摘 要

随着信息技术的迅猛前进，软件相关类的产品渗入到我们的生活的层层面面之中，成为不可或缺的组成部分。随着全球经济一体化的到来，信息技术逐渐成为社会发展的基础载体，所有国家都以信息化为基础，以其发展带动其他产业的发展，信息技术已经被应用到所有的社会生活之中。

软件项目风险管理的目的就是找出导致项目需求不明晰、不能按进度计划及时交付、产品质量存在缺陷、开发费用超支等各种不良后果的风险因素，对风险因素及可能造成的后果和危害进行定性和定量分析，从而为软件项目管理人员、软件开发者、软件产品用户等提供有效的风险控制方案和措施，使其对软件项目的损失或影响降到最低程度或者使决策者可以接受的限度。

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。

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，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的各功能和模块，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目的是，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；详细地分析了影响软件开发的风险因素，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；着重介绍了软件风险定量、定性分析方法，并比较其优缺点

关键词：软件风险管理；风险评价；软件开发

ABSTRACT

Along with information technology's swift and violent advance, the software related kind of product permeates our life layer upon layer, becomes the indispensable constituent. Along with the global economic integration's arrival, the information technology becomes the social development gradually the foundation carrier, all national take informationization as a foundation, led other industrial by its development the development, the information technology was already applied during all social life.

The software project risk management's goal is discovers causes the project demand not defined, not to be able according to the progress schedule prompt payment, the product quality existence flaw, the development cost overspending and so on each kind of adverse consequences risk factor, and possibly creates the consequence and the harm to the risk factor carries on qualitative and the quantitative analysis, thus for the software project administrative personnels, the software exploiter, the software product user and so on provides the effective risk control plan and the measure, causes it falls to the software project loss or the influence to the lowest degree or causes the policy-maker acceptable limit.

This text join togethers 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 : 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; 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; this text put great emphasis to introduce the software risked the fixed amount, qualitative analysised the methods, and compare its advantage is with the defect, 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.

Key Words: Software risk; Risk assessment; Software development

大连海事大学学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明：本论文是在导师的指导下,独立进行研究工作所取得的成果，撰写成博/硕士学位论文 “ 软件项目风险管理和控制研究 ”。除论文中已经注明引用的内容外，对论文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本论文中不包含任何未加明确注明的其他个人或集体已经公开发表或未公开发表的成果。本声明的法律责任由本人承担。

学位论文作者签名：

学位论文版权使用授权书

本学位论文作者及指导教师完全了解大连海事大学有关保留、使用研究生学位论文的规定，即：大连海事大学有权保留并向国家有关部门或机构送交学位论文的复印件和电子版，允许论文被查阅和借阅。本人授权大连海事大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，也可采用影印、缩印或扫描等复制手段保存和汇编学位论文。同意将本学位论文收录到《中国优秀博硕士学位论文全文数据库》（中国学术期刊（光盘版）电子杂志社）、《中国学位论文全文数据库》（中国科学技术信息研究所）等数据库中，并以电子出版物形式出版发行和提供信息服务。保密的论文在解密后遵守此规定。

本学位论文属于：　　保　密□ 在____年解密后适用本授权书。

不保密□ （请在以上方框内打“√”）

论文作者签名： **导师签名：**

日期：　　年　月　日

第 1 章 绪论

1.1 研究背景及意义

随着信息技术的迅猛前进，软件相关类的产品渗入到我们的生活的层层面面之中，成为不可或缺的组成部分。同时，随着相关开发公司数量与规模的迅速扩大，其结构也不断复杂化，软件项目也成倍增加，复杂度也相应增加。随着全球经济一体化的到来，信息技术逐渐成为社会发展的基础载体，所有国家都以信息化为基础，以其发展带动其他产业的发展，信息技术已经被应用到所有的社会生活之中。在欧洲，美洲等的经济大国如美国、日本，信息产业的已经超过了国内生产总值的五成以上，信息产业成为许多国家发展经济的中流砥柱。从某种角度上来说，判断一个国家的综合经济能力的强弱，可能从其信息产业的发展程度上来判断，同时，信息产业也会直接影响其之后在全球经济竞争中的表现。现在，中国也充分认识到软件行业其不可或缺的地位，随着中国经济的蓬勃发展，信息公司如雨后春笋般 出现在中国的许多城市之中，并且有许多公司已经上市，成为中国软件市场的中流砥柱，从而带动了软件市场优胜劣汰的交替过程，提高了整体的竞争力，以使我们的产品在国际市场上占有更多的份额。

但是，在很大一段时间里，软件开发项目也遇到了重重地阻力。在信息行业中，由于不能符合客户的要求，不能在预定的时间里交付项目，或财政预算严重超标，以使项目陷入无粮之地，这样就导致了许多的软件项目失败的案例。根据报告显示，大约 75%以上的项目不能按期完成，平均用了比原先多五分之一或者一半的时间，只有 10%的项目会在成本预算之内完成，然而，这个比率会随着项目的规模而减少[1]。软件项目的成功率在所有的工程项目之中是最低的。软件行业的前景不容乐观，它曾经重创了许多国家的经济发展。

这引发了我们深刻的思考，是什么原因导致了软件行业的危机呢？不难得出，软件产品其本身的特点使其与众不同。作为一种特殊的逻辑产品，不具备实体的可见性，它是由人经过智力劳动而产生出来、具有特殊性质的复杂事物[2]。此外，其开发过程也异于其他的工程项目，它具有自身的特性。与其它工程

项目和工程产品相比较，我们不难得出软件项目的特点：

(1) 软件的不完善性，其提高与完善是在软件的开发以及使用之中进行的。所以，一个软件产品都不是终极产品，它需要在投入使用之后不断地完善以及维护。

(2) 由于软件项目不具有实体可见性，我们很难对其开发周期，自身的质量还有财政预算有一个明确的衡量标准，这直接导致了无法有效地进行软件开发的管理与监控。

(3) 项目的需求会随着时间的变化而更改，这一过程经常会发生在软件的开发过程中，赋予了软件开发管理的不稳定性。因为，在软件开发一开始，由于用户不能准确的阐述自己的需求，亦或者由于开发人员对用户需求理解的偏差会导致软件开发的不断更改。因此，软件的开发需要开发人员与软件用户之间进行不断地沟通从而不断开发与完善软件。在互相的沟通之中，需求会慢慢的明确，从而软件开发的波动性会随之平稳。

(4) 软件项目的生产不是一个完全的过程。每一个项目与其它的项目都有所区别。每一个项目都有与其它项目相区分的特点，即使同一个项目，因为背景不同也会有所不同。虽然新的开发项目可见借鉴之前的项目开发的经验与成果，但是如果想完全继承之前的项目是不可能的。所以，从这个层面上来说，项目软件管理总会有其顾及不到的地方，而且也很难完全避免危机情况的再次发生，它只能在最大程度上减少与预期估计不同而产生的损失。

(5) 对软件开发进行预算是一件比较困难的工作。因为每个软件项目总会与之前的项目有所不同，而软件开发的周期和参政的预算又大都会借鉴曾经做过的项目而积累出来的资料和经验，由此可以得出，这样的预估计的准确性很差。

(6) 随着越来越多的软件公司的出现，其行业内部的竞争不断加剧，软件开发面对的困难越来越多，时间少、责任重、压力大。在激烈的竞争环境下，软件公司必须采取各种方法保证自己可以在规定的时间里交付让客户满意的商品。

从以上软件及软件开发的特点我们可以得出，软件开发的过程是一个不确定的过程，有很多的因素我们无法预先的进行预测。正是由于这些不确定性让

软件项目具有更多的风险，同时软件成功的几率大大减少[3]。软件开发项目是一个精密的系统工程，要想使得软件开发获得成功，不能只从技术入手，如何进行科学的预测与管理同样会直接影响到软件开发的成败。

由此，软件工程应运而生，其形成是将工程化思想应用到软件开发之中，同时，软件工程的诞生又使软件管理初具模型。在过去的发展中，我们将工程管理中得到的成功经验试用于软件开发中，再连系软件开发自身所具有的性质，逐渐发展出软件项目管理这一科学分支。其出现减少了软件开发中由于时间和成本问题而导致的软件开发案例失败的数量，是软件开发得到了极大地提高。

项目管理的目标是在有限资源标注条件下，保证项目时间（进度）、质量、成本（花费）达到最优化[4]。而软件项目管理的主要目标是确保软件产品能够按预期方案交付，同时还要满足用户需求[5]。由于软件开发项目本身是一项耗费巨大的复杂工程，其开发过程涉及到软件规模、参与人员、开发技术和方法以及外部环境等诸多方面的因素，需要进行时间、人员、管理和财物等的大量投入，所以在软件项目中存在的不确定性因素就比较多。因此，软件项目与其它工程项目相比较具有更大的风险，对其进行风险分析、控制和管理也就比较困难。

软件项目风险管理的目的就是找出导致项目需求不明晰、不能按进度计划及时交付、产品质量存在缺陷、开发费用超支等各种不良后果的风险因素，对风险因素及可能造成的后果和危害进行定性和定量分析，从而为软件项目管理人员、软件开发者、软件产品用户等提供有效的风险控制方案和措施，使其对软件项目的损失或影响降到最低程度或者使决策者可以接受的限度。因此，从某种意义上讲，软件项目管理在很大程度上就是软件项目风险管理[6]。

值得提出的是，现在的对项目风险的探索中，重点主要放在进度、预算目标和质量上。然而，针对软件项目，关于质量方面的探索已经探出了项目风险的领域。比如，在软件工程方面，关于质量所产生的风险大多数属于软件可靠性的领域。而针对于软件质量及其控制问题也有分别与之相对应的领域对其加以研究。比如，软件度量学，统计过程控制等技术。以此，此上的内容不属于本篇文章的研究领域。所以，本篇文章主要关注软件开发的周期和预算目标，

而忽略质量管理方面的内容。

上个世纪 80 年代后期，软件项目风险管理从软件工程衍生出来，经过近半个世纪的发展，无论从理论方法还是具体的实践都取得一些成果。当前，随着软件开发渐成规模，并且软件企业也在茁壮成长，有关软件项目的风险控制的问题慢慢在当前领域浮现出来，并且受到了越来越多的注意[7]。

本篇文章开始先从基本概念入手，介绍软件风险控制及管理的相关发展状况，在对软件风险有了笼统的了解后，从系统工程入手，将重点放在软件的风险分析、过程控制等方面。在文章中，本文针对当前软件项目风险控制领域存在的诸多问题进行了详细的分析，比如，缺少统一的量化标准，风险控制的模型有所欠缺。试图建立一套更加可行的风险管理模型和风险管理系统。本篇文章将风险管理的理论与实际相结合，进行项目风险因素的识别与分析，然后根据分析的结果进行风险管理的规划，从而达到软件风险的管理与控制。以提高软件开发项目的成功率。

1.2 国内外研究现状

1.2.1 国外研究现状

国际上关于软件风险管理研究大体上是从 2 0 世纪 8 0 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[8]. .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[9]。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[10]。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[11]；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[12]。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。

1.2.2 国内研究现状

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[13]。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[14]。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[15]。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不确定性进行描述，从而模拟出项目变化的规律。

当前，软件风险控制及管理引起了许多软件公司的注意，为了进一步发展软件的风险管理，我们引进了或者自行研究出多种软件系统。各自的开发环境琳琅满目，耗资巨大，但是并没有给我国的软件风险管理有一个质的提高。这里面存在众多的因素，总的来说有以下几方面：

1）软件风险控制还没有引起大多数公司的重视，其结果也是显而易见的

2）开发公司在很多情况下没有给予那些提出可能引起项目失败的问题的人足够的重视，所以开发人员会有抵触情绪

3）风险管理并没有成为项目开发的核心部分，即使一些大的软件开发公司也没有完整的、有效的风险管理体系。

4）开发人员对软件风险没有引起足够的重视理解不完全。还有许多开发者正在应用的都是比较原始的风险管理技术。例如：简单的数据分析或者直接凭借开发者的经验或者直觉。缺少一个量化的标准惊醒风险的管理与控制[16]。

1.3 主要的研究内容

本文通过对软件的特点及其开发中的风险作为基本的研究对象。将风险的控制应用到软件项目的整个过程中。罗列了众多的典型的软件项目风险管理模型，并对它们进行了比较，提出了一种新的风险管理模型。

首先要建立软件项目风险的一个实用性的模型。在研究了国内外的许多经典风险管理的模型之后，提出一种新的给予改进后的新模型，提高目前中国许多软件开发项目缺少定量分析的缺点，是软件分析及控制具有更高的可靠性。

根据前文的分析与新建立的模型，开发出一个具有定量分析、可进行风险预测及控制的风险管理系统，并使系统具有良好的交互性。

1.4 本文的组织结构

第一章：介绍了本文的研究背景及意义，软件风险管理研究的国内外现状；同时介绍了本文的研究内容和组织结构；

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风险因素，并进行了分类归纳整理，同时阐述了风险因素与风险事件之间的关系。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。并分析了影响软件风险的因素。

第四章：在前面详细分析了软件项目风险的分类和概念后，建立基于改进后的贝叶斯的网络模型。

第五章：构建了软件风险量化评估的原型系统。描述个系统中的功能，并进行了系统的设计。

最后，对全文进行总结与对未来的展望。

第 2 章 软件风险管理的理论基础

要想对风险管理进行深入的讨论，我们应该从了解软件风险的相关概念开始。进一步掌握软件风险管理的相关理论基础。

本篇文章涉及的是软件风险的控制问题，根据软件项目的多样性，风险可能发生在不同的领域。因此其发生的概率也不完全相同，这也就使得风险的管理方法的多样性。总的来说，由于软件项目的多样性，使其相应领域的风险管理也具有多元化。但是，从宏观的角度来看，风险管理的概念及基本的相关理论是相同的。

