽略，不同的级别代表了风险发生时造成的成本增加、进度滞后、需求无法完全实现的程度。）在进行数据分析和计算时，又将 Borda 的五个别转换分别为对应的 0.9、0.7、0.5、0.3、0.1 的数值（这主要是考虑到便于数据统计和分析）。在调查的最后总共收回了 31 份问卷反馈，进行综合分析后，得出了常见的风险项目：不合作的用户、多个用户或实现单位、在多个参与者之间交接、无法指定目标或者用途、忽视文档化、无法缓冲对其他人的冲击、缺少技术支持、缺少开发经验以及技术或投入的效力问题等问题。

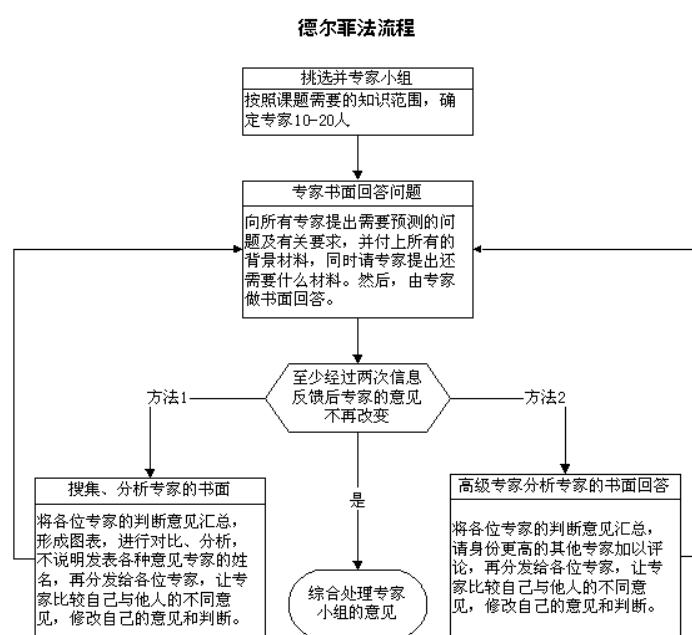


图 3.2 Delphi 流程

Fig3.2 Delphi flow chart

为了验证这次调查的真实度，参考和使用了数据分析中的信度概念，采用 Cronbach 提出的用 α 系数来验证反馈数据的真实可靠性（Cronbach 的 α 系数是目前最常用的信度系数）。

$$\alpha = \frac{k}{k-1} \left(1 - \frac{\sum_{i=1}^k \sigma_i^2}{\sigma_i^2} \right)$$

其中，K 表示题目的总数，这里为 41 道题， σ_i^2 为第 i 题得分的标准方差（标准方差的计算公式是：每一个数与这个数列的平均值的差的平方和，除以这个数列的项数，再开根号）。最后计算可得，调查反馈的 α 系数为 0.9437。这表明这次调查具有较高的可信度，从而也证明了对影响软件开发的危险因素和因素具有一定的普遍性。

3.4 小结

本章主要结合了软件开发生命周期详细地分析了影响软件开发的危险因素，同时对这些危险因素按照类别、属性二个层次结构进行分类，在这些工作的基础上对软件企业和开发人员进行风险管理调查，得出了常见的危险因素，最后利用了 Cronbach 提出的 α 可信度系数进行调查可信度验证，得出了分析具有较高的可靠性结论。

4 软件风险分析方法及策略设计

风险评估和分析, 又称风险预测, 常采用两种方法估价每种风险。一种是估计风险发生的可能性或概率, 另一种是估计如果风险发生时所产生的后果。一般来讲, 软件风险管理者要与软件项目计划人员、技术人员及其他管理人员一起执行四种风险活动:

- (1) 建立一个标准(尺度), 以反映风险发生的可能性。
- (2) 描述风险的后果。
- (3) 估计风险对项目和产品的影响。
- (4) 确定风险的精确度, 以免产生误解。

另外, 要对每个风险的表现、范围、时间做出尽量准确的判断。对不同类型的风险采取不同的分析办法。

4.1 定性分析方法

定性分析的目的是界定风险源, 并初步判明风险的严重程度, 以给出系统风险的综合印象。初步危险分析是用于识别系统中可能存在的风险源, 而以下的几种方法则是用于定性地量化各种风险源可能对系统造成的破坏, 从而判明系统风险大小^[18]。

定性风险评价主要包括风险评估指数法RAC(Risk Assessment Code)、总风险暴露指数法TREC(Total Risk Exposure Code)、直接风险评估法SCRAM(Short-Cut Risk Assessment Method)等。其主要特点是对危险的可能性和严重程度不做精确的数值分析, 而是大致将其划分一些等级, 然后把这两者用不同的方式综合起来衡量风险的大小及重要程度, 并按大小和重要程度进行分类排序, 最后分别对不同种类的风险采取不同的措施, 确定是否接受风险还是改进设计、改善材料工艺、改变工作流程、加强人员培训等方法 and 措施以减小风险。在软件风险定性分析方法当中, 人们常用因果分析法:

因果关系分析法用于揭示影响及其原因之间的联系, 以便追根溯源, 找出风险的根本原因。其目的是通过确定问题的根本原因来阻止问题的发生。在因果关系分析法中, 包含鱼骨图法和 5 个为什么的方法因果关系分析法的哲学是, 如果一个错误发生了, 除非及时采取措施阻止, 否则它将再发生。通过吸取过去的教训, 可防患于未然。因此, 应该将评估自己的错误视为软件系统生产过程的一部分。因果关系分析法是一个简单的三步过程:

- 1) 确定错误原因;

2) 确定阻止未来错误的行动;

3) 实施校正行动。

定性分析方法主要是通过分析需求分析阶段、设计阶段、编码阶段、集成和测试阶段常见的风险因素,如在软件需求分析阶段有:错误的理解需求;开发错误的软件;系统设计困难;实现的系统与客户的需求不一致;测试困难;理解错误;系统不稳定;编码困难;无法在规定的时间内完成工作;花费的时间长;增加成本等风险因素。按照其发生的概率、对本阶段目标的影响力、对其它阶段目标的影响力,可以归纳出该风险相对软件开发的综合影响力。主要评定标准是从出现的概率的角度分为:很高、高、一般、低、很低;从对软件项目的影响力角度来讲分为:很大、大、一般、小、很小。

当然,进行定性分析的尺度有很多,可以根据不同情况,在实际工作中对不同的风险因素进行针对性不同的分析,比如设计阶段的设计复杂这一风险因素,可以用设计范围、设计难度、设计层次等各种尺度来衡量设计复杂程度。

优点:对风险因素的发生概率、对不同阶段的影响力等因素的尺度估计比较容易获得,通常取之于专家或者有关人员根据经验、实际情况或者有关资料。而且标准制定比较灵活。

不足:标准尺度范围比较狭窄,很难用文字准确的表达。有时也很难对某些具体的特性对总体的影响力进行准确的估计。

4.2 定量分析方法

定量风险分析过程的目标是量化分析每一风险的概率及其对项目目标造成的后果,也分析项目总体的风险程度。这一过程涉及的工作包括测定取得某一特定项目目标的概率;量化项目的风险暴露,决定可能需要的成本大小和进度计划应急准备金;通过量化各风险对项目风险的相应贡献,分析出最需要关注的风险;找出理想的和可实现的成本、进度计划及工作范围目标。定量风险评价比定性方法更精细,也更复杂,形成定量风险分析结果有多种途径。定量风险评价比定性方法更需要有可靠数据的支持,否则其优越性会无法发挥确定型风险估计

4.2.1 确定型分析

①盈亏平衡分析

盈亏平衡分析(Break-Even Analysis)通常又称为量本利分析或损益平衡分析。它是根据软件项目在正常生产年份的产品产量或销售量、成本费用、产品销售单价和销售税金等数据,计算和分析产量、成本和盈利这三者之间的关系,从中找出它们的规律,并确定项目成本和收益相等时的盈亏平衡点的一种分析方法。

法^[21]。在盈亏平衡点上，软件项目既无盈利，也无亏损。通过盈亏平衡分析可以看出软件项目对市场需求变化的适应能力。

②敏感度分析

敏感度分析法的目的是考察与软件项目有关的一个或多个主要因素发生变化时对该项目投资价值指标的影响程度。通过敏感度分析，可以了解和掌握在软件项目经济分析中由于某些参数估算的错误或是使用的数据不太可靠而可能造成的对投资价值指标的影响程度，有助于确定在项目投资决策过程中需要重点调查研究和分析测算的因素。它是通过将每一个输入变量设为最大值(其他变量保持正常值)来帮助确定模型对输入变量变化的敏感程度。对决策有影响的变量较为重要其他变量则相对次要对变化不敏感的变量设为正常值，将其作为已知变量而不是不确定的变量进行处理。敏感度分析法把注意力集中在最重要的变量上有重要意义，并有助于按优先级进行数据收集。敏感度分析法有两种有用的工具，分别是龙卷风图和效力函数^[22]。

