

TP311.52/212

2009



普通高等教育“十一五”

国家示范性软件学院系列教材

# 软件项目管理 案例教程

第2版

韩万江 姜立新 编著 宋茂强 审

软件企业的实际案例  
贯穿始终

- 项目管理理论：全面、系统
- 项目管理运作：客观、易懂
- 项目管理案例：真实、可操作



机械工业出版社  
China Machine Press

# 序

进入 21 世纪，人类社会进入了信息经济时代，信息技术已成为所有高新技术发展的支撑技术。同时，信息技术及网络的高速发展，极大地推动了社会、经济的进步和人的思想、观念、文化的变革。其中最为显著的变化是推动了经济的全球化；反过来，经济的全球化又对信息技术及网络提出了更高的要求。

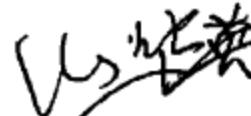
在信息技术中，硬件技术的进步往往容易引起人们的重视和兴趣，而软件技术往往不被人们关注。20 世纪微软公司等国际大型企业将软件技术的开发、生产和应用推向了高峰并使软件技术首次超越了硬件技术引起世人的关注。传统的软件开发、生产一直是依赖软件开发者的个体劳动，没有形成工业化的生产，从而制约了软件业的发展，也影响了信息技术的发展。尤其是软件的开发进度、质量和成本、风险的控制都是软件业的软肋。

软件项目管理技术的产生正是为了解决这一问题。近年来出现了许多有关软件项目管理技术的研究报告和专业书籍，它们都从不同的领域和角度对软件项目管理进行了介绍。我国的软件产业起步比较晚，相比世界先进国家有较大差距，甚至和印度相比也有不小的差距。在软件开发的水平、软件产品的大规模高速高质量生产、软件产品的出口等方面，我国都是后进者。究其原因在于我国缺乏大批高质量的专业软件人才，尤其是缺乏高水平的软件业管理人才和掌握专业技能的软件产业工人。专业的软件产业人才的教育培养是中国软件业乃至信息业的当务之急。两年前，在国家教育部的领导下，一些大学成立了软件学院，开展软件专业人才的培养。但是，大学的教材几乎都是国外教材的翻译或改编，从国外引进教材无疑是一条捷径，可以吸收别人的成果避免走弯路，站在巨人的肩膀上前进。但是，国外的教材总是缺乏结合国内实际的内容，这是困扰我国软件专业人才培养的难题。

北京邮电大学软件学院是教育部批准成立的第一批软件学院，经过几年的研究和实践，积累了许多经验。将这些经验和成果积集成书，教授给学生，无疑对我国软件业的发展具有重要意义。本书由北京邮电大学软件学院院长宋茂强教授主审，韩万江、姜立新编著，描述了软件项目管理的全过程，以一个完整的案例贯穿全书，将软件项目管理的理论、方法、技巧综合在一起。读者在学习本书时有如身历其境，通过对案例的分析、实验、讨论，能很好掌握软件项目管理的各种知识和技巧。本书是目前比较优秀的一本软件项目管理教材。全书深入浅出、图文并茂，每章后面都附有习题，便于读者掌握本章的内容，同时，本书的结构和内容适应面宽，可供不同层次的读者从中选择自己所需的知识。

期望本书的出版能对我国软件专业人才的培养发挥其应有的作用！

北京邮电大学经济管理学院  
教授/博导 舒华英



2004 年 10 月 29 日

## 第2版前言

本书第1版于2005年由机械工业出版社出版，得到了广大高校师生和读者的好评。第2版列选为普通高等教育“十一五”国家级规划教材，同时也是2007年度“教育部-IBM精品课程”的指定教材。在得到肯定和鼓励的同时，作者也收到了很多有益的建议，在此表示衷心的感谢。作者在这些年的软件项目管理教学和项目实践中也感觉到需要将项目管理教学和实践中的新技术、新发展、新需求加入到教材中，因此对第1版进行重新修订。

本书第2版在第1版的基础上，根据新技术、新信息的发展，总结了软件开发实践过程和教学过程的经验教训，完善了第1版精华部分，删除不适宜部分，同时，增加了新的知识元素。第2版在补充和完善第1版案例的基础上，增加了课程实验和实践过程，结合项目管理软件工具展示项目管理实例。本书是一本系统、有针对性、有实效性的书籍，对于从事软件项目工作的人员以及所有项目管理者都会起到非常好的借鉴作用。

参与本书编写的有韩万江、姜立新、郭士榕、刘砚颖、孙泉、岳鹏、韩冰等，刘砚颖提供了本书第16章的案例素材，在此一并表示感谢！

当然，由于作者水平有限，书中难免有疏漏之处，诚请各位读者批评指正，并希望读者能将实际工作中运用本书的经验和体会告诉作者，以便作者在下版中加以改进和完善。作者的Email是：casey\_han@263.net。