2.1 软件风险管理的相关概念

2.1.1 风险的定义

软件项目风险的基石是风险概念。但是由于在许多其他的领域例如自然经济及工程，所以我们没有一个统一的定义，只是从各个方面对风险进行陈述。所以我们不妨从各个方面来理解风险，包括它的陈述、特点及分类。这将帮助我们更深一步的开展对风险管理的讨论。

风险在字典中有如下定义“可能发生的危险”。

现代汉语字典把风险定义为“可能发生的危险”，韦氏字典中将风险定义为“遭到伤害或损失的可能性”，美国 Coper D. F 和 Chpamnac B 在《大项目风险分析》一书中给出了较权威的定义:“风险是由于从事某项特定活动过程中存在的不确定性而产生的经济或财务的损失，自然破坏或损伤的可能性。”在美国国防部(DOD)文件中，将风险定义为可能危及计划或工程项目的潜在问题，并用问题发生的可能性及其后果(经度量或评估)的综合影响来度量风险。美国的 MI-LSTD- 882c 和 MIL-HDBK-764 把风险定义为事故的先决条件。ESA 对风险的定义则是可能造成危害或对安全性具有潜在危险之源。我国在 1990 年制定的 GJB90O《系统安个性通用大纲》中对风险的定义为事件的风险就是该事件的发生概率和损失程度的函数[17][18]。

任何风险都包括三个方面要素:即发生了什么有害事件?有害事件发生的可能性有多大?如果发生产生的后果如何？这三个方面构成了评估风险的基础。据此，Kaplna 和 Garrick 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

$$R_{risk} = \{<s_i, l_i, x_i>\}_c \quad (2\text{-}1)$$

其中，R_{risk}代表风险，s_i代表第 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 代表第 i 个事件的结果，是一种损失指标，c 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)只是风险的一个答案，整个集合才是全部风险。在 1997 年风险分析学会的大会报告中，Kaplan 进一步完善了这种完备集风险的定义。他从学术界对概率定义的争论出发，指出可能性有三种表达: 频率、概率和频率的概率，其中频率的概率是最有说服力、最适用的。基于这种认识，公式(2-1)可转化为：

$$R_{risk} = \{<s_i, p(\varphi_i), p(x_i)>\}_c \quad (2\text{-}2)$$

其中 s 仍然代表第 i 个有害事件; ψ_i表示第 i 个事件发生的频率，$p(\varphi_i)$代表第 i 个事件发生频率为 φ_i 的概率, $p(x_i)$代表第 i 个事件的结果为必 i 的概率，它是一个向量，与事件不独立。显然，该完备集风险的定义在量化上是一个进步[19]。

2.1.2 软件风险的特征

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，而且在项目的生命周期内，风险是无处不在、无时不有的[20]。这些都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的原因。

第二，风险的不确定性。不确定性是风险最本质的特征，由于客观条件的不断变化以及人们对未来环境认识的不充分性，导致人们对事件未来的结果不能完全确定。风险是各种不确定因素综合的产物。

第三，风险的行为相关性。行为相关性是指决策者面临的风险与其决策行为是紧密关联的。不同的决策者对同一风险事件会有不同的决策行为，具体反映在其采取的不同策略和不同的管理方法上[21]。因而也就会面临不同的风险结

果。风险的行为相关性表明，任何一种风险实质上都是由决策行为与风险状态结合而成的，是风险状态与决策行为的统一，风险状态是客观的，但其结果会因不同的风险态度和决策行为而不同。

第四，风险的可变性。这是指在项目实施的整个过程中各种风险在质和量上会发生变化，随着项目的进行，有些风险得到控制，有些风险会发生并得到处理，同时在项目的每一阶段都可能产生新的风险。

第五，多样性和多层次性。这一特征主要体现在大型项目中，因啦大项目周期长、规模大、涉及苑围广、风险因素数量多且种类繁杂，致使其在全生命周期内面临的风险会多种多样，而且大量风险因素之间的内在关系错综复杂、各风险因素之间的影响以及与外界的交叉影响，又使风险呈现出多层次性[22][23][24]。

2.1.3 软件风险的种类

(1) 按风险的内容分类

一般来说，根据风险的内容，一个规范的软件项目在实施过程中面临的风险可分为 6 个方面:商业影响、社会环境、技术(与性能有关)、费用、进度和管理，如图 2.1 所示。

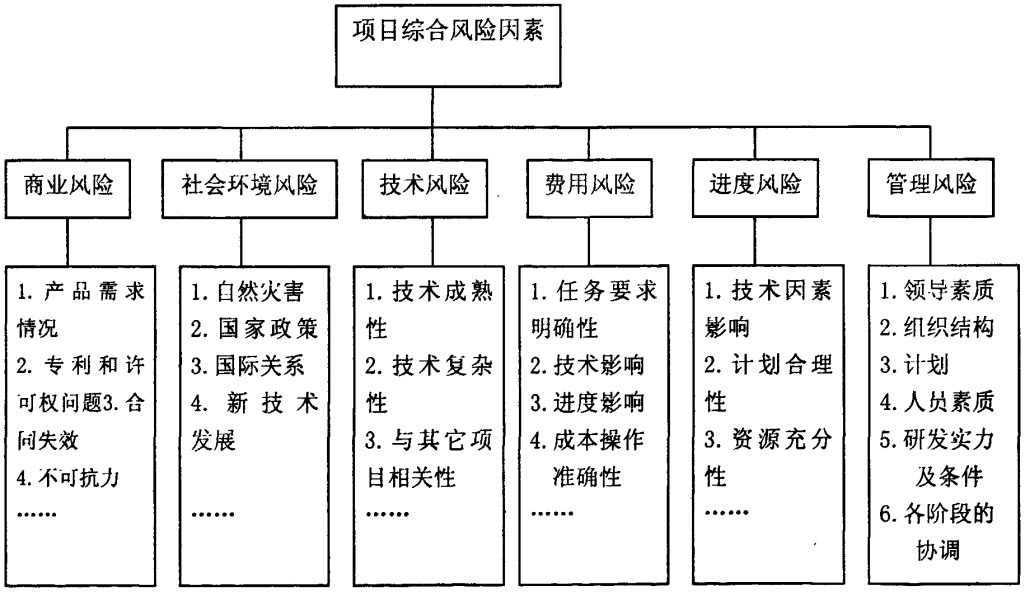

图 2.1 风险内容分类图

Fig .2.1 Risk Content Classification

(2) SEI 的风险分类

SEI 把风险分为两大类: 管理和技术。软件风险是度量不如人意的结果的可能性和损失的方法，这些结果影响到软件项目、过程和产品。管理包括项目风险和管理过程的风险。技术包括产品风险和技术过程的风险。如图 2.2 所示。

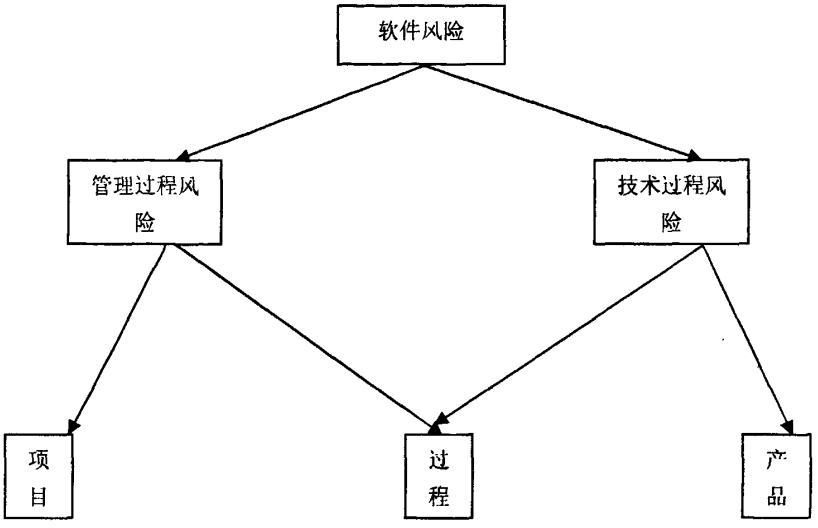

图 2.2 SEI 的软件风险分类

Fig. 2.2 SEI Software Risk Classification

软件项目风险定义了操作过程的、组织过程的和合同等软件开发参数。项目风险是主要的管理责任。项目风险包括资源制约、外界因素、供应商关系和合同制约。其它风险还包括不负责任的厂商和缺乏组织的支持。

软件过程风险包括管理和技术工作规程。在管理规程中，人们可能在一些活动(如计划、人员分配、跟踪、质量保证和配置管理)中发现过程风险；在技术过程中，过程风险可能存在于工程活动中(如需求分析、设计、编码和测试)中。

软件产品风险。这类风险包括中间的和最后的产品特征。产品风险主要是技术责任。产品风险可能存在于需求稳定性、设计性能、编码复杂度和测试明细清单中。因为软件需求通常被视为灵活的，所以产品风险难于管理。

(3) 按确定性分类

按确定性分类，可将风险分为已知风险、可预知风险和不可预知风险三

类。

已知风险：此类别指经过审查项目进度、成本计算、技术背景以及其他的一些评估内容之后觉察出的一些风险。

可预知风险：此类别是通过对过去经验的继承，根据过去项目中出现的问题（人员频繁更换对项目进度的影响、开发人员对项目的理解程度对项目的影响）而得出的风险。

不可预知风险：对软件项目进行了系统的分析与评估后，将可能出现的风险进行控制与管理，但是仍不能保证在未来的软件开发中会出现的不能预测的风险[25]。

2.2 风险管理的特征

风险是相对于即将要来临的事情而言，它包含了很多方面，最宏观的是从时间、空间上产生的因素；随着事物的发展，环境也相应的随之变化，与此同时，风险的特征也会随之变化[26]。所以，如果我们想要控制风险，就要从引起风险变化的因素入手，控制好这些风险因素，将有助于我们进行对风险的控制。以下是风险的特点：

(1) 风险是客观存在的。并且存在于所有的事物之中。风险是不以人的意志为转移而发生的。并且在项目的整个周期中，任何一个过程都会存在风险。从这些特点中我们不难理解，在长期的与风险斗争的过程中，我们努力地影响风险的生存环境，期望让它们按照我们的意愿进行发展，从而提高软件的成功率，但是，我们并不能完全的控制住风险，其总有一定的不可控性。

(2) 从科学辩证的角度来看，某一风险发生的偶然和众多风险的发生一定是必然的。任一个风险的发展都是由其环境中的各个条件相互反应而发生的，其结果是不可预见的。如果来看单个的风险，我们能发现，它们的发生没有规律可循。当我们把大量的风险示例进行总计分析，可以发现它们呈某种分布，人们开始用概率统计等方法来评估风险的各种数据，这让风险管理有了一个质的飞跃。

(3) 风险的不确定性。在软件项目开发期间，随着环境的变化，风险的大小和严重情况可以随着条件的变化而进行变化。有的风险被成功的规避了，有的风险造成了部分的损失，与此同时，可能有新的风险从旧的风险中孕育而

来。

(4) 风险的多样化。在一些大的软件工程中，由于其开发的规模巨大，人员繁多，组织结构庞大，就有可能导致在开发过程中出现各种各样的风险。同时各个风险之间又相互缠绕在一起，相互影响，这是大型软件项目中非常常见的一个特点[27][28][29]。

2.3 风险管理模型

2.3.1 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[30]。

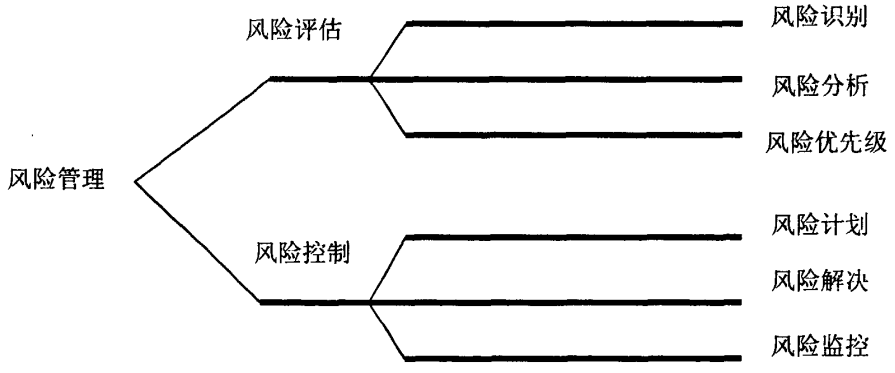

图 2.3 Boehm 风险管理模型

Fig .2.3 Boehm Risk Management Model

Boehm 模型思想的核心是 10 大风险因素。其中包含人员短缺、不合理的进度安排和预算、需求的不断变动等。Boehm 分析了每一个风险因素，然后提出了 相应的管理办法。对待不显示不合理的进度安排和预算，应该采用增量式开发方法对软件进行重新开发或者对要求做一些改动。在实际环境中，以 10 大风险清单为基础，对当前项目即将出现的风险进行分析，之后再对 10 大风险因素进行解决再次讨论，来提出新的 10 大风险因素名单，循环往复。

10 大风险因素清单的作用是将决策层的所应关注的事情聚集在优先级较高的，对软件项目影响较大的因素上，从而忽略掉不是十分重要的因素。此外，这个清单来源于对一些重大的项目，通过对它们的详细研究最后总结而出的，因此具有一定程度的实用性。但是它也存在一些不足，它只是针对风险因素的全集进行了分析整理，并没有对风险因素进行具体的定义，阐述其分析的方法。此外，由于用于参考的对象过少，集合并不完整，造成 10 大风险清单没有描述整个风险因素集合。甚至有些因素在描述的是同一事件。这说明清单不完整，需要进一步的完善。

Boehm 被认为是软件风险管理的创始人。所以在其理论上的研究并不十分的完备。在他提出了风险管理的观点后，涌现除了越来越多的机构和科学家开始对这一学科进行深入的研究，软件风险管理也逐渐引起大家的注意力。