“龙卷风图”首先展示最敏感的变量。（如图 4.1）通过绘制每个变量的范围而形成的图形确实类似于龙卷风。最敏感的变量在最顶部，最不敏感的变量在最底部。绘制一幅龙卷风图所需的数据是一些变量及其可能的数值范围。每个变量的最高和最低值确定其可能的影响力。对任何指定的变量，本图表中条形的长度都代表利润对该变量的敏感程度。

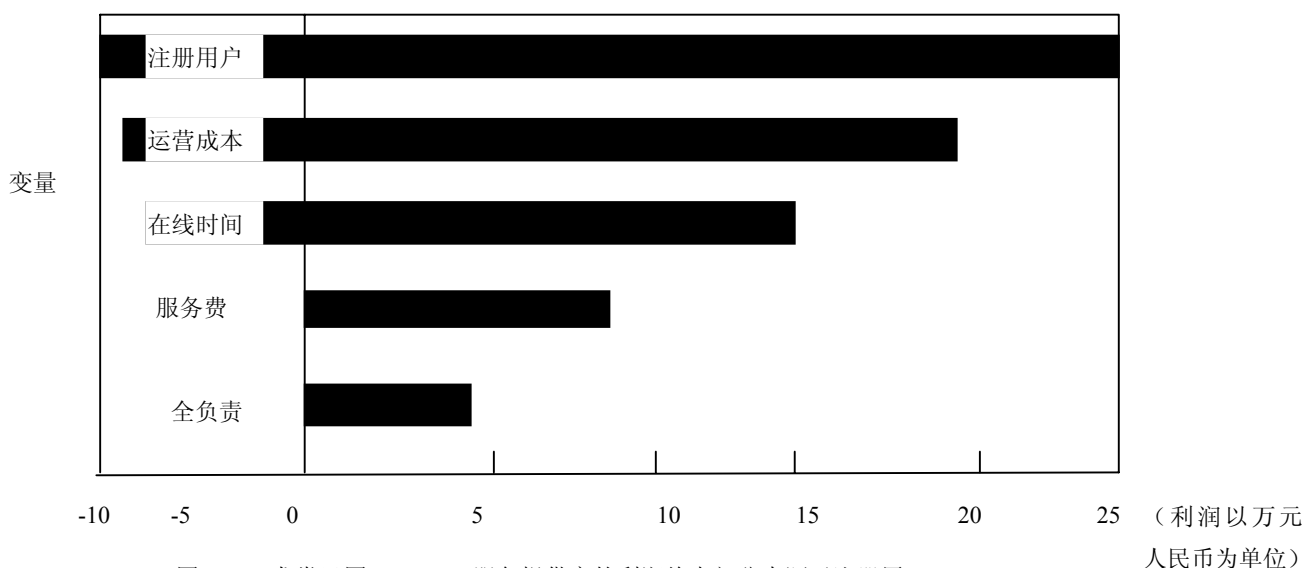


图 4.1 龙卷风图 (Internet 服务提供商的利润绝大部分来源于注册用

Fig 4.1 Tornado diagram (The most profits of Internet serves provide company in the registration consumer)

“效力函数”结合决策者的风险态度将期望效力而非期望价值最大化。效力函数代表个人对风险的态度。对待风险，不同的人有不同的反应，接受风险的程度有所不同。有些人倾向于乐于接受风险，另一些人则趋于保守，尽量避免风险。

指数形式的效力函数取一个被称为风险忍受程度的值作为参数，它决定效力函数反对风险的程度。风险忍耐程度的值越大，意味着个人越能承受和接受风险。进行敏感度分析可以了解风险偏好，改变风险忍耐程度则可确定决策会在哪能一点发生变化。

③概率分析

它是运用概率论及数理统计方法，来预测和研究各种不确定因素对软件项目投资价值指标影响的一种定量分析。通过概率分析可以对项目的风险情况做出比较准确的判断。主要包括解析法和模拟法（蒙特卡罗 Monte Carlo 技术）两种。

4.2.2 不确定型风险估计

主要有小中取大原则、大中取小原则、遗憾原则、最大数学期望原则、最大可能原则。

4.2.3 随机型风险估计

主要有最大可能原则、最大数学期望原则、最大效用数学期望原则、贝叶斯后验概率法等。

在确定型分析、不确定型风险估计和不确定型风险估计这三种风险估计中，人们往往是把一些方法结合起来交叉使用，以便取得更为精确的评判，从而达到控制风险的目的。一般来说，对项目进行风险评估和分析的软件风险定量分析方法的常用方法还有很多，如 Monte Carlo 模拟法，计划评审技术 PERT (Program Evaluation and Review Techniques)，主观概率法 (Subjective Probability Method)，效用理论 (Utility Theory)，灰色系统理论 (Grey System Theory) 故障树分析法 FTA (Fault Tree Analysis) 外推法 (Extrapolation)，模糊分析方法 (Fuzzy Analysis)，影响图分析法 (Influence Diagram)，概率风险评价 PRA (Probabilistic Risk Assessment)、GO-FLow 法、火灾爆炸指数法 FEI (Fire Explosion Index) 等^[20]。

①PRA 和 DPRA 分析法

PRA 和 DPRA 都是在 FTA 分析基础上的量化，在可靠性及运行系统风险分析领域内应用广泛。稍作改造，我们便可将其运用到项目风险分析领域。其分析步骤如下：

- 1) 别项目研制过程中的困难环节，找出风险源；
- 2) 对各风险源考察其在项目研制中的地位，及相互逻辑关系，给出项目的风险源树；
- 3) 标识各风险源后果大小，及风险概率；
- 4) 对风险源通过逻辑及数学方法进行组合，最后得到系统风险的度量。如果是用 DPRA 进行评估，则尚须考虑它们在时间上的关系。

②VERT 分析法

另一种被广泛运用于风险评估的方法是 VERT。VERT 是国外在八十年代初期发展的一通用仿真软件，它对项目研制构造过程网络，将各种复杂的逻辑关系抽象为时间、费用、性能的三元组的变化。网络模型面向决策，统筹处理时间、费用、性能等风险关键性参数，有效地解决多目标最优化问题，具有较大的实用价值。它的原理是通过丰富的节点逻辑功能，控制一定的时间流、费用流和性能流流向相应的活动。每次仿真运行，通过蒙特卡洛模拟，这些参数流在网络中按概率随机流向不同的部分，经历不同的活动而产生不同的变化，最后至某一终止状态。用户多次仿真后，通过节点收集到的各参数了解系统情况以辅助决策。如果网络结构合理，逻辑关系及数学关系正确，且数据准确，我们可以较好地模拟实际系统研制时间、费用及性能的分布，从而知道系统研制的风险。

③故障树分析法(FTA Fault tree analysis)

故障树分析法是美国贝尔电话实验室的维森于 1962 年首先提出的，我国于 1976 年开始进行介绍和研究这种方法，并随之应用于许多项目中，取得了不少成果。FTA 是一种具有广阔应用范围和发展前途的分析方法。故障树是由一些节点及它们间的连线所组成的，每个节点表示某一具体事件，而连线则表示事件之间的关系，FTA 是一种演绎的逻辑推理分析方法，遵循从结果找原因的原则，分析项目风险及其产生原因之间的因果关系，即在前期预测和识别各种潜在风险因素的基础上，运用逻辑推理的方法，沿着风险产生的路径，求出风险发生的概率，并能提供各种控制风险因素的方案。FTA 具有应用广泛、逻辑性强、形象化等特点，其分析结果具有系统性、准确性和预测性。

④风险决策树分析

风险决策树分析是一种演绎的风险分析法。决策树是一种图表，它反映了尚在考虑中的一项决策，以及选择一个方案和两个被选方案中的另一个的暗示。决策树将风险概率、事件的每一条合理路径的成本和报酬、以及未来的决策综合在一起。一般用来专门研究项目开发过程中特定的不希望发生的事件。通过分析造成风险事件的各种可能因素（如：硬件、软件、环境、人为因素等）及因果关系，画出逻辑关系图（即风险决策树）。

在对软件风险进行充分的数学分析之后，可以利用风险决策树进行风险分析。风险决策树分析的步骤和顺序如下：

- 1) 熟悉和定义软件项目，采用 WBS（工作结构分析）分析项目任务；
- 2) 绘制开发软件的系统功能图 and 安全性框图；
- 3) 确定软件所有的潜在风险事件，并确定这些事件对软件项目相关功能的影响；

- 4) 按最坏的潜在后果评估每一种风险事件, 确定其严重性;
- 5) 确定每一种风险事件发生的概率;
- 6) 确定每一种风险事件发生的频数比;
- 7) 分析风险事件的危害度;
- 8) 分析项目的风险危害程度或风险损失

⑤外推法(Extrapolation)

外推法是进行项目风险评估和分析一种十分有效的方法, 它可分为前推、后推和旁推三种类型。前推就是根据历史的经验和数据推断出未来事件发生的概率及其后果。如果历史数据具有明显的周期性, 就可据此直接对风险作出周期性的评估和分析, 如果从历史记录中看不出明显的周期性, 就可用一曲线或分布函数来拟合这些数据再进行外推, 此外还得注意历史数据的不完整性和主观性。

⑥后推法

后推法是在手头没有历史数据可供使用时所采用的一种方法, 由于工程项目的一次性和不可重复性, 所以在项目风险评估和分析时常用后推法。后推是把未知的想象的事件及后果与一已知事件与后果联系起来, 把未来风险事件归结到有数据可查的造成这一风险事件的初始事件上, 从而对风险做出评估和分析。旁推法就是利用类似项目的数据进行外推, 用某一项目的历史记录对新的类似项目可能遇到的风险进行评估和分析, 当然这还得充分考虑新环境的各种变化。这三种外推法在项目风险评估和分析中都得到了广泛的采用。

这些定量分析方法都有其独到之处, 结合起来使用能够有效地帮助软件开发人员识别、控制风险。总的来说, 软件定量分析也有其优缺点。

优点: 定量分析可以将风险因素发生的概率、对本阶段目标的影响力、对其它阶段目标的影响力和对软件开发的综合影响力得以量化, 从而大大提高了风险分析、风险排序以及之后的风险解决和控制的准确性和有效性。

不足: 对风险发生的概率等几个因素的估计很难, 而且这些估计中也是存在大量的不确定性的, 因为这些估计通常是由专家或者有关人员通过其他资料而进行主观的估计得来的。为了尽量减少这些估计的不确定性, 可以尽量使用与实际相符的信息, 努力提高数据的客观正确性和有效性。

4.3 软件风险定量、定性分析的策略研究

软件的开发过程可以划分为不同的阶段, 如系统分析阶段、系统设计阶段、系统实施阶段和系统维护阶段^[25], 各个阶段所面临的风险是不相同的, 即使是同样的风险在不同阶段发生时其影响程度及后果也是不同的, 同时每一个阶段不同人员(用户、开发人员、系统网络及软件供应商、风险管理及决策人员)对这一阶

段面临的风险认识程度也不相同, 因此对于风险发生时所产生后果的估测既要考虑风险所处的阶段性, 同时也要考虑对识别风险的对象的认识程度, 软件开发中风险所带来的后果不仅是经济损失, 更重要的是风险发生后系统的开发工作能否继续进行。因此不能简单的用经济损失来反映风险发生时所产生的后果。

软件开发过程中, 风险事件发生时影响程度的大小, 需要从这种风险事件发生时所导致的软件系统的技术性能的下降、系统建设费用的上升和系统建设工期的延长, 即技术性能、费用、进度这三个方面来综合衡量, 由于这种估计的不确定性因素很大^[26]。

在软件开发过程中, 不管是使用定量、定性分析方法, 一般都要进行以下几个步骤:

- (1) 研究熟悉系统;
- (2) 分析初始事件和中间事件;
- (3) 对初始事件和中间事件发生的概率进行评估;
- (4) 量化和不确定性分析; 为了得到某个终态的概率以及将不同的初始事件、中间事件和系统响应对风险的贡献大小进行排序, 风险模型要进行量化的评估。在建模的假设和输入数据中, 存在着技术和统计上的不确定性, 这种不确定性最终会传播到事件序列的终态事件, 因此需要对最后的结果进行不确定性分析。
- (5) 后果分析;
- (6) 风险排序和管理;

4.3.1 风险因素发生的概率

风险概率指的是在风险声明中的风险结果部分提到的事态实际发生的可能性。使用风险概率的数值对风险分级有着十分重要的意义。风险概率必须大于 0, 否则风险就不可能造成威胁^[23]。以此类推, 风险概率也必须小于 100%, 否则风险就是确定的事情了。尽管行业或企业风险数据库基于大量项目的实例, 可能对提供已知概率估计值有用, 但对于个体来说, 估算和应用概率是相当困难的。不过, 大部分项目团队可以描述他们以往的经验, 产生行业报告, 并提供自然语言术语来映射数字概率范围。这可能和将“低-中-高”映射为不连续的概率数值(17%, 50%, 84%)一样简单^[27], 也可能和映射不同的自然语言术语一样复杂, 例如“不一定高”、“不可能”、“很可能”、“几乎肯定”等, 这些术语描述了概率中的不确定性。下面的示范表格 9 列出了一个三段概率分级的例子。紧随其后的示范表格则列出了一个七段概率分级的例子。(如表 4.2)

表 4.2 概率分级
Tab4.2 probability classification

概率范围	用来计算的概率值	自然语言表达	数字得分
1%至 33%	17%	低	1
34%至 67%	50%	中	2
68%至 99%	84%	高	3
概率范围	用来计算的概率值	自然语言表达	数字分数
1%至 14%	7%	非常不可能	1
15%至 28%	21%	低	2
28%至 42%	35%	可能不	3
43%至 57%	50%	一半	4
58%至 72%	65%	可能	5
73%至 86%	79%	非常可能	6
87%至 99%	93%	几乎肯定	7

在软件开发进行过程中,各风险因素对软件开发的综合影响力是取决于风险因素发生的概率、风险因素对本阶段目标的影响力和风险因素对其它阶段目标的影响力等各种因素的共同效应。但是,在实际工作中,软件开发人员通常是关注那些发生概率很高或者对各阶段目标影响力很大的风险因素,而忽略了对整个项目结果起决定作用的是几个因素共同发挥作用的。例如,对需求的变动无法做出及时的反应以及设计复杂这两个风险因素发生的概率都为3,对本阶段目标的影响力都为4,而因为他们对其它阶段目标的影响力不同而导致综合影响力相差很大。因此,应该在注意各个因素的基础上,将主要精力放在风险因素对软件开发的综合影响力上。

通过风险识别过程所识别出的潜在风险数量很多,但这些潜在的风险对项目的影响是各不相同的。风险分析即通过分析、比较、评估等各种方式,对确定各风险的重要性,对风险排序并评估其对项目可能后果,从而使项目实施人员可以将主要精力集中于为数不多的主要风险上,从而使项目的整体风险得到有效的控制。确定了项目的风险列表之后,接下来就可以进行风险的定量化分析了^[28]。风险定量分析的目的是确定每个风险对项目的影响大小,从两个角度分析:风险出现的概率和风险影响的程度,风险值=风险概率×风险影响。

风险概率:它是风险发生可能性的百分比表示,是一种主观判断。(如下表 4.3)

表 4.3 风险定量分析：发生的概率

Tab 4.3 Risk quantitative analysis: happen rate

级别	数值	判断标准
几乎肯定	1	任何与之相关的都会成为问题
很有可能	2	这样的风险转换成问题的机会很高
可能(对半)	3	这样的风险转换成问题的可能对半
不大可能	4	这样的风险偶尔会成为问题
极不可能	5	这个事件几乎不可能成为问题

测量风险影响项目的程度：它是指一旦风险发生可能对项目造成的影响大小。如果损失的大小不容易直接估计，可以将损失分解为更小部分再评估它们（如下表 4.4）。

表 4.4 风险定量分析：影响的程度

Tab 4.4 Risk quantitative analysis: The influence's degree

级别	数值	判断标准
灾难性的	5	导致项目失败、中途夭折、无法满足客户的关键要求
严重的	4	影响关键任务的完成、严重质量缺陷、进度推延、超支等
一般程度	3	导致可以接受的质量缺陷、进度推延、范围缩小等
微小的	2	导致项目的进度、成本、质量等有微小的损失
可忽略的	1	对项目产生的影响可以忽略不计

风险影响可用相对数值表示，损失大小折算成对计划影响的时间、成本或者工作量表示^[29]。例如：某一风险概率是 25%，一旦发生会导致项目计划延长 4 周，因而，风险值=25%×4 周=1 周。下面是对 XX 公司 ERP 项目计划书（2000 年）中预警风险的定量分析，其中对风险的影响程度用工作时间周来表示的（如表 4.5）。

表 4.5 xx 公司 ERP 项目计划书中预警风险的定量分析图

Tab 4.5 The quantitative analysis diagram of the early-warning risk in xx company ERP item plan

风险标识	风险的含义	风险值
人员参与	XX 公司管理层和业务人员的支持和积极参与的程度	26 周
模式匹配	XX 公司的管理、组织、业务流程的模式与 ERP 应用系统不一致	13 周
变更风险	项目的实施目标、XX 公司管理与经营结构、项目成员等的变动	31 周
合作风险	各个部门人员的协调与合作，项目成员的管理	36 周
数据风险	数据的完整性和准确性	2 周
技术风险	技术：项目组对系统掌握的程度，ERP 公司的支持	41 周

风险影响及概率从管理的角度来考虑，是起着不同作用的（见图 4.6）。

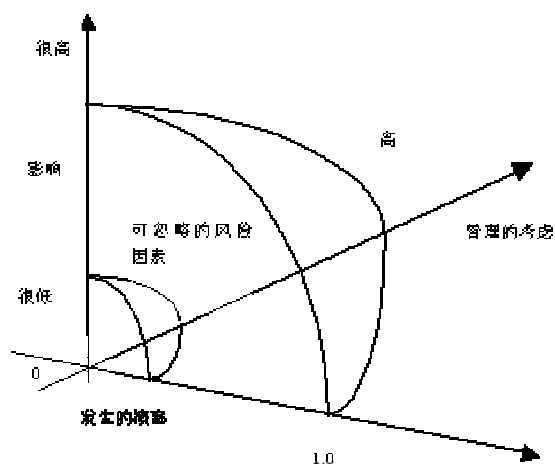


图 4.6 风险影响及概率对风险管理的作用示例图

Fig 4.6 risk influences and rate function example to risk management

一个具有高影响但发生概率很低的风险因素不应该花费太多的管理时间。而高影响且发生概率为中到高的风险以及低影响但高概率的风险，应该首先考虑如果风险真的发生了，所产生的后果有三个因素可能会受影响：风险的性质、范围、时间^[30]。风险的性质是指当风险发生时可能产生的问题。例如，一个定义得很差的与客户硬件的接口（技术风险）会妨碍早期的设计和测试，也有可能项目后期阶段的系统集成问题。风险的范围结合了严重性及其整体分布情况。风险的时间主要考虑何时能够感到风险，风险会持续多长时间。在大多数情况下，项目管理者希望“坏消息”越早出现越好^[24]。