2.3.2 SEI 的 CRM 模型

SEI（美国卡内基•梅隆大学软件工程研究所）认为，在软件项目的进行中，风险是由各种风险因素的作用而致使计划遭受的损失。其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的失败。风险管理的本质就是，找出可能导致项目失败的风险因素、为所有的风险因素设定优先级、建立可以对风险进行有效管理的计划；随时关注风险管理计划的运行情况，保证其实施的正确性。SEI 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[31][32]。

CRM（Continuous Risk Managements）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。SEI 描述了这个模型。并设计出一个风险管理模型。如图 2.4 所示，可以从图中看出信息流的方向。沟通则是逻辑流的最重要的实现方法，它描述了基于 CRM 的五大要素之间的关系，说明这个模型会在软件开发过程中循环的出现。

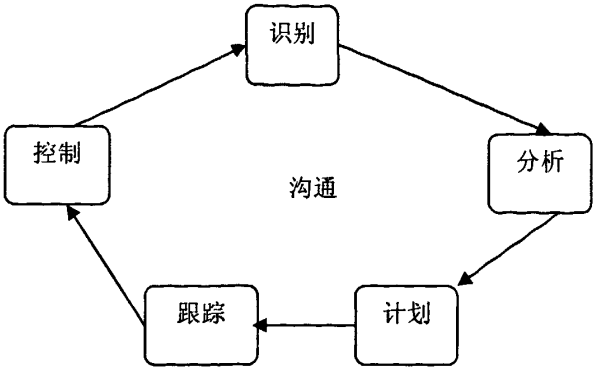

图 2.4 SEI 风险管理范例

Fig .2.4 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 IDEFO[33][34]（Integrated Computer-Aided Manufacturing Definition 简称 IDEFO，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 2.5)；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。

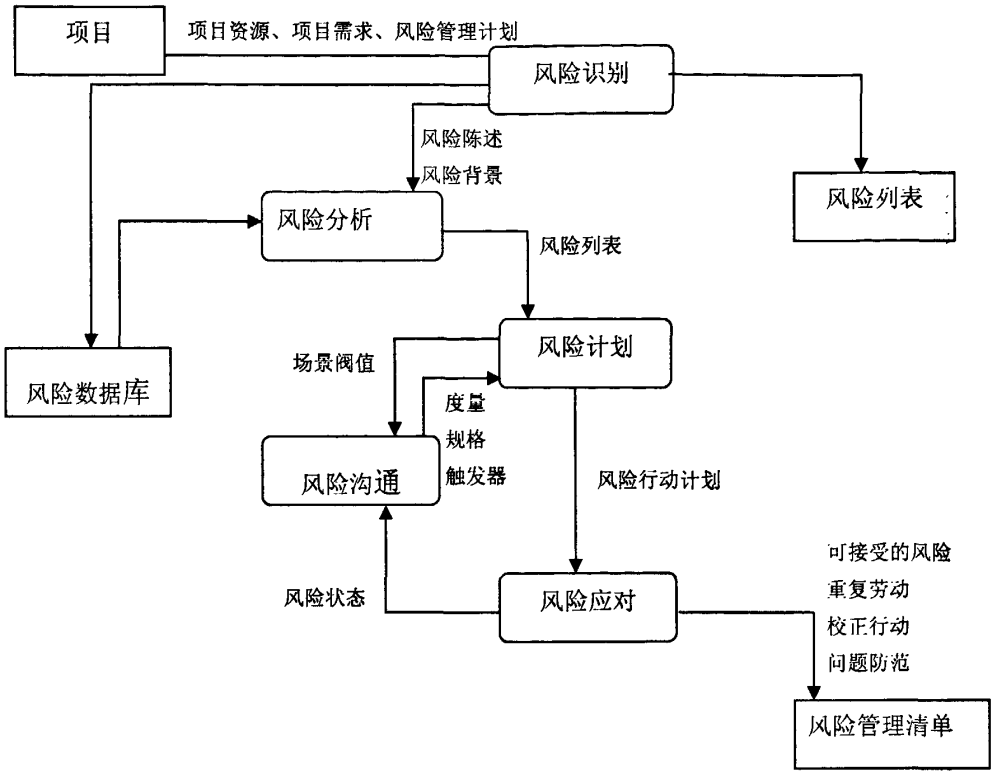

图 2.5 SEI 风险管理过程模型

Fig .2.5 SEI Risk Management Process Model

2.3.3 基于 Leavitt 模型的风险管理

Leavitt 模型由四大构成要件（如图 2.6），它们划分了各样系统的构成，这四个要件分别是：任务、角色、构成和技术。这四个构成要件可以与在项目开发过程中所遇到的各种环境很好的映射起来。角色指的是所有的参与项目开发的人员，从最终的购买者，到负责签署计划的高层管理者。构成表示这些人是以什么身份和结构来参加到项目之中。技术就是指的软件开发过程中所要用的语言、环境或者硬件的配置。任务指的是在软件开发的最后要达到一个什么样的标准才是标志着软件开发的成功。Leavitt 模型的思想是，这四个构成要件之间相辅相成，互相之间有直接的影响。也就是说，其中任一个组成要件的变化都会使其他的构成要件与之相应的变化，从而引起整个系统的变化。如果四个构成要件的状态不完全一致，可能会由此降低系统的表现性[35]。

将 leavitt 用软件风险管理的概念描述一下就是，在软件项目的开发过程

中，当其构成要件由于某些原因而发生变化，并由此会引发出一些风险，最终可能导致项目开发的某种程度上的损失。通过分析 Leavitt 模型，我们可以得出，模型的四个组成部件涵盖了在软件开发过程中可能导致软件失败的所有原因，即为风险因素。此外，四个组成部件之间的相互作用也会对系统的开发产生影响，像用户如何参与到项目的开发过程中。所以我们可以得出，Leavitt 模型是一个相对比较成熟的模型，其非常具有实用性。

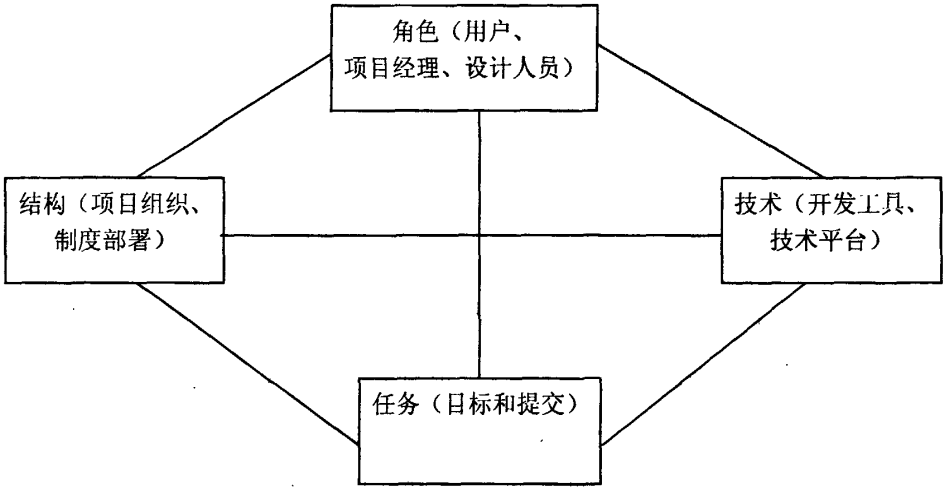

图 2.6 Leavitt 模型

Fig .2.6 Leavitt Model

Leavitt 模型只是一个大的架构，可以将其应用到软件风险管理的系统之中，将其与系统进行紧密的联合。其理论成果已经在许多的软件项目的开发中加以应用，它将注意力放在软件开发的核心步骤上，同时，又具有简单易懂，易于操作，具有很强的实用性[36][37][38]。

2.3.4 Riskit 风险管理过程

Riskit 风险管理过程从风险管理的另一个角度进行分析。它从总体上对风险的起因、可能对项目造成的结果来进行管理和控制。其使用的方法是用工具分析图来对当前进行的项目进行风险分析。它可以借鉴以前的项目经验或数据进行预测[39]。

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，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。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 2.7 所示。

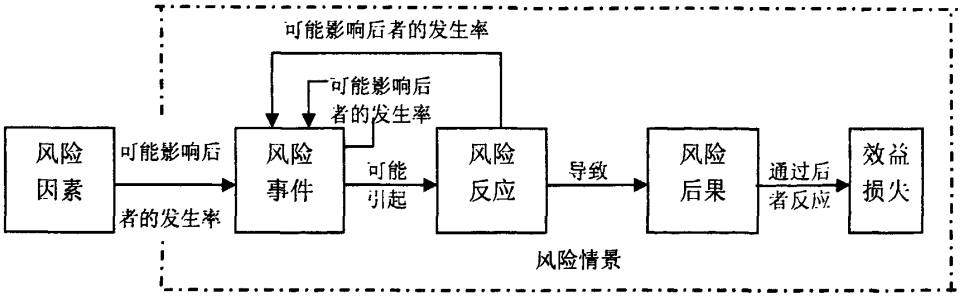

图 2.7 Risikit 分析图

Fig. 2.7 Risikit Analysis

风险因素指的是有可能引起项目损失的事件。比如说，新员工加入项目组对项目的理解程度较之前的员工可能会有差别。换句话说，风险因素可以用来抽象项目的环境变化。风险事件的定义是有可能引起项目损失的事件。它可能是由多个风险因素引起的，与此同时，也可能再作用域其他的事情或者风险因素，从而引起连续性的反应[40]。比如，高层领导的突然调动，用户突然对项目要求有大的改动都有可能引发风险事件的发生。风险反应指的是风险事件发生后为了应对风险所做的工作，反应措施的得当与否也会影响风险所带来的后果。风险后果是指风险发生后给软件开发所带来的损失。比如，项目成本超支、不能按期交付使用。它是衡量风险对项目的损失程度的手段。

在 riskit 风险管理模型中，我们用不同的色块和图形来代表各种因素，因而可以直观易懂的呈现影响软件开发的各种信息及彼此之间的作用力。以此为根本进行 Riskit 风险管理的控制。Riskit 风险管理模型含有 7 个活动过程，在一个软件的开发过程中，这些过程可以重复的顺序发生，或者并发的进行。以下是对起个活动过程的简要描述：

(1) 风险管理定义。是对风险的一个基本的界定，包括风险管理所能控制的区域，着重点应该放在什么地方，各自的任务等。并且界定出当风险出现时各自的负责人。

(2) 目标查阅。在确定了软件开发的目标之后，重新进行审视。发掘出潜

在的还没有清楚地描述出来的但同样十分重要的目标。后者将原先定义的语义并不明确的目标再加以明确化。

(3) 风险识别。可以用不同的策略或者手段发掘出可能对项目的顺利进行造成损失的因素，并将其罗列出来，用于以后的进一步识别和规划控制。

(4) 风险分析。在进行了风险识别之后，对已经识别的风险因素进行分类，生成完整的 Riskit 分析图。从而对每一个会对项目产生影响的因素有一个直观的了解。

(5) 风险控制计划。指的是在对风险分析之后，根据风险后果的严重性从上往下进行相应对策的设计。

(6) 风险控制。在风险控制计划的基础上，进行风险的管理与控制，以期在最大的限度内避免风险的发生或者将已发生的风险所造成的损失降到最低。

(7) 风险控制保证。在软件开发过程中对风险进行实时的监控，以便及时的调整策略降低软件开发的损失。

针对每个过程，Riskit 风险管理控制模型都进行了具体的描述，包括过程的定义，运行方法及每个过程所具有的特点等。之后，每个过程可能由下一级的子过程构成，然后以 Riskit 模型为基础，将各个风险元素写入文档，并按照其对软件影响的严重程度进行排序。总的来说，Riskit 模型为风险管理呈现了一个全面的指导方案[41][42]。

2.4 本章小结

风险是无法避免的，不去正面风险永远无法成功的进行软件开发。要想在软件开发过程中没有风险的出现，是不可能的。所以，在进行软件项目的时候，要敢于挑战风险。想要达到零风险的目标是不可能的。我们应该做的是将由于风险而造成的损失控制在一定范围内，与此同时尽可能的追求最大利润。风险管理就是在于风险的博弈中，平衡在项目的顺利进行和对风险的控制之间的一种管理手段。只有充分详细的分析软件风险的相关理论，才能从更深的角度挖掘出各风险因素之间的关系。并且，从以前的项目中不断地积累经验，也是有效控制风险的有力手段。

第 3 章 软件风险因素的分析及分析方法

3.1 软件项目风险识别与分类

风险识别是通过一定的方法和手段，将影响项目的风险因素识别出来，并对其进行量化的整个过程[43]。在一般情况下，风险识别由开发者等有关专家一起开展。通常是经过分析当前环境的细节，将最终目的进行分层、探讨，然后将会对项目产生作用的各种元素进行归纳整理。之后，将整理后的风险清单按照对项目结果影响的严重性进行重新整理。

3.1.1 基于分类的软件项目风险识别

分类是人们认识客观世界的基本思维方式。风险分类就是根据风险项的公共特性将风险项按照一定的组织结构整理排列[44][45]。分类的目的一是为了便于管理，二是为了便于分析风险项之间的关系。利用分类手段能够揭示风险的性质，不同的分类方法归因于人们认识事物的不同角度。在风险管理中，风险的分类合理与否将关系到下一步风险评估和风险控制等一系列活动的顺利进行，所以应该针对具体应用领域采取合理的风险分类方法。

我们经常可以使用到的分类的思想的策略有分类法、SEI 的分类学方法等。这些方法在风险识别的过程中已经被用到。关系分类法是根据风险因素对项目所造成的影响将属类相同的元素进行分类控制。将风险因素进行分组，在很多情况下可以减少风险管理的工作量，将它们作为同一个风险进行控制[46][47][48]。SEI 的分类学方法就是在学习了许多之前的开发项目之后而总结出的开发模型，可用于对风险因素进行架构。相比较可以得出关系分类法更加灵活。分类学方法的条件则相对多一点。