以下的步骤用来确定风险的整体影响：

- 确定每个风险元素发生的平均概率。
- 使用前面的表格，基于其中列出的标准来确定每个因素的影响。
- 完成风险表，分析其结果。

风险预测和分析技术可以在软件项目进展过程中迭代使用。项目组定期复查风险表，再评估每一个风险，以确定新的情况是否引起其概率及影响的改变。在风险分析过程中，经常使用三元组 $[R_i, P_i, E_i]$ 描述风险。

其中 R_i 代表风险， P_i 表示风险发生的概率， E_i 是风险带来的影响， $i = 1, 2, \dots$ P 是风险序号，表示软件项目共有 P 种风险。软件开发过程中，由于项目超支、进度拖延和软件性能下降都会导致软件项目的终止，因此多数软件项目的风险分析都需要给出成本、进度和性能三种典型的风险参考量。当软件项目的风险参考量达到或超过某一临界点时，软件项目将被迫终止^[31]（如图4.7）。

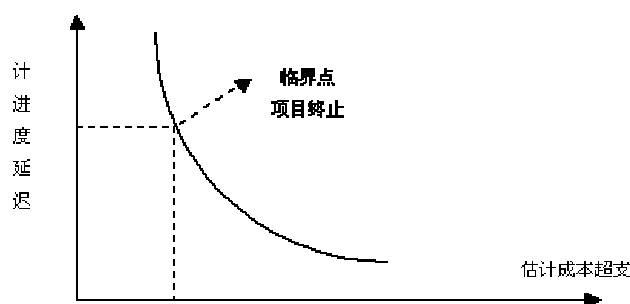


图 4.7 成本、性能和进度三元风险参照系

Fig 4.7 three department of Cost, performance and progress of risk

在软件开发过程中，成本、进度、性能是相互关联的。例如，项目投入成本的增长应与进度相匹配，当项目投入的成本与项目拖延的时间超过某一临界点时，项目也应该终止进行。

通常风险估算过程可分为四步：

- 1) 定义项目的风险参考量；
- 2) 定义每种风险的三元组 $[R_i, P_i, E_i]$ ；
- 3) 定义项目被迫终止的临界点；
- 4) 预测几种风险组合对参考量的综合影响。

三元组 $[R_i, P_i, E_i]$ 是风险管理的基础。设高级职员流动给项目带来的风险为 R_1 。根据历史的经验或直观感觉，高级职员离开课题组的概率： $P_i = 70\%$ 。这一事件的出现带来的影响 E_i 是项目开发时间延长15%，项目成本增加20%。在这种情况下，就相应地对这种风险进行相应的防范和采取必要的措施及时化解和消除这种风险。

进行风险评价时,可根据风险估计的结果,建立一个系统三元组 $[r_i, p_i, e_i]$,其中 r_i 表示风险, p_i 表示风险出现的可能性, e_i 表示风险的影响, $i=1, 2, \dots, m$ 。在进行风险分析时,尽量按以下步骤执行:

(1) 定义项目的水平参照值;(定义水平参照值时,主要是结合专家的知识库和软件开发人员的经验值,以及软件企业的风险数据库中的历史数据来确定参照值)

(2) 找出每组 $[r_i, p_i, e_i]$ 与每个水平参照值间的关系;

(3) 估计一组临界点以定义项目的终止区域;

(4) 估计风险组合将如何影响风险水平参照值。

参照第三章软件风险分类的结构,结合这些系统三元组 $[r_i, p_i, e_i]$,建立一个风险分析的控制矩阵。通过不同情况下风险概率的计算和分析,从定量的角度来考察和实施风险控制与管理。

$$R = \begin{bmatrix} r_1 & p_1 & e_1 \\ r_2 & p_2 & e_2 \\ M & M & M \\ r_i & p_i & e_i \end{bmatrix}$$

4.3.2 集合风险

集合风险指对软件开发过程中各种风险事件的综合评估,定义为:

$$RISK = \sum_{i=1}^m (w(r_i) \times p(r_i))$$

r ($i=1, 2, 3, \dots, m$) 表示风险事件, m 为风险事件的个数;

$W(r_i)$ 表示风险事件的权重;它的取值方法通常有最佳贴近度取值法、集值统计取值法和专家评定法三种。最佳贴近度取值法是依据以前所开发的软件的风险管理过程中的各种风险因素估测结果,从中找出与在建软件相类似的一个,将其数值赋给 $W(r_i)$;集值统计的取值法利用集值统计迭代方法确定; $W(r_i)$ 专家评定法是由风险管理及决策人员根据软件开发过程中的风险管理经验,确定各阶段各层次的风险估测模型中各权向量的数值,这种方法比较简单。

$P(r_i)$ 表示风险事件发生的概率。 $P(r_i)$ 反映了在软件开发的不同阶段,由于各个风险因素在软件项目开发的过程中出现的概率不同,其数值通常由风险管理及决策人员根据以往的软件开发过程的风险管理经验和在建软件的实际情况来确定。

通过对软件项目集合分析和评估,可以更加精确地掌握项目所存在的潜在风险、经济损失期望值,从而为选择控制方式、应急方案等决策提供有力的支撑。

4.3.3 评估软件风险的策略

软件项目周期长、规模大、涉及范围广、风险因素数量多且种类繁多致使其在全寿命周期内面临的风险多种多样。而且大量风险因素之间的内在关系错综复杂、各风险因素之间并与外界交叉影响又使风险显示出多层次性，所以在进行风险评估与决策一定要根据潜在风险发生的概率和危害程度确定风险能否被接受，是否需要采取对应措施进行防范。

在对软件风险起先量化分析过程中，由于在其生命周期内跨越的时间长，并且变化因素多，针对这一特点就可以结合软件的生命周期，从建立项目、需求分析、编码、测试、维护等多个阶段的特点，进行动态地分析、跟踪、规避软件开发过程中随时可能出现和发生变化的风险因素，实施持续地风险管理。一般来说，首先要做好风险识别，找出项目在各阶段的风险因素，然后结合项目组的情况，利用一些历史数据和专家知识库，对这些风险因素进行评价，进行通过对风险的可能等级和严重性等的定性分析得到可比较的风险评价，或通过风险事件发生的权重和概率的定量分析得到风险状态的指数，再利用风险决策树方法，综合各个阶段的风险影响程度，计算出项目总的风险影响程度，进行风险决策，进而对风险作出正确的评价，最后制定出一些强有力的风险防范措施对风险进行控制（如图 4.8）。。

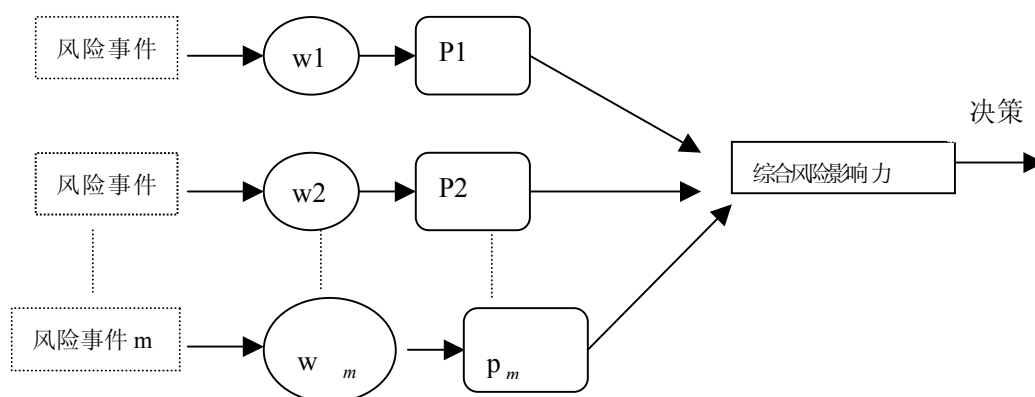


图 4.8 软件风险影响决策

Fig 4.8 software risk influence decision tree

4.4 小结

本章主要工作是对软件风险定量和定性分析方法进行了详细的分析，比较了两种不同方法的优缺点，较为详尽地讲述了常见的定量和定性分析方法，同时结合定量和定性分析方法进行研究，对风险出现的概率进行了深入的探讨，提出利用风险决策树的对策对软件风险进行控制和决策。

5 软件风险分析原型系统的设计

5.1 功能介绍

系统功能描述如下:项目风险评估原型系统是利用 MISREM 估算项目风险程序的工具它主要利用对一些项目风险指标的评估分数作为输入参数,按照风险计算规则估算项目的风险程序.该系统的主要使用者是项目管理工作人员.系统的使用信赖于风险评估模型库,该数据保存了需要考虑的风险指标、指标相关的问题、指标对风险的影响程度等。系统还将对项目指标的评分作为历史记录保存在数据库中,以备查阅。特别地,项目管理人员可以通过修改其中一些题目的答案重新计算项目风险,可以依据新的风险数值决定减低风险的策略。因为修改题目的答案即针对题目内容采取措施进行一些弥补。此外,系统维护人员还可以对风险评估模型库进行更新维护。

将系统需求用 UML 中的用例图表示(如图 5.1)其中有:

1、执行者

“系统维护人员”:主要维护管理并保持系统的正常运行。

“项目经理”(项目过程管理人员):负责通过回答项目情况的调查问卷,对要求系统对项目的当前风险进行评估进而查阅项目风险评估结果。

“风险评估数据库”:相当于一个外部系统。记录了风险评估模型所需的指标及其之间的关系,还用来存储问卷答复和风险评估的历史记录。

2、维护模型

“修改模型”用例:主要是对风险评估数据库进行管理,需要使用到“修改指标与权重”和“修改指标与问卷”两个用例。这是因为对模型的修改具体包括:更新风险指标,更新风险指标和问卷之间的对应关系、更新风险指标在评估模型中的权重信息等。其中有权重的修改也可以反映项目经理对个别风险指标的侧重点。

3、评估风险

“建立项目”用例:对于新的项目,系统要求项目经理先设置新的项目的有关信息,例如项目名、开始时间、预计结束时间、项目规模、投入人力等。这些数据一方面可以作为项目管理的基础信息,另一方面也有助于收集更多的信息对风险评估系统进行优化。对于已经用系统进行风险分析过的项目,则导入该项目,供项目经理查看或者修改。

“调查项目现状”用例:从风险评估模型创建用例子中读出问卷内容,展示给项目经理进行回答。问卷内容即调查风险评估时需要了解的项目现状。项目经

理完成问卷后，将结果存储在数据库的历史数据中。

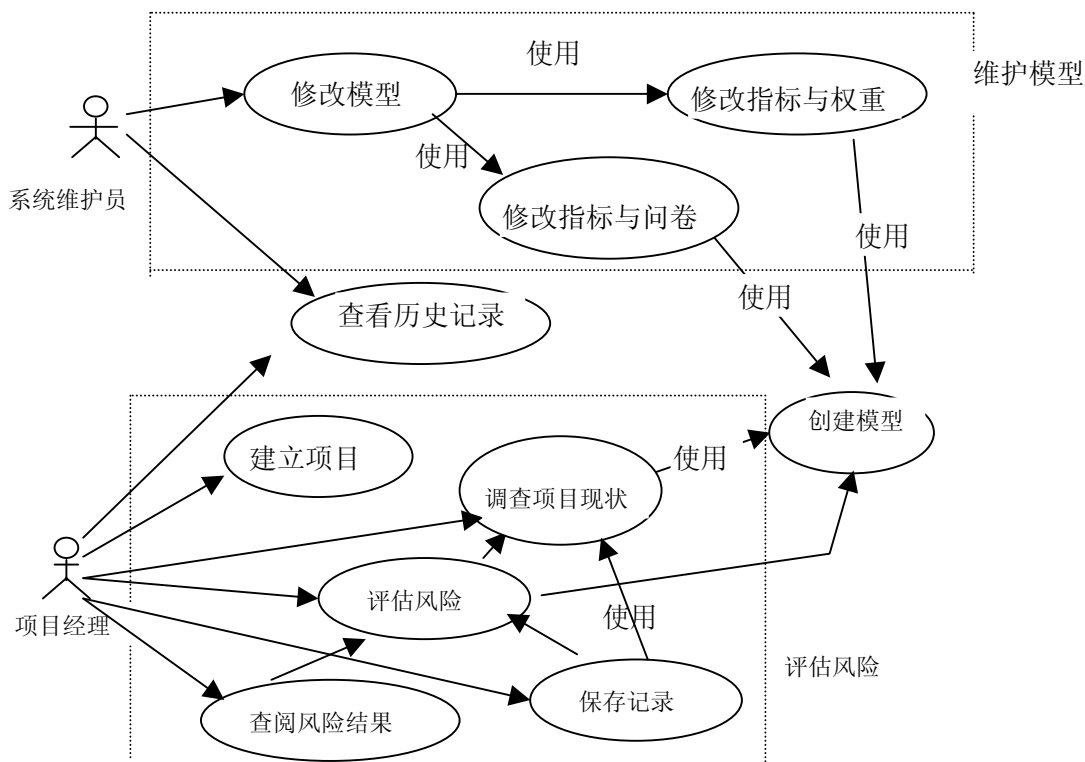


图 5.1 原型系统需求的 UML 用例图

Fig 5.1 UML example diagram of prototype systems

“评估风险”用例子：使用“调查项目现状”用例子的结果，针对项目经理对风险指标的评分，根据“创建模型”用例子的风险评估模型信息，根据本章公式(1)到(3)估算项目风险。

“查阅风险”用例：使用“评估风险”用例子的结果，将项目风险评估后的结果展示给项目经理看。项目经理可以选择查看总体风险和各个分层的风险等。

“保存记录”用例：将项目建立时的基本信息项目现状的问卷反馈以及评估的结果保存在数据库中。

4、其他用例

“查看历史记录”：项目经理要求查看以往相关项目的资料，以便从中取得借鉴。系统维护人员查看历史资料则是为了改进系统，使系统更好的运作。

“创建模型”：从数据库中读取记录，建立软件风险分析模型。即产生出模型的问卷，建立问卷对应指标及其权重的关系。被“修改指标与权重”、“修改指标与问卷”、“调查问卷现状”、“评估风险”等用例子使用，相当于功能用例子部分与数据库的接口。

由于用例包“评估风险”是系统是最重要的，本文特地将其实细化，得到 UML

顺序图（如图所示）：

1-3 首先项目管理人员启动系统，申请创建一个新的项目或者打开一个已存在的项目。并暂时保存项目的基本信息。

4-7 项目创建成功后，项目管理人员申请进行项目现状评估，问卷产生器向模型创建器要求问卷，模型创建器从数据库中读取风险指标及其相应问题，返回给问卷产生器生成一份问卷，展示给项目管理人员。

8-10 项目管理人员依次对风险指标评分，最后提交问卷反馈。问卷产生器汇总所有结果，向项目管理人员确认填写是否正确。经过项目管理人员确认后，用记录保存器暂时保存问卷反馈。

11-14 问卷产生器将经过确认的问卷反馈传递给风险评估器，风险评估器通过模型创建器获得所需要的权重等信息，根据软件分析计算公式计算每个风险总值和多个子风险值。并利用记录保存器将评估结果暂时保存。