SEI 的分类学方法将软件项目风险分为产品工程、开发环境和项目约束三大类，产品工程包括的内容是开发活动的技术方向；开发环境包括产品生产过程中的方法、规程和工具；程序约束则关心的是本地管理无法直接控制的软件开发过程中合同的、组织的和操作的因素。在三个大类下又包含了相应的元素和元素的属性，组成了一个树形结构（其概况如图 3.1 所示）[49]。SEI 还强调风险识别是一个有计划、有步骤、系统性的跟踪过程，它贯穿于项目整个生命

期，从前一时期的一次性静态描述走向了阶段性的重复更新，从严格的结构型清单走向了由分类树和问卷调查过程的统一。

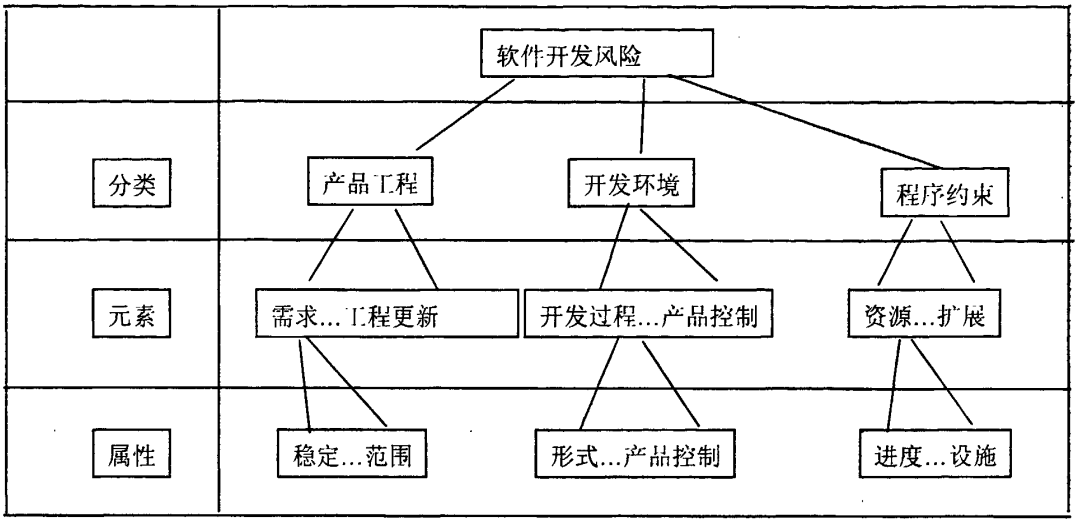

图 3.1 SEI 的风险分类结构示意图

Fig. 3.1SEI Risk Structure

SEI 的分类学方法的分类结构较为系统地刻画了软件开发过程中的风险结构，为风险项的管理提供了一个很好的框架。当然，在项目应用中，我们可以根据组织和项目的特点对这个传统分类结构进行裁剪以适应项目的需要。

3.1.2 软件项目风险因素/风险事件的分类

通过对风险识别的充分调研我们可以知道，要想得出一套万能的风险管理模型是不可能的。我们无法用单一的模型后者方案来描述出项目风险的全部因素及其基本特征。同时，如果只将单一的风险门类进行组合，并不能提高软件项目开发的成功率。所以，要想从根本上提高软件项目的成功率，我们必须从系统的宏观方面对风险进行研究。这就用到了系统工程这一学科。

从以往开发重大项目的经验来看，在研究一个应用在系统上的模型时，一般的着重点是，全局性的眼光，微观层面的分析以及各组成部分之间的相互影响。前两个方面主要是从系统的外部进行分析。最后一个着重点是从系统的内部，研究由于系统内部的各个组成部分在环境变化的过程中如何互相作用，从而进一步理解系统的模型。

从系统的角度出发，将风险进行分门别类的原则，我们研究 CMMI 等软件再将得到的成果运用到具体的项目之中。可以得到，在提高软件开发的质量过程中，能力成熟度高的公司，它们的项目开发情况应该更加平稳，更有规律并且易于管理的特点，成功完成项目开发的几率更高。此外，相对于外界的作用力而言，需要企业的内部有更强的灵活性以便与环境相配合。在研究了 CMMI 等管理体系后，及阅读了大量的有关风险管理的资料后，着眼于基于分类的风险识别，从风险因素或者其特点上、内部的相互作用力及外部对项目的影响等方面进行考虑，在此研究上给出多级化的模型结构。

通过对软件风险的分类研究，我们生成了一个基于整体的软件风险因素分类结构表（如表 3.1）。从该表中我们可以看出，将风险按照不同的组成方式分成三个模块，同时结合实际的软件开发情况再进一步细分成八个组成部分及更进一步的元素。在这个分析中，第一个模块主要从系统的内部来考虑软件风险的因素，它突出了项目组织的内部的协调能力对软件开发成功的作用力，从而可以充分的觉察到软件内部的变化，从而采取及时的措施。

表 3.1 基于组织的软件项目风险分类结构表

Tab.3.1 Risk Classification Based On Organization Structure Table

类别	组成部分	代表属性
组织过程能力	1 管理过程能力	a. 计划编制 b. 项目实施 c. 沟通渠道
	2 管理方法	a. 规范的需求规格说明 b. 正规的项目跟踪表 c. 正规的技术培训 d. 有效地激励机制
	3 系统开发	a. 硬件能力 b. 系统软件性能 c. 用户需求理解能力
组织外部环境	1 市场环境	a. 市场竞争 b. 市场需求
	2 客户方面	a. 需求变更 b. 客户间冲突 c. 知识水平 d. 相互交流 e. 协作水平
	3 社会及政策因素	a. 社会环境 b. 政策支持和导向 c. 资金和设备
产品性能方面	1 开发技术	a. 采用的开发工具 b. 相关开发软件
	2 产品设计复杂度	a. 软件功能要求 b. 性能要求 c. 接口定义 d. 硬件约束

第二个模块是从系统的的外部研究软件风险对项目的影响，这也是软件的风险识别一个关键的步骤。我们可以从竞争环境、购买力以及国家政策等方面进行考虑，重点研究在项目开发的过程中，来自项目外部的风险因素对项目开发的影响。

第三个模块讨论的是最终产品在实际运行起来的效果的风险。主要指的是产品的性能和质量的一些因素。

我们可以上述的模型进行风险调查。通过详尽的调差研究，风险管理者可以发掘出在系统开发过程中可能潜在的风险元素或者时间。从而进一步开展风险分析、控制及管理等活动。

在进行具体的软件风险管理的活动中，首先应该进行的就是对有可能对软件开发造成影响的元素进行调查。通过问卷调查的形式是一个被广泛应用的用于研究风险因素的方法[50][51]。其调查的成果一般会被加入到风险列表之中。

风险列表里面包括了许多风险因素。是通过调查问卷或其他方法所得到的原始数据。以便以后进行更详尽的风险分析时，风险列表就是分析来源。表 3.2 是常见的风险信息记录表。

我们应该注意的是，风险发生所造成的损失有时候坑内不是对同一过程而言。有时候，风险在次过程中发生，造成一定的影响，这种影响可能会波及到其他的过程。同时，我们所指的损失也是宏观的，即有时候的损失是负数。其实就是我们所说的盈利。

表 3.2 软件项目风险信息记录表

Tab.3.2 Software Project Risk Information Table

风险发现和评估：	
风险项名称	来源项目标注：
风险项编号：	风险评估等级：
风险发现日志：	生命期阶段标注：
风险项描述：	
发生概率：	
潜在经济损失：	
风险计划：	
风险出现前所需的缓解措施：	
风险出现后所需的应急措施：	
可以接受而不需要预先计划：	
最终结果：	
风险是否出现？	
如果出现，实际经济损失：	潜在时间损失：
效果总结	

3.1.3 软件项目风险因素/风险事件之间的关系

上一小节我们用基于分类的方法进行了软件项目开发的风险分析。本节我们分析风险因素与风险事件之间的作用。

在软件项目的开发周期中，一般由多个风险事件与同一个风险的发生相关联。与此同时，一个风险事件又是由若干个风险元素所组成。我们可以得出，无论是何种风险事件，其根本原因都是由组成它的风险元素所决定的。从风险元素之间关系连系的紧密程度可以划分出以下几种关系：完全独立关系、部分关联关系以及完全关联关系。

(1) 独立关系

独立关系指的是一个因素引起风险事件的发生与另一因素引起风险事件的发生没有任何关系。它们之间发生的概率没有任何关系。可以独立的导致一件

风险事件的发生或者引起任何其他的改变。当风险因素间表现为独立关系时，通常情况下，在引起风险事件的发生时它们之间没有什么必然联系。

举例来说，如果有两个风险因素 RF_1 和 RF_2 导致某个风险事件 R 发生的概率分别为 P_1 和 P_2，后果分别为 C_1 和 C_2，且两个风险因素 RF_1 和 RF_2 自身存在概率为 P_{RF1} 和 P_{Rf2}，

则该风险事件发生的概率 PR 为：

$$P_r = 1 - (1 - P_{RF1} \cdot P_1)(1 - P_{RF1} \cdot P_2) \tag{3-1}$$

两个风险因素共同导致的风险量 R 为：

$$R = P_{RF1} \cdot P_1 \cdot C_1 + P_{RF1} \cdot P_2 \cdot C_2 \tag{3-2}$$

完全独立关系是一种在软件开发过程中经常会遇到的关系。例如，风险元素“软件开发人员的大量调动”和“超出成本预算”都会导致项目不能如期交付使用的发生，但是这两个风险元素之间是没有联系的。这时就可以说这两个风险因素是独立关系，并且可以用上述的两个公式进行概率及其影响力的计算。

(2) 部分相关关系

此种关系是指，当一个风险因素导致一个风险事件发生或者引起其它的变化时，与另一个风险因素存在着一定的关系。也就是说，这两个风险因素不是独立存在的，一个的变化会引起或者受到另一个变化的影响，或者两个因素需要共同作用与一个风险事件。因此，两个风险元素之间存在着一定的概率联系。

举例来说：假设两个风险因素 RF_1 和 RF_2 自身存在的概率为 P_{RF1} 和 P_{RF2}，当风险因素 RF_1 存在时，风险因素 RF_2 导致风险事件发生的概率为 P_{12}；当风险因素 RF_1 不存在时，风险因素 RF_2 导致风险事件发生的概率为 P_2。当风险因素 RF_2 存在时，风险因素 RF_1 导致风险事件发生的概率为 P_{21}；当风险因素 RF_2 不存在时，风险因素 RF_1 导致风险事件发生的概率 P_1。在这种状况下，风险事件发生概率为 P_r：

$$P_r = 1 - \omega \cdot \upsilon \cdot \xi \cdot \psi \cdot \zeta \tag{3-3}$$

其中

$$\omega = 1\text{-}P_{RF1} \cdot P_{12} \cdot (1\text{-}P_{RF2} \cdot P_{21}),\ \upsilon = 1\text{-}(1\text{-}P_{RF2}) \cdot P_1$$

$$\xi = 1\text{-}P_{RF2} \cdot P_{21} \cdot (1\text{-}P_{RF1} \cdot P_{12}),\ \psi = (1\text{-}P_{RF1}) \cdot P_2$$

$$\zeta = 1\text{-}P_{RF1} \cdot P_{12} \cdot (1\text{-}P_{RF2} \cdot P_{21}) \cdot P_{RF2} \cdot (1\text{-}P_{RF1} \cdot P_{12}) \tag{3-4}$$

(3) 完全相关关系

此种关系是指，当一个风险元素要作用于软件开发过程或者引起某种变化时，是要以另一种元素的存在为前提的。两者缺一不可，共同存在，互相作用。当两个风险因素只有一个达到触发条件时，并不能引起变化。只有两个条件同时具备时，才会以一定的机会引起风险事件的发生。

以单向条件为例，可以描述为：当风险因素 RF_1 存在时，风险因素 RF_2 才有可能导致风险因素状态的改变或风险事件的发生，且状态改变或事件发生概率为 P_2，其中风险因素 RF_1 存在的概率为 PRF_1。在这种状况下，风险因素状态改变或风险事件发生的概率为 P_r。

$$P_r = P_{RF1} \cdot P_2 \tag{3-5}$$

我们可以从一个例子加深对这种关系的理解：某软件企业应客户要求开发一套管理系统，要求在规定的时间内交付使用，从而提高自己企业的竞争力。在此，由于开发人员对需求分析不准确这一风险元素以项目开发中要进行大规模的目标改动这一风险元素为前提。若果“目标改动”发生的几率是 0.2%，“开发人员对需求分析不准确”发生的几率是 20%，则我们可以得出由于软件开发人员对需求分析不准确而导致的在项目开发时进行大规模的目标改动这一事件发生的概率是 4%。

通过分析风险元素之间的相互依赖关系会为我们进一步明确各种风险因素在风险事件中的影响力及影响范围有很大的帮助。

3.2 软件风险评价方法

风险评价又可以称作风险预测。一般情况下，我们采用两种策略进行风险的评价。一是评估风险发生的概率或者可能性。另一种是评估当风险事件发生后所带来的损失。通常我们通过以下的步骤进行风险的评估过程：

(1) 首先要创建一个量化模型，来衡量每个风险因素发生的概率。

(2) 对风险发生所带来的损失进行定义。

(3) 预测潜在的风险的发生对软件项目的成功开发的影响力。

(4) 对风险进行检查，确保确定的风险不存在错误。

除此之外，要针对风险清单中的每个风险进行单独的分析，尽量做到详细的表述。因为针对不同的风险事件所采取的应对方法是不同的。

3.2.1 风险的定性分析法

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[52]。

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。它们的特征是不从细节入手，对风险的发生概率以及所带来的损失不做精准的预测，而是通过分析将它们分成几个层次。然后将发生的概率及损失用各自的方法进行融合从而来界定风险因素对项目开发的影响力。之后按照影响力的不同重新进行优先级的排序，以此来决定采取什么样的应对策略。