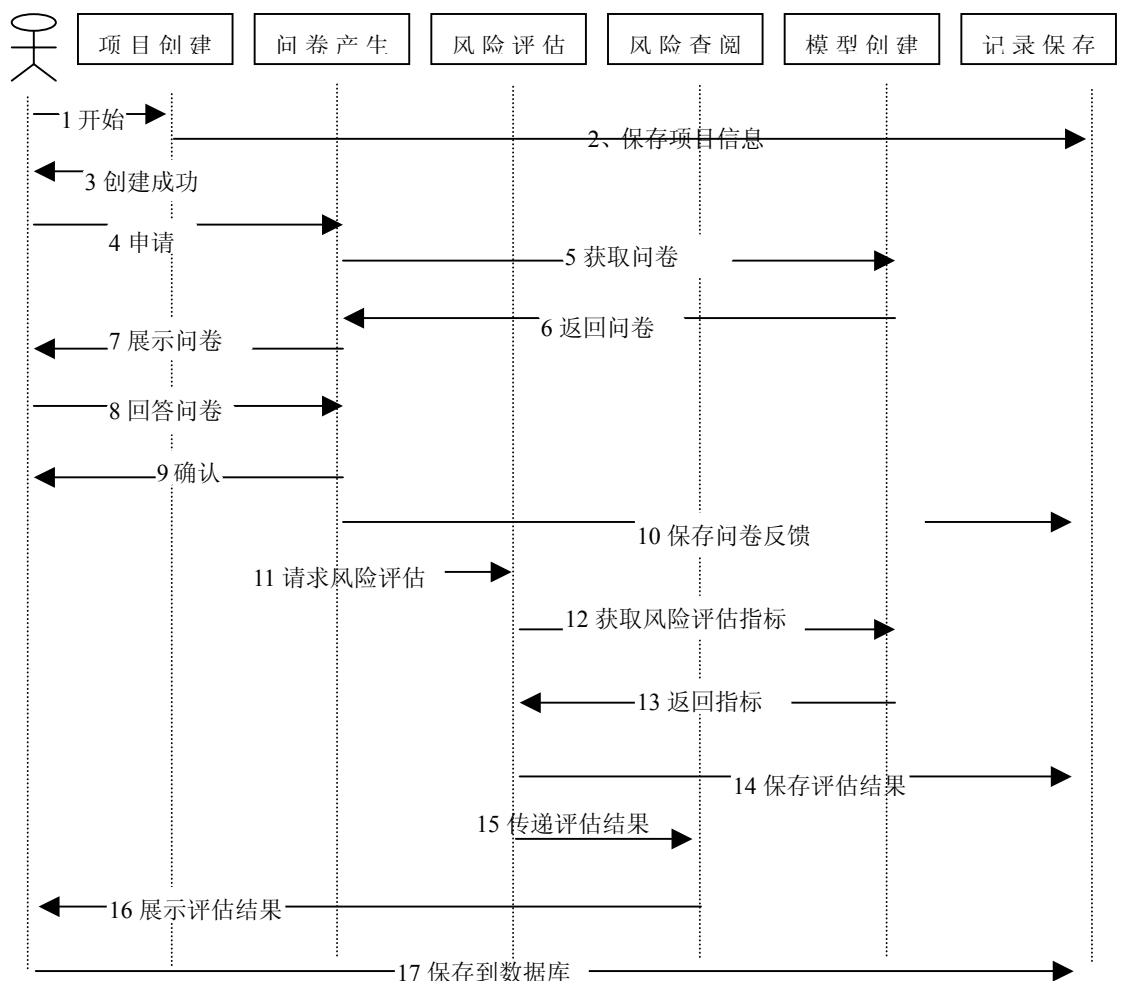


图 5.2 软件风险原型系统评估风险用例包顺序图

Fig 5.2 Order diagram of prototype systems

15-16 风险评估器将评估结果通过风险查阅器展示给项目经理看。项目经理不仅可以从风险总值对项目的状态有个总体的了解，还可以从中进一步地查看风险分布，以了解技术、进度和成本哪一种风险更为严重，甚至了解哪一类风险因素值得引起注意。

17 项目管理人员可以最后决定是否在数据库中保存此次风险评估过程中的数据。

5.2 系统设计

系统是一个在 PC 机上运行的单机版本，由于其功能主要是问卷产生和数学计算，所以软件技术要求不高，可以选择 Windows 2000 作为操作系统，以 PowerBuilder 等作为开发语言，以 PowerDesigner 等软件作为设计工具，创建开发环境。在数据库管理系统的选择上，因为数据库主要是存储风险问题、风险因素及其之间的关系，还要存储在风险评估时输入数据的历史记录，所以数据库结构相对比较简单，数据量也比较少。于是就选择以 Sybase 作为系统的数据库，这主要是考虑到其 Sybase 良好的性能。

进行类的分析主要是标识出每个类的所有属性和在相关用例中的职责，以及与其它类之间的关系。由于文章篇幅的问题，本文只是给出原型系统中主要的几个实体类及其主要的方法（如下图 5.3 和图 5.4）。

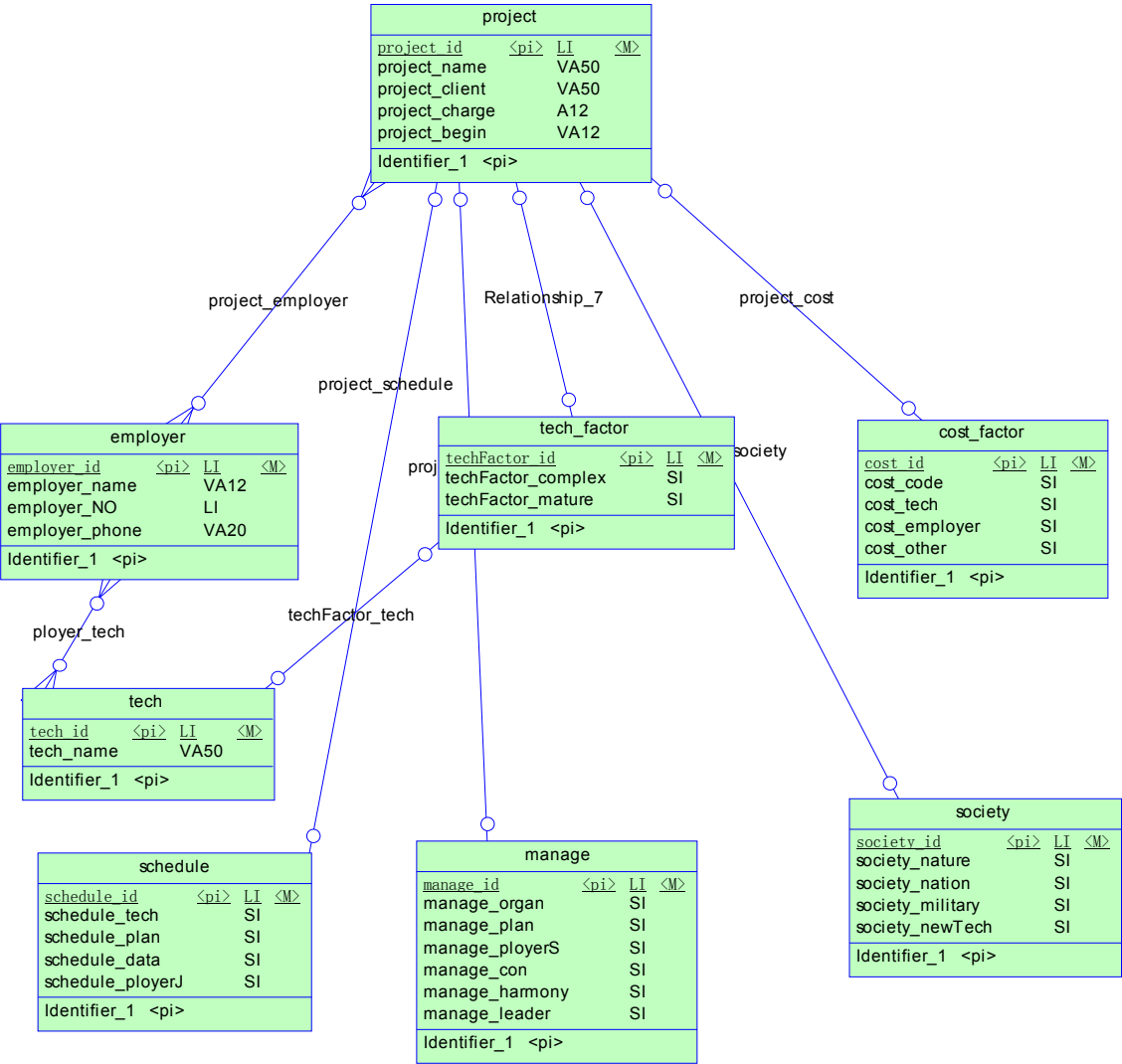


图 5.3 软件风险数据库数据概念模型

Fig5.3 Software risk database conceptiondata model

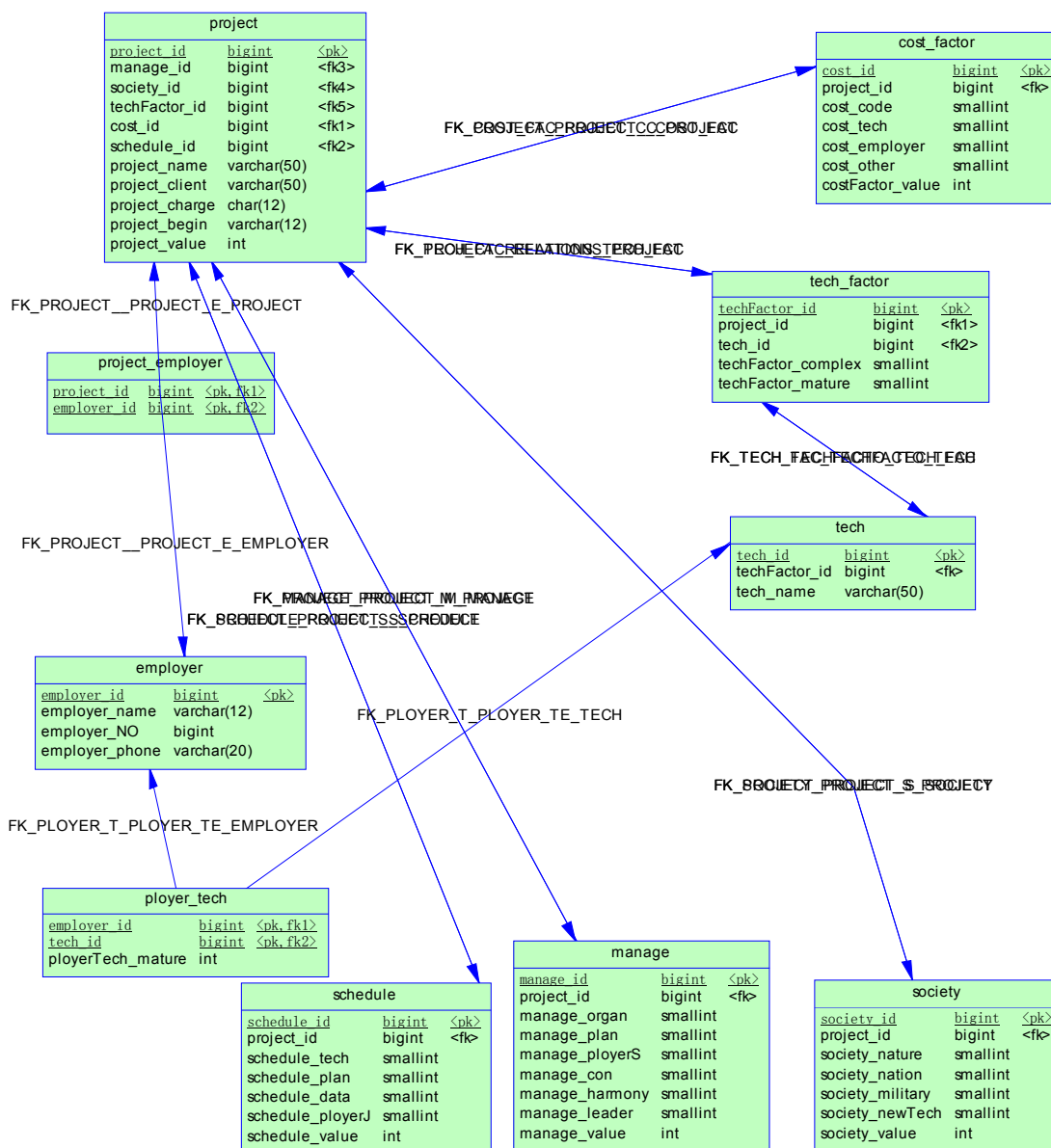


图 5.4 软件风险数据库物理概念模型

Fig 5.4 Software risk database conceptionphysical model

在构建软件风险原型系统过程中，考虑到软件开发人员在使用时各个风险因素出现的概率不尽相同，同时人们在评判风险因素对软件开发的影响程度也不尽相同，所以在建立软件风险分析模型时，这些因素我们都要考虑进去。在这里我们以软件开发技术为例进行说明。

在第三章软件因素的分类中,技术的成熟性、复杂性、与其他项目的相关性、平台的选用、接口技术、用户界面等因素都属于技术风险,而软件开发人员在开发过程中可能使用的技术绝对不止一种,他们往往是多种技术交叉使用,但是每个人对各种技术掌握和理解的程度不尽相同,能力也有高有低,所以在录入技术人员的掌握技术时,不仅仅是录入个人的,而且还要把整个项目小组的技术做一

个评估（在系统中设定小组的技术能力和掌握程度的值是把个人的值相加后再除以参加开发小组的人数，得出平均值，然后再和项目开发组结合软件风险数据库里的历史数据和专家的知识库给出的阈值进行衡量，如果大于阈值，就认为可以进行开发，因为技术人员的经验和能力等能够胜任）。其他的技术风险因素也基本上类同。所以在系统设计中加入了有关技术、成本、进度、管理和社会环境的度量值。下面就是实现系统的人员和技术因素表的部分 SQL 代码。

```

/*=====*/
/* Table: tech                                     */
/*=====*/
create table tech (
    tech_id          bigint          not null,
    techFactor_id    bigint          null,
    tech_name        varchar(50)     null,
    constraint PK_TECH primary key  (tech_id)
)
go
/*=====*/
/* Index: techFactor_tech_FK                       */
/*=====*/
create index techFactor_tech_FK on tech (
    techFactor_id
)
go
/*=====*/
/* Table: tech_factor                             */
/*=====*/
create table tech_factor (
    techFactor_id    bigint          not null,
    project_id       bigint          null,
    tech_id          bigint          null,
    techFactor_complex smallint      null,
    techFactor_mature smallint      null,
    constraint PK_TECH_FACTOR primary key  (techFactor_id)

```

```
)
go
/*=====*/
/* Index: Relationship_8_FK */
/*=====*/
create index Relationship_8_FK on tech_factor (
project_id
)
go
/*=====*/
/* Index: techFactor_tech2_FK */
/*=====*/
create index techFactor_tech2_FK on tech_factor (
tech_id
)
go
alter table cost_factor
add constraint FK_COST_FAC_PROJECT_C_PROJECT foreign key
(project_id)
references project (project_id)
go
alter table manage
add constraint FK_MANAGE_PROJECT_M_PROJECT foreign key
(project_id)
references project (project_id)
go
alter table ployer_tech
add constraint FK_PLOYER_T_PLOYER_TE_EMPLOYER foreign key
(employer_id)
references employer (employer_id)
go
alter table ployer_tech
add constraint FK_PLOYER_T_PLOYER_TE_TECH foreign key (tech_id)
references tech (tech_id)
```

```
go
alter table project
    add constraint FK_PROJECT_RELATIONS_Tech_FAC foreign key
(techFactor_id)
    references tech_factor (techFactor_id)
go
alter table project
    add constraint FK_PROJECT_PROJECT_C_COST_FAC foreign key
(cost_id)
    references cost_factor (cost_id)
go
alter table project
    add constraint FK_PROJECT_PROJECT_M_MANAGE foreign key
(manage_id)
    references manage (manage_id)
go
alter table project
    add constraint FK_PROJECT_PROJECT_S_SCHEDULE foreign key
(schedule_id)
    references schedule (schedule_id)
go
alter table project
    add constraint FK_PROJECT_PROJECT_S_SOCIETY foreign key
(society_id)
    references society (society_id)
go
alter table project_employer
    add constraint FK_PROJECT__PROJECT_E_PROJECT foreign key
(project_id)
    references project (project_id)
go
alter table project_employer
    add constraint FK_PROJECT__PROJECT_E_EMPLOYER foreign key
(employer_id)
```

```
        references employer (employer_id)
    go
    alter table schedule
        add constraint FK_SCHEDULE_PROJECT_S_PROJECT foreign key
(project_id)
        references project (project_id)
    go
    alter table society
        add constraint FK_SOCIETY_PROJECT_S_PROJECT foreign key
(project_id)
        references project (project_id)
    go
    alter table tech
        add constraint FK_TECH_TECHFACTO_TECH_FAC foreign key
(techFactor_id)
        references tech_factor (techFactor_id)
    go
    alter table tech_factor
        add constraint FK_TECH_FAC_RELATIONS_PROJECT foreign key
(project_id)
        references project (project_id)
    go
    alter table tech_factor
        add constraint FK_TECH_FAC_TECHFACTO_TECH foreign key (tech_id)
        references tech (tech_id)
    go
```

5.3 小结

本章结合前面软件风险因素的分析、分类和风险决策树，构建了软件风险估测模型，并在此基础上使用面向对象思想实现了一个软件风险分析模型，分别使用 UML 用例图、顺序图进行功能介绍，最后对软件风险分析模型进行了分析和设计。

6 结论与展望

6.1 本文的结论

本文主要是结合软件风险管理的实践，探讨了影响软件开发的风险因素和分类，并利用软件风险管理的定量和定性风险分析方法，提出了软件风险管理的风险分析原型系统。

逃避风险永远无法获胜。除非在软件的开发过程中，一直处于无风险的状态。只有充分地理解和学习软件风险管理的知识和技能，同时在实践中不断地积累经验才能有效地进行风险防范和控制，达到减少风险的影响程度和实现利益最大化追求的目的。因此，本文详细地介绍了软件风险管理的一些基础知识，找出了影响软件开发成功的风险因素，并结合 SEI 风险分类方法对影响软件风险的因素进行分类，在实际中我们又进行问卷调查、访问，进行了数据分析，用软件开发企业和人员的实际经验教训充实研究。

接着，本文详细地介绍了软件定量和定性分析方法，比较了他们的优缺点，并对软件定量和定性方法进行研究。然后根据软件风险的特征，建立了多阶段多层综合风险估测模型，并对这个模型的可行性进行验证。此外，还采用 UML 进行系统分析设计，以 PowerBuilder 等为工具，使用了面向对象思想实现了一个软件风险分析模型。

本文所有这些工作的意义是：1) 较为详尽、系统地介绍了软件风险管理的理论基础，有助于软件开发人员对软件风险管理能有一个全面的认识；2) 本文详细地分析了影响软件开发的风险因素（项），并对其进行分类，组织成系统的层次结构，使软件开发人员能够有效地识别和规避风险；3) 本文着重介绍了软件风险定量、定性分析方法，并比较其优缺点，同时还结合风险决策树提供了软件风险定性、定量分析方法的策略，使用了面向对象思想实现了一个软件风险分析模型，这对提高软件产业的投资回报率和成功率具有一定的理论意义和工程参考价值。

6.2 进一步的工作

本文的工作结论很大程度上建立在两基础上：他人的风险管理理论研究成果以及本人对软件风险管理的调查研究和实践。因此调查调查研究越深入越细致，所得的结论就会越趋向于正确性和有效性。而在调查过程中，受调查者对有的风险指标的看法并不统一，有的表示并不重要或者不十分重要，而有的则认为十分重要，同时部分软件开发企业未形成有历史数据，部分问卷回收不全，所以在形成调查数据时不十分充分，所以还应该加大调查面，多收集一些历史数据，以