在定性风险评价中，最长被采用的是因果分析法：

顾名思义，因果分析法探究的就是引起风险事件的原因以及其产生的后果之间的关系。通过研究两者之间的相互关系，找出引起风险事件的根本原因。这样做可以从风险的萌芽就开始进行干预，以最少代价得到最大收益。在这种方法中，其思想是：一个风险事件被触发了，如果不采取措施进行干预和管理，这个风险很可能会重演。通过对过去的项目进行总结，可以防止相同的损失发生多次。由此，我们应该不断的审视之前发生的错误，并将它作为软件开

发过程中不可或缺的步骤。

因果关系分析法的步骤如下：

(1) 明确引起损失的因素。

(2) 找出合理的可以解决问题的策略。

(3) 进行有效地风险管理过程。

定性分析方法的主要任务是分析软降项目开发的各个过程中存在的不同的风险元素。软件的开发过程包括：分析阶段、设计阶段、编码设计、系统集成和系统测试。各个阶段存在的风险因素是不同的。例如：开发人员不能准确理解用户的需求；数据库的设计有问题；最终的系统不能满足用户；系统性能达不到预期的效果；成本超支；进度落后，不能按期将产品交付使用等等。按照这些风险因素发生的几率和它们对本过程带来的损失以及对其他过程的损失，可以确定这一风险因素的综合重要性。一般根据其发生的可能性分为五个级别，很高，比较高，中等，比较低，低，非常低；根据风险因素对项目的作用力可以也可以分为相应的五个等级：很严重、严重、一般、比较小、轻微。

但是，进行定性分析的方法有很多，标准也不是唯一确定的。而是要根据环境和具体开发的项目的不同进行具体的分析。例如，在分析软件系统的实施阶段的实施这一风险因素时，可以从实施域、实施难度，实施顺序等方面进行分析。

这一方法的优点是，没有统一的标准，可以根据先前的开发项目总结的经验或者凭借开发者的直觉进行评估。这样就会减少要获得风险因素的几率及其所带来的损失等要付出的成本。

这一方法的缺点是，标准衡量的作用域不宽广，很难用文档进行记录。同时，由于其标准的不统一，很难进行一些对损失估算的精准评估。

3.2.2 风险的定量分析法

定量分析法的定义是，将每一个风险因素发生的几率及其对软件开发所带来的损失进行量化，同时也应用于量化项目总体的风险度。在实施此种方法时，我们要取得发生某一事件的精准几率。量化开发过程的风险程度，从而确定财政预算和风险发生时所要追加的投资。从而找出最合理的财政预算，开发周期及开发目标。定量风险评价对数据的可靠性要求更高，否则将使定量分析

的优点大打折扣。

(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[53]。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[54]。

“龙卷风图”首先展示最敏感的变量。（如图 3.2）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。

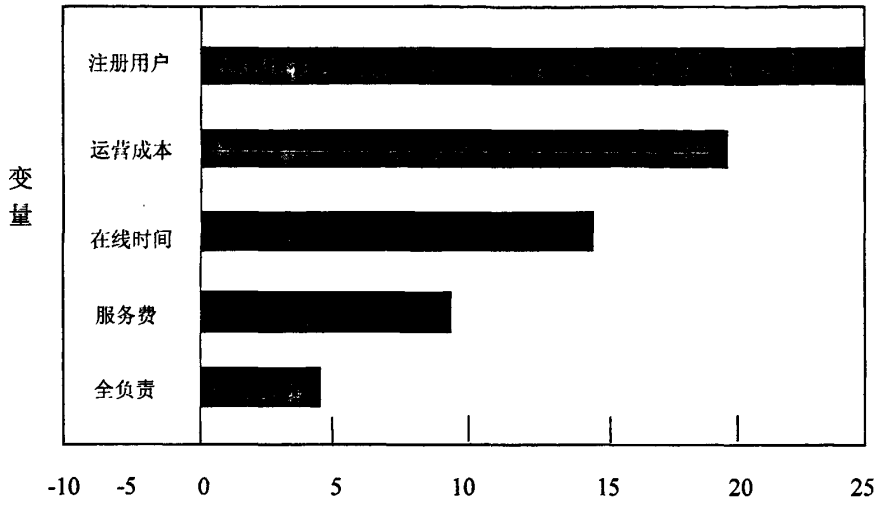

图 3.2 龙卷风图

Fig. 3.2 Tornado Diagram

“效力函数”综合了风险管理者对风险的态度将期望效力最大化而并非是期望价值。此函数代表了个体对风险的理解。当遇到风险的时候，每个人都有每个人的应对办法，对风险的接受能力也因人而异。有的人敢于挑战风险。同时，另有一些人则在面对风险的时候更多的选择了逃避。

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。参数的取值越高，也就说明个体越具有对风险的耐受力，更加挑战风险。开展敏感度的分析可以充分掌握个人对风险的态度。

③概率分析

此种方法是运用数学的方法来对软件项目开发中的各种风险因素进行评估。进行分析的主要方法一般是概率论及数理统计方法。通过概率分析可以对软件开发中的各风险因素有一个比较全面的了解。主要包括解析法和模拟法（蒙特卡罗 Monte Carlo 技术）两种。

(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。

(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[55]。

常见的风险的定量分析法：

(1) PRA 和 DPRA 分析法

这两种分析方法都是基于故障树分析法而进行分析的。其进行量化，稳定性的分析的能力在软件风险管理领域内被广泛的应用。同样，我们可以将其应用到风险分析领域。一般的步骤如下：

1）着重研究软件项目开发中的关键部位，找出潜在的可能发生风险的事件。

2）确定风险因素在软件开发中的作用于位置。理出风险因素之间的相互关系，由此建立风险因素结构图。

3）评估出个风险因素对软件系统开发的影响，并计算出其发生的概率。

4）用量化的方法对风险因素进行分析重组。如果采用后者进行分析，则还要再考虑它们在进度上的关系。

(2) VERT 分析法

VERT 分析法也是经常被应用于软件风险估计领域。它是一种通用的仿真技术，它对项目研制构造过程网络。将各种负责的逻辑关系抽象为三元组的变化。这三元分别为时间、成本和性能。网络模型面向决策，统一管理三元组等风险因素参数，可以有效地达到进行多目标最佳性能的问题，具有广泛的应用性。其思

想是运用节点逻辑功能，进行时间流，成本流和性能流的管理工作。每次仿真运行，通过运用蒙特卡罗法，这些工作流在项目结构中以一定的几率向不同的结构运动。在经过不同的结构后，会发生不一样的变化，直到项目结束。当我们进行了多次仿真之后，用收集到的数据进行分析。将有助于我们进一步的理解项目开发过程。如果模型描述的恰当，逻辑关系及其他关系无误，统计的数据的可靠性高，开发者就可以比较贴近现实的模拟出在实际开发情况下，系统开发的周期，成本及所能达到的效果的情况，从而了解软件开发的风险情况。

(3) 故障树分析法(FTA Fault tree analysis)

故障分析技术是美国贝尔电报公司的电话实验室于 1962 年开发的，它采用逻辑的方法，形象的进行危险的分析工作，具有直观明了，思路清晰，逻辑性强的特点。故障分析树法具有很大的灵活性，既可以进行定性分析，又可以进行定量分析。体现了以系统工程方法研究安全问题的系统性、准确性和预测性，它是风险系统工程的主要分析方法之一。

故障分析树是一种逻辑因果关系图，他根据元部件状态（基本事件）来显示系统的状态（顶事件）。就像可靠性框图（RBDs）故障树也是一种图形化设计方法，并且作为可靠性图框的一种可替代方法。一个故障树图是从上到下逐级建树并且根据事件而联系，它用图形化“模型”路径的方法，使一个系统能导致一个可预知的，不可预知的故障事件（失效），路径的交叉处的事件和状态，用标准的逻辑符号（与，或等等）表示。在故障树图中最基础的构造单元为门和事件，这些事件与在可靠性框图中有相同的意义并且门是条件。

(4) 风险决策树分析

风险决策树分析法是由决策树发展而来的方法。决策树是用树形结构图来表示处理逻辑的一种工具，可以直观、清晰地表达加工的逻辑要求。特别适合于判断因素较少、逻辑组合关系不复杂的情况。建立决策树的过程，及树的生长过程是不断的把数据进行切分的过程，每次切分对应一个风险问题，也对应一个节点。对每个切分都要求分成的组之间的差异最大。

一般情况下，决策树应用于确定在软件项目开发过程中某一期望之外的事件的发生。通过从风险库中提取的风险因素及它们之间的联系，画出树形结构图。

在应用数学中的概率统计进行处理后，再利用风险决策树展开进一步的研

究。其步骤如下：

1）首先进行项目需求的分析，了解软件项目的过程，之后采用工作结构分析方法确定项目目标。

2) 构造所要开发的产品的系统结构图及风险结构图。

3) 建立风险列表，并找出其中的各个风险分别对应着软件哪一方面和会产生的威胁。

4) 最大限度的预估出风险因素引发的项目损失，从而明确其重要性。

5) 评估出每个风险事件发生的几率和频率。

6) 确定风险事件为软件项目带来的损失的影响程度。

7) 评估有风险事件引起的后果所带来的损失。

(5) 外推法(Extrapolation)

外推法是进行项目风险评估和分析的一种十分有效地方法，它可分为前推、后推和旁推三种类型。前腿指运用以前开发的项目所积累的经验和结果对即将开展的工程进行预估计。如果经验结果具有明显的重复性，则我们可以以此来进行对未来项目的风险评价和管理。如果无法从历史经验中找到规律，则采用其他数学方法来综合统计来的数据再进行外推。注意，要注意之前数据并不见得是完整的客观的。

(6) 后推法

当我们没有可以用来作为参考价值的经验时，就可以用到后推法。因为有很多的软件开发项目具有明显的特点，进行重复的可能性非常小，此时，就可以用后推法进行软件项目的风险预估。后推法是在事情发生前，把其与已经发生的事件相联系。用已知事件的数据来进行未知事件的结果的分析，从而采取不同的风险应对措施。旁推法就是指利用相对来说比较相同的开发资料进行分析，用相似的项目开发数据对即将开始的软件开发工程进行分析。其中也要注意随着环境的变化，历史数据所表现出来的局限性。三种方法各有利弊，应用在不同的条件下，是项目风险管理中被广泛采用的方法。

这些定量分析的方法都有各自的特点，应用于不同的条件和开发环境下。将它们有机的结合在一起可以帮助我们进行风险的有效管理。以下是定量分析的优劣分析。

定量分析法的优点是，可以将风险事件出现的几率、对软件开发所能造成的损失、在各个过程中，风险事件发生的频率及其影响力进行量化。从而有效地提高了对风险预估的准确性，从而帮助软件开发人员进行更有效的风险管理工作。

定量分析法的缺点是，很难保证对风险事件出现的几率的评估是准确的。并且由于环境是在不断变化的，这些评估的数据稳定性很差，因为这些数据是由之前的开发人员凭借他们的经验进行总结的，缺乏科学的论证。为了使统计得来的数据具有更高的可用性，我们应该多采用与当前开发环境一致的数据，尽量保证数据真实有效。

3.3 影响软件风险的因素

由于软件的开发过程任务繁多，过程复杂，所以我们不可能注意到开发中的每一个细节，因此要想让软件开发顺利进行，必须将重点放在影响开发成功的重要环节上。因此，经过总结，组织结构，开发过程，基础结构和运行这几个环节是软件开发的重点。P^2I^2 模式描述了上述几个因素所产生的影响。如图(3-5)。

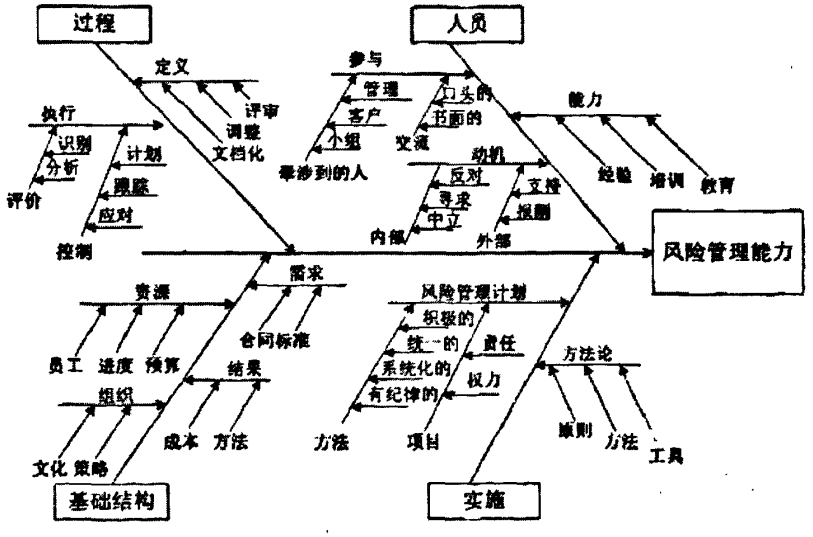

图 3.3 P^2I^2 成功模式图

Fig. 3.3 P^2I^2 Model

3.4 本章小结

本章主要对软件开发中存在的风险因素进行了分析，并且介绍了分析所使用的方法。目前，被广泛应用于风险分析的方法是风险定量分析和风险定性分析方法。在本章中，我们分别介绍了两种方法的基本概念和相关理论，在比较了各种方法之后，我们给出了两种方法的优缺点。此外，我们综合了定量分析和定性分析方法，对风险因素对项目所造成的影响展开进行了研究，并提出了相应的解决对策。