便获得更具广泛性和精确性的风险分析模型。其次由于在软件开发过程中，各个项目的风险都有所区别，我们在建立软件分析模型过程中只能考虑到普遍性，寻找普遍、常见的影响软件开发的危险因素，所以还不能涵盖所有的软件开发风险，这有待于在今后工作中进一步积累知识，构建一个比较完善的软件风险管理专家知识库。

致 谢

本文的研究工作是在我的导师曾一副教授的精心指导和悉心关怀下完成的，在我的学业和论文的研究工作中无不倾注着导师辛勤的汗水和心血。导师的严谨治学态度、渊博的知识、无私的奉献精神使我深受的启迪。从尊敬的导师身上，我不仅学到了扎实、宽广的专业知识，也学到了做人的道理。在论文刚刚成稿时，由于论文涉及的理论太过广泛，很多相关的理论没有组织恰当，在导师的指导下，论文作了几次较大的改动。在此我要向我的导师致以最衷心的感谢和深深的敬意。

在本文的写作期间，得到了我的同学张小华、徐珂等人的帮助和支持，在此对他们一并表示衷心的感谢。

在此，向所有关心和帮助过我的领导、老师、同学和朋友表示由衷的谢意！

最后，我还要感谢我的父母以及我的妻子李红英、女儿涂婧雯对我的督促和支持。

衷心地感谢在百忙之中评阅论文和参加答辩的各位专家、教授！

参考文献

- [1] Boehm B. "A Spiral Model of Software Development and Enhancement." IEEE Computer 21(5) (1988): 61~72,1988.
- [2] Defense Systems Management College. Risk Assesment Techniques. Fort Belvoir, VA, 1983
- [3] Air Force. Software Risk Abatement. AFSC/AFLC plmphet800-45. Wright Patterson Air Force Base. OH: Air Force systems Command, Air Force Logistics Command,1988.
- [4] VanScoy R. Software Development Risk: Problemor Opportunnity. Trchnical report CMU/ SEI-92TR-30, PA: Software Engineering Institute, Carnegie Mellon University,1992.
- [5]薛华成,《管理信息系统(第三版)》,清华大学出版社,1995,5
- [6]谢科范,《技术创新风险管理》,河北科学技术出版社,1999,1
- [7] Bob Hughes Software project Management , IEEE Trans. Software Eng. 2002,8
- [8]陆正中,许铁军译《高质量网络项目管理》,中国铁道出版社,2003
- [9] Carr M, Konda S, Monarch I, Ulrich C, Taxonomy Based Risk Identification. Technical report CMU/SEI-93-TR-6.PA: Software Engineering Institute, Carnegie Mellon University,1993.
- [10] Houman Younessi Objec-Oriented Defect Management of Software, IEEE Trans. Software Eng. 2002,8
- [11] Integrated Computer Aided Manufacturing Architecture, Part II ,Vol. IV: Function Modeling Manual(IDEFO).AFWAL-TR-81-4023. Wright—Patterson Air Force Base , OH: Air Force Systems Command, June 1981
- [12]Aletr, S .Implementation risk analysis .TIMS Studies in Management Sciences 13,2(April 1979),103-119
- [13]Zmud ,R .W ,Management of large software development efforts .MIS Quarterly ,4,2(June 1980),45-55
- [14] FergusO' Connell How to Run Successful ProjectsIII The Silver Bullet, IEEE Trans. Software Eng. 2002,8
- [15] Humphrey,PSP Technical Report. IEEE Trans. Software Eng. 2000,7
- [16] Bernstein Peter L. Against the Gods: The Remarkable Story of Risk. New York: Wiley 1996
- [17] 何云,《项目管理中风险评价的必要

性》, <http://www.51cmm.com/ProjectManage/No095.htm>

[18] 丁志强, 张宇琨. 《计算机系统管理与项目管理》, 电子科技大学出版社, 1993, 3

[19] Gordon Schulmeyer, Handbook of Software Quality Assurance, IEEE Trans. Software Eng. 2002, 8

[20] Hall E. "Formal Risk Management: #1 Software Acquisition Best Practice." Proc. 4th SEI Conference on Software Risk, Monterey, CA, November 1995.

[21] 沈备军, 宿为民译. 《软件同级评审》, 机械工业出版社, 2003, 6

[22] 王海鹏, 周靖译. 《风险管理-软件系统开发方法》, 清华大学出版社, 2002, 9

[23] Karl E. Wiegers Peer Reviews In Software: A Practical Guide, IEEE Trans. Software Eng. 2002, 8

[24] 何云, 软件系统安全性风险评价

<http://www.51cmm.com/ProjectManage/No107.htm>

[25] Sami Zahran Software Process Improvement Work, IEEE Trans. Software Eng. 2002, 8

[26] 唐毅鸿, 杨朝晖, 刘倩羽. 《软件性能工程》, 机械工业出版社, 2003, 6

[27] C. Smidts, R. Stoddard, and M. Stutzke, "Software reliability modeling: An approach to early reliability prediction," IEEE Transactions on Reliability, vol. 47, no. 3, pp. 268 — 78, 1998.

[28] 邹丰忠, 徐仁佐. 《软件可靠性多模型综合评估》, 同济大学学报, 2002

[29] 杨立群. 软件开发项目失败的原因, 中国计算机学报, 2001, 11

[30] Schulmeyer, G. Gordon. 《零过失错误软件》. McGraw-Hill, Inc. 1990. ISBN 0_13_819780_6

[31] 陈兵. 《软件开发质量和风险的定量监督》

<http://www.sawin.com.cn/doc/SE/SPI/psupervise2.htm>

附录 A 在校期间的研究成果

- [1] 涂争光,曾一,吴小平.软件分析方法研究.《重庆大学学报》(自然科学版)于 2005,3.录用
- [2] 张小华,曾一,涂争光.架构软件配置管理模型.《计算机工程与设计》于 2004,11.录用
- [3] 宋维平,曾一,涂争光.B/S 模式下 OA 系统的权限控制设计与实现.《计算机工程与应用》于 2004,12.NO.35.刊出
- [4] 高旻,曾一,涂争光.MATLAB 的 WAB 技术应用与方法探讨.《计算机应用》于 2004,6.Vol. 24.NO. 6 刊出.

附录 B 项目风险变量

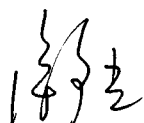
- 1、项目组的经验
 - 1a 类似项目业务的经验
 - 1b 类似规模/复杂度项目的经验
- 2、项目经理的经验
- 3、综合评价体系
- 4、项目评估模型
- 5、可供借鉴评估的以前的经验
 - 5a 使用以前的开发效率指标
 - 5b 以前预测的成本/进度与实际情况的吻合度
- 6、里程碑的明确性
- 7、对项目成本和进度的监控
- 8、建立问题及解决日志
- 9、对数据的重视程度
 - 9a 以数据为中心进行开发
 - 9b 对数据规范化的重视
- 10、技术复杂度
 - 11a 使用的软件技术的成熟度
 - 11b 使用软件技术对于开发人员而言的复杂度
- 11、软件使用的外部支持
 - 11a 开发方提供的系统使用的培训
 - 11b 开发方提供的用户手册的齐全性和指导性
 - 11c 用户在原型系统演示或者软件早期版本审核中的参与性
- 12、需求分析的质量
 - 12a 能否辨识所有的需求
 - 12b 文档化需求的明确性和无二义性
 - 12c 对项目目标等关键指标达成书面协议
 - 12d 妥善处理未能清晰确定的需求
- 13、容错性
 - 13a 设计和实现系统时对出错处理的考虑
 - 13b 对硬件出错的检测和处理
 - 13c 对输入数据的合法性校验

- 13d 数据备份
- 13e 运行日志的完善性
- 14、 人员的流动性
 - 14a 项目开发对少数人员的依赖性
 - 14b 开发人员的稳定性
 - 14c 人员流动对工作持续性的影响程度
- 15、 资源的充足性
 - 15a 开发时间宽松度
 - 15b 开发资金的充足性
- 16、 信息交流、沟通性
 - 16a 开发人员之间的交流沟通
 - 16b 使用方和开发方的信息交流和沟通
- 17、 使用方业务的规范程度
 - 17a 业务流程运行的规范度
 - 17b 基础数据的规范标准化程度
- 18、 开发人员的开发技能
 - 18a 开发人员对开发方法的掌握程度
 - 18b 开发人员对开发实现工具的掌握程度
 - 18c 开发人员对开发过程中使用的技术支持工具的理解和掌握程度
 - 18d 开发人员对所开发的软件的业务领域专业知识的理解和掌握程度
- 19、 系统的可扩充性和易维护性
- 20、 技术的成熟性
- 21、 复查和调整评估结论的频率

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得重庆大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：



签字日期：2005 年 5 月 28 日

学位论文授权使用授权书

本学位论文作者完全了解重庆大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权重庆大学可以将学位论文的全部或部分内 容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

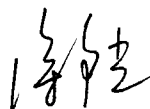
保密（☐），在____年解密后适用本授权书。

本学位论文属于

不保密（☐）。

（请只在上述一个括号内打“√”）

学位论文作者签名：



导师签名：



签字日期：2005 年 5 月 28 日

签字日期：2005 年 5 月 30 日