第 4 章 建立风险管理的模型

4.1 问题的描述

现代质量管理理论和核心内容是过程的管理，ISO9000 定义所谓过程就是“将输入转化为输出的一组彼此相关的资源和活动”。在软件企业中我们将过程分为两类:即管理过程和软件工程过程[56]。管理过程与软件过程过程不同，前者是企业级的需求，而软件工程则是就单一项目而言，指的是一个项目的开发周期，如系统功能结构设计、数据库设计等。有时候一个活动就能代表一个过程，而另一些过程则是由多个分别的活动组成。这些活动之间存在着一定的联系，并且相互之间会产生影响。本章我们着重放在对软件开发过程的分析与控制方面，而不是开展过程所用到的方法。我们所要完成的就是以软件项目的风险管理为中心，以风向控制为目的的研究。

在软件开发的各个阶段中，初始阶段项目会遭受失败的可能性很高，因此风险会集中出现在此阶段。但是，由于在这个阶段，各种资源的投入都十分有限，所以即使风险出现，所带来的影响也是最轻微的。随着软件开发过程的慢慢进行，失败的可能性就逐渐变小。此时风险发生的几率也逐渐变小，但是相反的，一旦风险事件被触发，所带来的影响通常都是巨大的。图 4.1 对上述问题有了一个直观的描述。

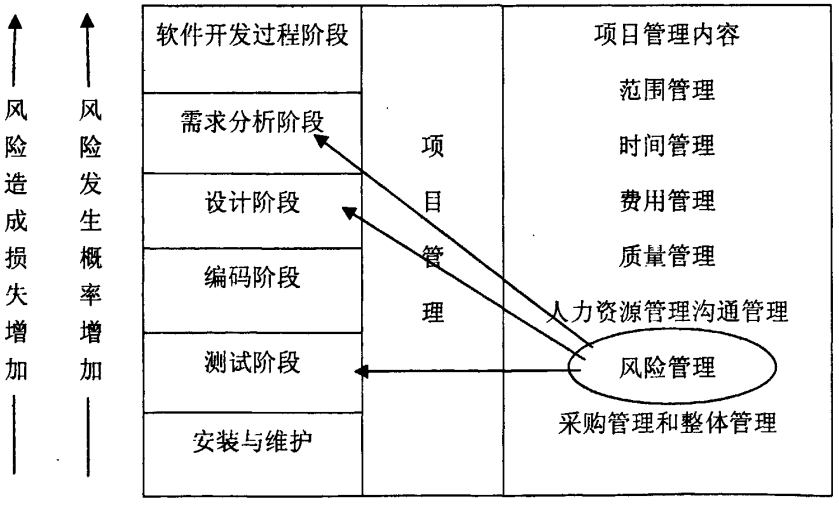

图 4.1 软件开发过程和项目管理的关系

Fig. 4.1 Relationship between Software Development and Project Management

4.2 对软件项目管理经典模型的改进

美国卡内基•梅隆大学软件工程研究所经典的 CRM 模型指出在软件开发的过程中都要进行软件风险的监控和评估。并且指出这是一个在软件进行过程中要重复进行的步骤。每个风险元素基本上都要按照一定的次序经过这些过程。只是它们进行的活动可以是同时或者间或进行的。

CRM 模型所具有的同时进行或者间或进行的特点也有其不好的方面。表现在几个模块之间的联系不明确，在进行过程中不能用一个时间序列进行计算。一个模块即可以作为输出，也可以反过来再从输出的模块就行接受。对于经验不是很丰富的开发者而言，无论是理解还是具体的操作都有一定的难度。因此，我们需要找出一个办法，对模型进行简化。使其可以用一个时间序列进行计算，并将每个模块具体化。再分次重新组合的模型我们可以明确的观察各个风险元素在一个开发过程中的全部运动，这样有助于我们为每个模块单独进行分析，从局部制定风险管理策略。同时，我们也从下向上对风险管理有不同的认识，从而更好的把握风险管理和控制。

改进之后的模型由以下几部分组成：风险识别、风险评估、风险应对、风险管理。我们用图示来展示几个模块以及模块之间的关系。在每个模块中，各个过程通过模块的输入和输出进行交互。它们之间的联系是这样的：

输入——文档或其他类型的文件，触发活动的命令。

输出——通过得到的输出在完成指令后将得到的结果输出给下一个模块。也是文档或其它形式。

(1) 风险识别过程

制定风险清单，将有可能影响软件开发的因素罗列出来，并详细的记录它们的特点，为以后的风险分析部分进行数据源的建立。如图(4-2)所示：

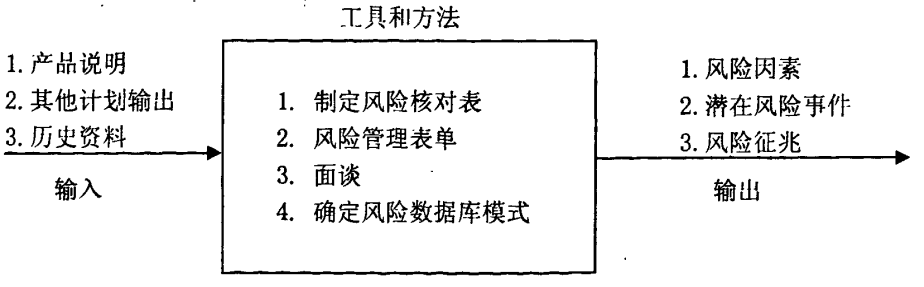

图 4.2 风险识别

Fig .4.2 Risk Identification

(2) 风险评估过程

采取适当的方法进行对风险因素进行分类。我们常用的方法有风险定性分析方法和风险定量分析方法，然后确定风险的优先级别。

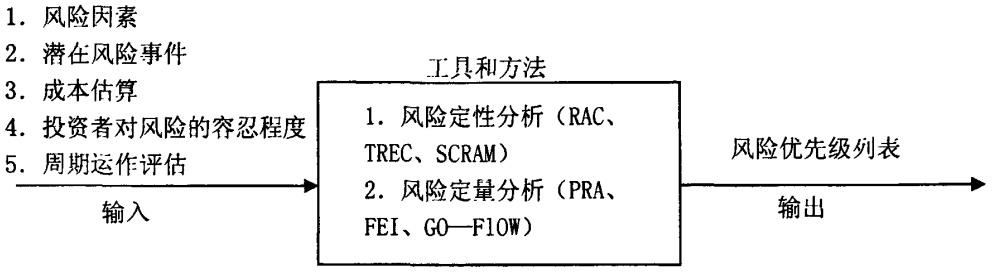

图 4.3 风险评估

Fig. 4.3 Risk Assessment

(3) 风险应对过程

根据风险评估得到的分析结果制定应对风险应对计划。

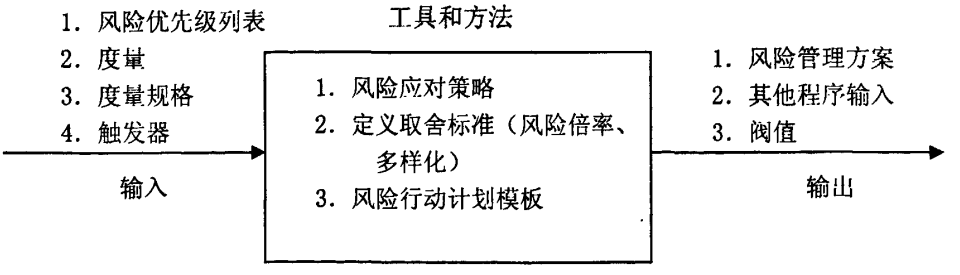

图 4.4 风险应对

Fig. 4. 4 Risk Reflection

(4) 风险管理过程

根据制定的风险管理方案进行风险的管理过程，校正错误行为，修正软

件开发的方向。

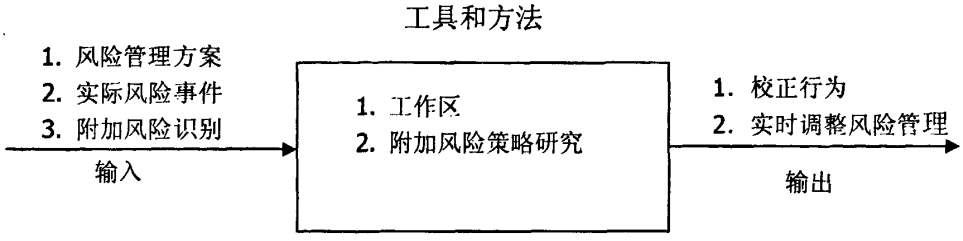

图 4.5 风险管理

Fig. 4.5 Risk Management

在系统内部，这四个模块会发生相互的作用。同时，模块的活动需要很多的资源与其相配合，并且，在每个软件项目的开发过程中，这些模块会重复出现。

为了便于理解，在这里我们将这些模块划分边界。然而在现实的软件开发过程中，它们之间的界限并不是这么清楚，往往会相互渗透，相互影响，在具体的环境中，我们会加以说明。

4.3 基于贝叶斯网络模型的建立

贝叶斯网络的建立是一项比较复杂的、困难的工作，它没有现成的规矩，只能根据实际问题，依靠相关领域专家的经验建立。对于软件风险评估模型的建立，我们必需综合考虑开发过程中各种信息源及其之间的相互作用关系，建立相应的网络模型，并根据以往开发过程中各种信息源之间相互影响程度的大小，确定网络中各结点的条件概率(即在父结点处于某种风险模式时，子结点发生某种风险模式的条件概率)。作为一个实用的贝叶斯网络，我们必须明确定义每一个结点，决定它所有可能的状态。另外，对每一个结点需要为每一种可能的状态确定一个条件概率。整个网络包括所有的结点及其条件概率分布。

4.3.1 确定变量集和变量

在构建模型之前我们首先要进行风险因素的调查。在本文中所用到的风险因素来源于一个实际的工程项目，在项目伊始，对搜集到的风险元素进行模拟，确定其对软件开发的作用力，从而建立相应的风险模型。该项目的开发人员来自多

个软件公司或者相关的组织，具有丰富的软件项目开发经验，并且分别参与过多个重大项目的开发。

通过对风险因素进行模拟分析，生成风险清单，并且根据其对项目的影响度进行排序，最后罗列对项目有至关重要影响力的风险元素如下：

技术掌握有所欠缺；项目成员对项目重视程度不够；管理混乱；开发进度缓慢；对客户需求不明确；团队之间协作效率低下；缺乏组织成熟度；反复需求修改；

4.3.2 风险局部模型网络结构

在这一节我们从局部出发，进行风险模型的建立。模型会反映风险因素之间的作用关系，在各个模型中，对每一个风险因素都进行了详细的描述，同时，也分析了风险因素之间的相互作用和它们引发的风险事件所带来的影响。我们用因果图来表示这种关系，有些可以作为局部的贝叶斯网络模型来分析。此外，我们用这些分散的模型组成了一个完整的贝叶斯模型。并且，提出了简化局部模型的方案，从而得到一个风险决策管理模型。

首先，我们来说明一下各种符号所代表的意义。各节点所代表的意义不同，所以我们用不同的符号来表示不同的节点。如图 4.6：

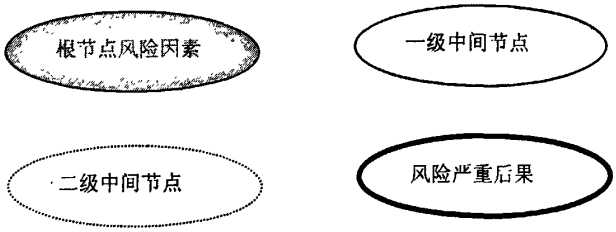

图 4.6 节点图例

Fig .4.6 Node Model

根节点风险因素是来自风险清单，是触发风险事件的风险源。从风险源到最后产生的结果，中间还有一些过程，我们就用第一、二级节点进行描述，这样有助于分析中间的过程。两级节点之间的区别在于前者在模型简化后仍然作为模型的构成部分，而二级节点就被相应的忽略掉了。后果节点，顾名思义，是用来描述由风险因素所造成的损失的，是进行评估后得到的。相对于根节点，它是叶子

节点。

有的风险因素在项目开发的不同过程所发生的作用是不同的。例如，“对用户的需求理解不充分”，在开始阶段，因为目标的不明确要与顾客不停地沟通，会影响项目的进展，从而导致不能按期完工。由于出现分支，所以我们要在图中加入条件判断。但是在最后的贝叶斯风险模型中我们会进行替代。

我们现在来分析一下这些风险因素之间的关系:

(l) 开发技术欠缺

在开发某些大型系统时，由于需要的人员数量很大，有些开发者的开发经验不充分，可能会对项目造成潜在的影响。比如，当前软件开发中比较尖端的技术，由于开发人员素质有限，虽然知道新技术可以带来更高的利润，但是由于缺少必要的知识，而无法实施。

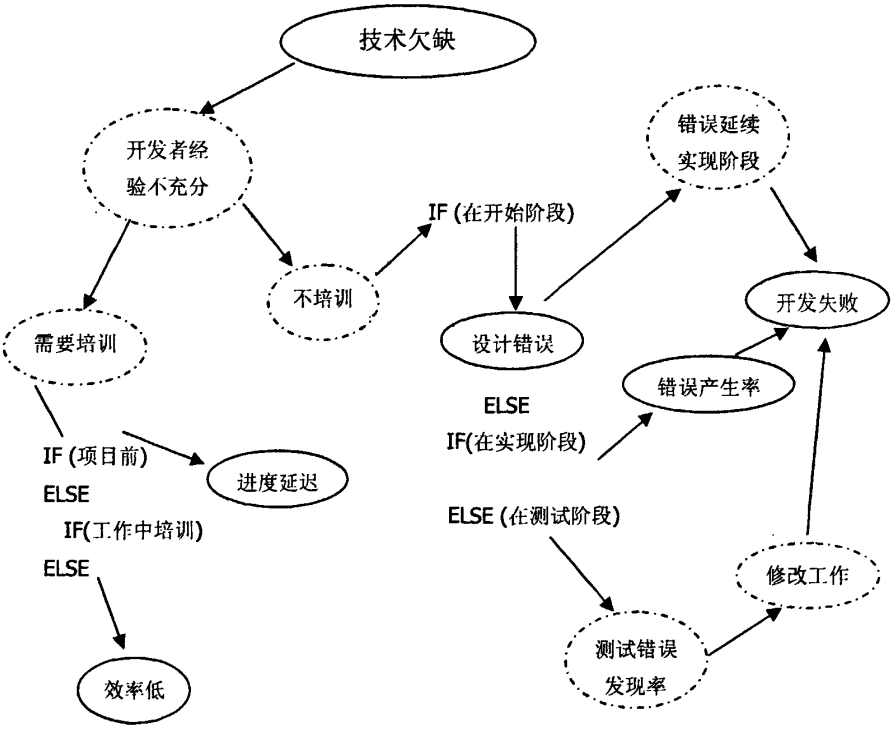

图 4.7 开发技术欠缺模型

Fig. 4.7 Lack Development Technique Model

(2) 项目成员对项目重视程度不够

由于没有进行有效地组织与培训，并且缺乏明确的纪律约束，开发人员缺干劲，不能有效地进行工作。此外，因为缺乏纪律感，以团结为特点的软件开发必然会受到影响，各自为战必然不会有利于软件的开发。开发人员不愿意掌握最新的开发方法，固步自封。

(3) 管理混乱

软件规模越大越不利于人员的管理，由于组织间的不协调必然会影响项目的进展，从而导致开发周期及费用的增加，还有可能导致项目失败。如图 4.8:

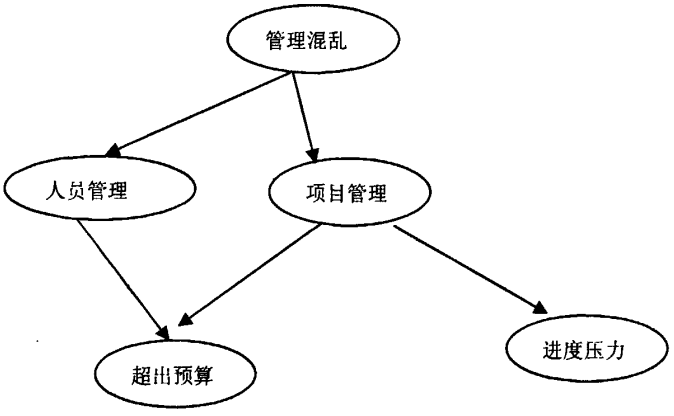

图 4.8 管理混乱模型

Fig. 4.8 Confused Management Model

(4) 对客户需求不明确

通常情况下，用户不懂得软件开发过程，所以在对自己的需求的描述时会有所出入，这就为以后频繁的更改需求埋下了伏笔。而需求的频繁修改必然会影响软件开发的进度，还会增加成本。另一方面，这也是由于开发方不能有效地引导客户进行自我的需求分析，或者是调研的不够彻底深入。在协议中由于双方的权利与义务不明确，会导致利益的边缘化模糊，从而引起一些不必要的矛盾。

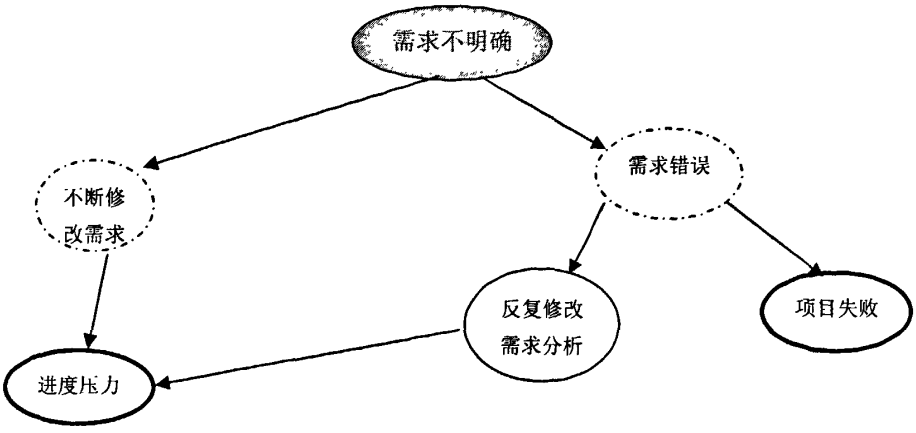

图 4.9 对客户需求不明确

Fig. 4.9 Lack of Customer Requirement

(5) 团队之间协作效率低下

软件开发不同于其他的商品生产，它对集体的凝聚力要求较高，一个协调的团队往往能创造超出预期估计的成果。如果团队内部不能团结一致，或者缺乏沟通必然会为软件开发带来负面的影响。此外，对公司的依附感，公司的制度是否完善，与用于之间的沟通是否有效也是需要注意的因素。

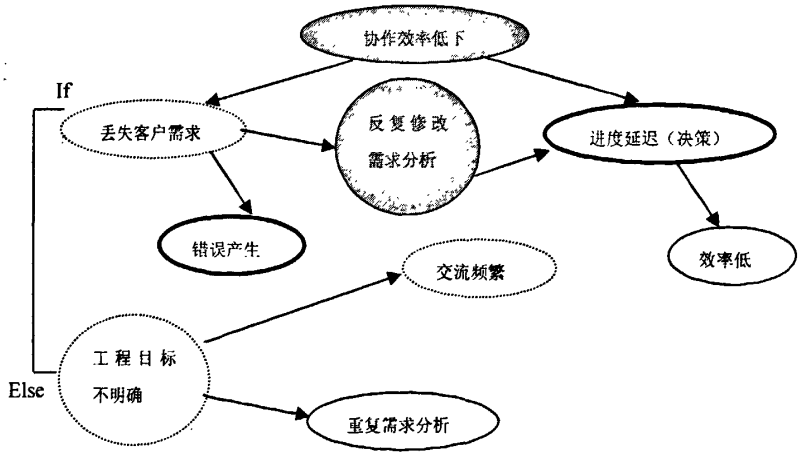

图 4.10 开发者缺乏交流风险模型

Fig. 4.10 Developer Lack Risk Model

(6) 开发过程混乱

软件开发人员对于应用于软件开发的环境，技术没有充分的认识和掌握，必然会导致软件开发的质量，甚至会影响软件的后期维护工作。

开发过程中，如果没有一个明确的开发流程，则会在开发过程中出现许多的问题，例如：文档没有及时记录和更新；不做版本控制，混乱的代码库和开发环境；在项目过程中随意的更改开发工具和环境；开发人员不做单元测试。

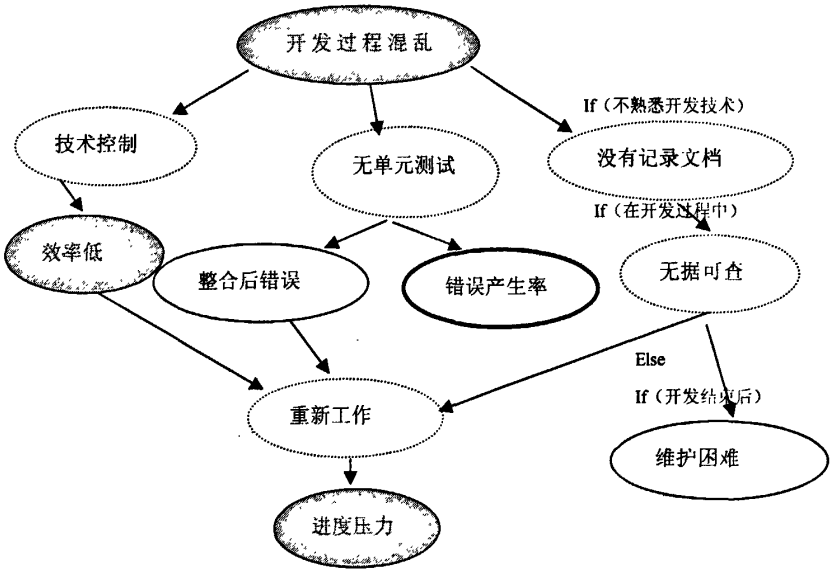

图 4.11 开发过程混乱

Fig .4.11 Confused Development Process Model

(7) 反复需求修改

这种风险一般出现在需求分析之后。由于环境的变化或者业务有了新的需要，用户的要求需要发生改变。其所带来的损失要根据后加需求的大小及需要进行改动的工程量的大小。通常，这是由于在需求分析阶段，客户不能正确的认识或者整理出自己的需求，导致最初的需求并不完备。另一方面，软件开发人员在需求分析阶段充分的了解用户的需求，也是导致需求反复修改的原因之一。

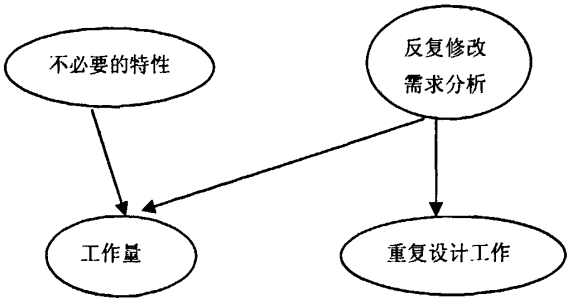

图 4.12 反复需求修改模型

Fig.4.12 Need Revised Model

4.4 本章小结

在前几章进行了风险的基本概念的描述，并介绍了风险分析方法的相关理论知识，为这一章的模型建立进行了理论铺垫工作。这一章，我们通过分析基本风险因素，在前章的铺垫下，建立了基于贝叶斯模型的风险管理模型。并分析了各个风险模块。由于与风险系统的设计工作的展开。

第 5 章 软件风险分析原型系统的设计

5.1 系统实现技术

5.1.1 JSP 技术

JSP 是基于 Java Servlet 以及整个 Java 体系的 Web 开发技术，利用这一技术可以建立安全、跨平台的先进动态网站。JSP 以 Java 技术为基础，在许多方面做了改进，具有动态页面与静态页面分离、能够脱离硬件平台的束缚、以及编译后运行等优点[57]。

JSP 的设计目标主要是提供一种更为简便、有效的动态网页编写手段，并且增强网页程序的独立性、兼容性和可重用性。它是通过以下几个方面加以实现的:

(1)简便性和有效性

JSP 技术是在原来的 HTML 网页中加入一些 JSP 专有的标签，或是一些脚本程序。这样，一个熟悉 HTML 网页编写的设计人员，可以很容易进行 JSP 网页的开发，而且开发人员完全可以不自己编写脚本程序，只通过 JSP 独有的标签，就能利用已写好的部件来实现动态网页的编写。

(2)程序的独立性

JSP 拥有一般的 Java 程序的跨平台的特性，换句话说，就是拥有程序的对平台的独立性，即“编写一次，到处运行”。

(3)程序的兼容性

JSP 中的动态内容可以各种形式进行显示，所以它可以为各种客户提供服务。

5.1.2 J2EE

J2EE 即 Java2 Platform Enterprise Edition，其英文定义为“Open and standard based platform for developing, deploying and managing n-tier, web-enabled, server-centric enterprise applications”，也就是说，它是开放的、基于标准的平台，用于开发、部署和管理 N 层结构，面向 Web 的、以服务器为中心的企业级应用，换句话说它是一套体系结构，而不是一个具体软件，是一个方便于服务器方应用程序开发的中间件服务集。J2EE 技术的基础就是核心 Java 平台或 Java 2 平台的

标准版，J2EE 不仅巩固了标准版中的许多优点，例如"编写一次、随处运行"的特性、方便存取数据库的 JDBC API、CORBA 技术以及能够在 Internet 应用中保护数据的安全模式等等，同时还提供了对 EJB（Enterprise JavaBeans）、Java Servlet API、JSP（Java Server Pages）以及 XML 技术的全面支持。其最终目的就是成为一个能够使企业开发者大幅缩短投放市场时间的体系结构[58][59]。

5.1 功能设计

该系统的最终用户是软件开发的管理人员。系统将风险分析阶段的风险因素的发生概率，影响力等参数作为输入，按照一定算法对可能产生的风险进行预测。系统的风险源来自于风险分析阶段生成的风险库，里面对每个风险因素进行了详细的分析，包括它们的发生概率，对系统造成的影响等。为了方便以后的软件开发，系统将对风险因素的评估结果保存在风险数据库中，为以后的软件开发积累经验和历史数据。

除此之外，系统的用户还可以根据开发环境的变化，随时对系统进行更新等操作，以使系统保持最优状态。

下面描述系统需求：

(1) 系统参与人员

系统用户，也是软件的开发人员，将出现的风险进行录入，生成表格，以便进行进一步的分析，并定期对其进行维护，保证其以最优状态提供服务。

项目负责人，负责软件开发的大方向的制定，对正在开发的系统进行风险因素的分析与预估从而得出其潜在的可能存在的风险。

风险库，其作为外部的辅助工具。用于记载风险分析及建模所需要的各种风险因素。其中详细的记载了个风险因素的特点及其会产生的影响。同时，风险库还负责提供历史数据记载的任务。

(2) 维护修改风险模型

其任务是对风险库进行控制和管理。需要使用到修改指标与权重和修改指标与问卷两个用例。其中会对模型进行的操作有，修改风险因素标准，更新标准与调查报告之间的关系，更新指标在模型中的排序。

(3) 风险分析与评价

首先，对于新的软件开发项目，系统要求系统运行者收集相关的资料。例如：

开发项目的名称、开发周期（开始与结束）、项目成本等。这些资料作为开发项目的基本资料将有助于软件开发的进行。帮助我们了解系统的需求，有助于软件项目的开发。同时也可以将已经完成的开发过程导入新的软件开发过程，以减少重复劳动所带来的损失。

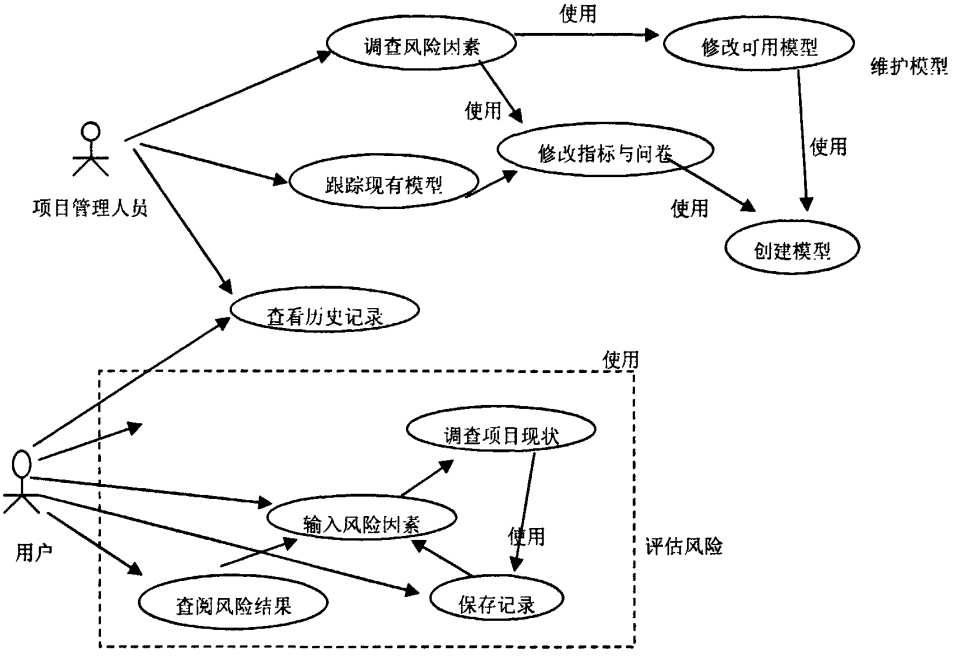

图 5.1 原型系统需求的 UML 用例图

Fig .5.1 System Need on UML

查阅风险结果，指的是项目相关人员可以根据需要查阅对风险的分析等内容。项目管理者可以查看各个风险因素。

保存记录，将项目进行过程中进行的对风险的应对操作进行保存，以便查阅和作为历史数据服务于以后的软件项目。

(4) 其他功能

查看历史记录，系统管理人员可以根据需要调出以往的软件开发过程中的相关信息，进行分析，以便新系统的开发。另一方面，历史数据为修正风险管理系统提供了一定的借鉴。

创建模型，根据系统管理员的输入条件调用风险库中的分析结果建立模型，

以便系统开发人员更好的进行系统的开发。

5.2 系统设计

此风险管理系统不是一个联网的系统，可以在离线的环境下进行操作。所以此系统无论是对硬件还是软件平台的要求都不是很高。因此我们选择 Windows XP 作为系统环境，以 Java 作为开发工具，JSP 来做前台页面，以期提供一个良好的交互式的系统。

我们选择 SQL Server2000 作为我们的数据库系统。主要用来存储各种分析的数据，例如，风险因素的各项指标，产生的影响，它们之间的关系等数据，数据构成不复杂，SQL 可以很好的满足系统的需求。

在设计软件风险管理系统时，由于风险因素作用而导致的风险事件的发生的概率不完全一样。同时，由于风险判定有主观的因素在里面，所以评估的软件风险因素的影响力也会所差异。上述的种种因素都会对软件系统的开发产生影响，所以我们都要加以考虑。

经过前面的分析我们可以得到，关键技术的掌握程度、系统的难易程度、模块之间的相互关系、开发技术等都在技术风险的范畴内。同时，在开发的过程中，并不就是单一的技术在应用，随着交叉学科的发展，技术的混合使用，各自发挥其所长的趋势也越来越明显。这就要求开发人员不能只是单单的掌握一门技术，他们需要对多种技术都有深入的了解。这就造成软件开发人员水平的高低之分，这样就不能单单的将项目的开发人员的水平做同等的处理。在这里，我们采用将每个开发人员的重要度求和，然后根据小组总共的参加人数求平均数。由此得出的结果再进行风险的分析。如果超过系统对技术风险的预评估数，则认为这个风险的发生是在控制之内的，软件项目可以顺利的开发。其他的风险因素也基本上采用此种思想。

5.3 本章小结

本章结合前面对风险的概念的描述，各种经典模型的剖析再加上风险分析方法等的介绍，再次基础上使用面向结构化的思想构造出一个软件风险模型。并用语言加以描述。之后根据前面的成果对风险管理系统的功能进行了阐述，并将其实现。

结论与展望

实际上，在开始这篇论文的杜撰前，我并不是十分了解风险管理的相关内容。我本来的主要研究方向是项目管理，但是随着风险管理的重要性日益增加，研究如何对在项目进行中出现的风险加以控制和管理从而提高项目开发的成功率成为了迫在眉睫的问题。由于其广泛的应用性，我选择这一课题开始研究。在查阅了大量的文献和各种调查报告后，我对风险由了进一步的认识。本篇文章就是我对风险管理的心得。

本文的主要工作有：

(1) 汇总关于风险管理的各种基本概念和相关理论，将它们加以整理，分类，为将从事这方面工作的人提供准确的信息。

(2) 在阅读了大量的参考文献并分析了多种经典的风险模型后，综合它们的特点力图构造心得模型。以改进原先模型的不足。

(3) 本篇文章主要研究的是风险管理系统的建立。我们从各个层面分析，研究各种风险因素在不同的条件和模块下对软件开发的影响程度，并根据分析的结果建立模型，开发系统，以期将风险的管理提高到一个新的层次上。

虽然风险管理从上个世纪就开始发展，但是我国的软件行业起步较晚，在风险管理方面的研究才刚刚开始，因此其自身还有很多需要改进和探索的地方。在理论方面，虽然将有关风险的概念加以清晰化，并且提出了多种风险分析的方法，但是对各种风险的评估的量化并不十分准确。还停留在一个很模糊的阶段。这时有经验的开发人员对项目的影响就会很大，需要依靠他们的经验来进行项目风险的主观评估。这样就提高了系统开发的准入门槛。本文从风险管理方面入手，对所有的可能对项目开发产生影响的风险因素进行分析，并尽可能的进行精准的量化，减少对少数的有经验的开发人员的依赖，使系统具有更广阔的应用性。

在目前的系统设计中，虽然完成了风险分析模型的建立，各种风险因素的分析，但是还没有使这些过程完全的自动进行，智能化程度不高。

根据以往的系统的不足，本系统加以改进。但是距离可以大规模的应用还有许多问题亟待解决。还有许多其他的技术可以应用到系统中，我将继续研究，希望可以早日设计出智能化足够高的风险管理系统，从而提高软件开发的成功率。

参考文献

[1] 孙艳春, 陈向群, 赵俊峰. 管理软件开发项目:通向成功的最佳实践.北京: 电子工业出版社, 2002. 1.

[2] Roger S. Pressman. 软件工程实践者的研究方法.北京:机械工业出版社,1999.

[3] R. P. Higuera and Y. Y. Haimes, "Software Risk Management" Software Engineering Institute, Carnegie Mellon University CMU/SE-96-TR-012, June 1996.

[4] 马国丰, 王爱民, 屠梅曾. CCM: 一种基于 TOC 的项目管理技术. 系统工程理论

[5] C K Chang, M Christensen, Tao Zhang. Genetic algorithms for project management. Annals of software engineering, 2001, Vol(11). pp107-139.

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

[7] Boehm B. W., Software risk management, IEEE Computer Society Press.CA,1989.

[8] Boehm B. "A Spiral Model of Software Development and Enhancement." IEEE Computer 21(5)(1988):61～72,1988.

[9] Defense Systems Management College. Risk Assessment Techniques. Fort Belvoir, VA, 1983.

[10] 杨立群.软件开发项目失败的原因,中国计算机学报,2001.11.

[11] Standards Australia, AS/NZS 4360: 1999, Risk Management, 1999.

[12] Air Force. Software Risk Abatement. AFSC/AFLC plmphlet800-45.Wright Patterson Air Force Base. OH: Air Force systems Command, Air Force Logistics Command,1988.

[13] VanScoy R. Software Development Risk: Problem or Opportunity. Technical report CMU/SEI-92TR-30,PA: Software Engineering Institute, Carnegie Mellon University,1992.

[14] 薛华成,《管理信息系统（第三版）》,清华大学出版社,1995,5.

[15] D. W. Karolak, Software Engineering Risk Management IEEE Computer Society, 1977.

[16] 谢科范,《技术创新风险管理》,河北科学技术出版社,1999,1.

[17] G. G. Roy, ProRisk User Guide: Murdoch University, School of Engineering Science, 2003.

[18] D. Vose, Risk Analysis: A quantitative guide, 2 ed: John Wiley & Sons, 2001.

[19] Choi, K. and Bae, D.-H., "Dynamic project performance estimation by combining static estimation models with system dynamics". Information and Software Technology, 2009.51(1): p. 162-172.

[20] 王青, 基 ISO9000&CMM 的软件质量保证模型, 中国科学院软件研究所.

[21] Coper G F. The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks, Artificial Intelligence, 1990, 42(2-3); 393-405.

[22] 郭红芳,曾向阳.《风险分析方法研究》,计算机工程,2001 年 3 月,第 27 卷 3 期,131-132.

[23] 胡晓明,《基于贝叶斯网络的软件风险管理问题的研究》,南京理工大学,硕士,2004.

[24] FergusO' Connell How to Run Successful ProjectsIIIThe Silver Bullet, IEEE Trans. Software Eng.2002, 8.

[25] 胡兆勇,屈梁生.一种贝叶斯诊断网络的拓扑结构.2003.37(11):1115-1119.

[26] Humphrey W. A Discipline for Software Engineering. Reading, MA: Addison-Wesley, 1995.

[27] Charniak E, Bayesian Networks without Tears, AI Magazine, 1992, 12(4): 50-63.

[28] Charette, R. N., "Why software fails", IEEE Spectrum, 2005. 42(9): pp. 42-49.

[29] Pandian, C. R., Applied Software Risk Management: A Guide for Software Project Managers, Auerbach Publications, 2006.

[30] 何云,项目风险管理研究,http://it.icxo.com/htmlnews/2004/04/27/200677.htm,2004,4.

[31] 赵强,乔新亮,J2EE 应用开发. 电子工业出版社. 2004.

[32] Shachter R D. Probalilistic Inference and Influence Diagrams. Operations Research, 1988, 36(4):589-605.

[33] R. Max Wideman, Software Project Risk Management, Success and Training, AEW Services, Vancouver, BC, November 2002, 2-4.

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

[35] Houman Younessi Object-Oriented Defect Management of Software, IEEE Trans. Software Eng.2002,8.

[36] 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.

[37] 潘春光,陈英武, 面向对象软件风险评估方法研究.计算机工程与应用.2004,Vol(40),32.

[38] Aletr, S. Implementation risk analysis. TIMS Studies in Management Sciences13,2(April 1979),103-119.

[39] 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.

[40] Karolak D W, Software Engineering Risk Management, IEEE Computer Society Press, 1996.

[41] 张庆华，浅谈软件开发中的风险管理，科技情报开发与经济，2004.

[42] 刘晓华，J2EE 企业及应用开发，电子工业出版社，2003.

[43] 方德英,IT 项目风险管理理论与方法研究,天津大学博士学位论文,2003 年.

[44] Kalle Lyytinen, Lars Mathiassen, Janne Ropponen. A framework for software risk management. Journal of Information Systems(11)4:275~286.

[45] 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.

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

[47] 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.

[48] Mark C. Paulk, Bill Curtis, Mary Beth Chrissis, Charles V. Weber, Capability Maturity Model SM for Software, Version 1.1 Pearl J., Probabilistic Reasoning in Intelligent System: Networks of Plausible Inference, San Mateo, CA, Morgan Ksufmann, 1988.

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

[50] Bob Hughes and Mike Cottrell: Software Project Management, Third Edition, McGraw-Hill International(UK).

[51] 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.

[52] 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.

[53] Capers J. Minimizing the Risks of Software, Cutter IT Journal; The Journal of Information Technology Management, 1998, 11(4):13-21.

[54] 胡晓明.基于贝叶斯网络的软件风险管理问题研究.硕士论文.11-16 页.

[55] 孙艳春，陈向群，赵俊峰.管理软件开发项目:通向成功的最佳实践.北京：电子工业出版社.2002.

[56] 杨爱华王静运用风险管理避免软件开发的失败项目管理技术.2004(2):pp23-24.

[57] Raymond F. Quantify Risk to Manage Cost and Schedule. Acquisition Review，Vol.6，No.2，PP.147-155.1999.

[58] 文亚栋.软件项目的风险管理.计算机系统应用.2002,17(2):72-74.

[59] Osamu Mizuno, T Adachi, Tohru Kikuno, Yasunari Takagi. On Prediction of Cost and Dura-

tion for Risky Software Projects Based on Risk Questionnaire. APAQS 2001:120-221.

致　　谢

本文是在我的导师陈如亮的精心指导下完成的。两年来，陈如亮老师在学术上渊博的学识、严谨的治学态度、敏捷的思维方式令我受益匪浅;工作中，老师的敬业奉献精神，诲人不倦的作风和虚怀若谷的态度，都是我今后为人处世的表率。生活中，老师也给予我无微不至的关怀。陈如亮老师不仅教导我如何做学问，更教会我如何做人，我不会辜负导师们的厚望。

感谢实验室的同学们，在于他们的探讨和交流中，使我学到了很多知识，因此才能够顺利完成我的毕业设计。同时，感谢他们在两年多的研究生期间对我生活方面帮助和关心。

感谢所有在我学习期间关心和帮助我的人。