韩万江

2009年1月于北京

## 目录

序	0.6.2 项目计划	14
第2版前言	0.6.3 项目执行控制	14
第1版前言	0.6.4 项目结束	15
第0章 软件项目管理概述	0.7 本书的组织结构	15
0.1 项目与软件项目	0.8 小结	15
0.1.1 项目及其特征	0.9 练习题	16
0.1.2 项目群与项目、项目与子项目的关系		
0.1.3 软件项目		
0.1.4 软件项目要素组成		
0.1.5 项目目标成功实现的制约因素		
0.2 项目管理		
0.2.1 项目管理背景		
0.2.2 项目管理定义		
0.2.3 软件项目管理的特征及重要性		
0.2.4 软件项目管理与软件工程的关系		
0.3 项目管理的范围		
0.3.1 项目管理的五要素		
0.3.2 从战略上看项目管理的三个关注点		
0.3.3 从战术上看项目管理的三个关注点		
0.4 项目管理知识体系		
0.4.1 项目管理9大知识领域		
0.4.2 项目管理知识体系的标准化过程组		
0.5 过程管理与软件项目管理的关系		
0.5.1 软件过程定义		
0.5.2 过程管理在软件项目中的作用		
0.5.3 过程管理与项目管理的关系		
0.6 软件项目管理过程		
0.6.1 项目初始		
0.6.2 项目计划	14	
0.6.3 项目执行控制	14	
0.6.4 项目结束	15	
0.7 本书的组织结构	15	
0.8 小结	15	
0.9 练习题	16	
<b>第一篇 项目初始</b>		
第1章 软件项目初始过程	18	
1.1 项目立项	18	
1.2 合同项目	19	
1.2.1 甲方初始过程	19	
1.2.2 乙方初始过程	22	
1.3 内部项目	24	
1.4 项目授权	25	
1.4.1 项目章程	25	
1.4.2 项目经理的责任和权利	26	
1.4.3 项目经理的能力	27	
1.5 初始项目范围	27	
1.6 生存期模型	27	
1.6.1 瀑布模型	28	
1.6.2 V模型	29	
1.6.3 原型模型	29	
1.6.4 增量式模型	30	
1.6.5 螺旋式模型	31	
1.6.6 渐近式阶段模型	32	
1.7 校务通系统案例分析	33	
1.7.1 甲方招标需求	33	
1.7.2 乙方建议书	37	
1.7.3 项目合同	37	
1.7.4 乙方项目授权书	39	
1.7.5 生存期模型	39	
1.8 小结	41	
1.9 练习题	41	

## 第二篇 项目计划

<b>第2章 范围计划 .....</b>	<b>44</b>	<b>3.6 项目历时估计 .....</b>	<b>70</b>
<b>2.1 关于软件需求 .....</b>	<b>44</b>	<b>3.6.1 定额估算法 .....</b>	<b>71</b>
<b>2.2 需求管理过程 .....</b>	<b>45</b>	<b>3.6.2 经验导出模型 .....</b>	<b>71</b>
<b>2.2.1 需求获取 .....</b>	<b>46</b>	<b>3.6.3 工程评估评审技术 .....</b>	<b>71</b>
<b>2.2.2 需求分析 .....</b>	<b>47</b>	<b>3.6.4 专家估计方法 .....</b>	<b>72</b>
<b>2.2.3 需求规格编写 .....</b>	<b>48</b>	<b>3.6.5 类推估计方法 .....</b>	<b>73</b>
<b>2.2.4 需求验证 .....</b>	<b>51</b>	<b>3.6.6 模拟估计方法 .....</b>	<b>73</b>
<b>2.2.5 需求变更 .....</b>	<b>52</b>	<b>3.6.7 基于承诺的进度估计方法 .....</b>	<b>73</b>
<b>2.3 编写需求规格的方法 .....</b>	<b>53</b>	<b>3.6.8 Jones 的一阶估计准则 .....</b>	<b>73</b>
<b>2.4 任务分解定义 .....</b>	<b>53</b>	<b>3.6.9 管理预留 .....</b>	<b>73</b>
<b>2.4.1 WBS .....</b>	<b>54</b>	<b>3.7 进度计划编制 .....</b>	<b>74</b>
<b>2.4.2 任务分解的类型 .....</b>	<b>55</b>	<b>3.7.1 关键路径法 .....</b>	<b>74</b>
<b>2.4.3 任务分解的过程 .....</b>	<b>56</b>	<b>3.7.2 时间压缩法 .....</b>	<b>78</b>
<b>2.5 任务分解的方法 .....</b>	<b>57</b>	<b>3.7.3 资源平衡 .....</b>	<b>80</b>
<b>2.5.1 模板参照 .....</b>	<b>57</b>	<b>3.7.4 关键链法 .....</b>	<b>81</b>
<b>2.5.2 类比方法 .....</b>	<b>58</b>	<b>3.7.5 项目进度计划的优化 .....</b>	<b>84</b>
<b>2.5.3 自顶向下方法 .....</b>	<b>58</b>	<b>3.8 校务通系统案例分析 .....</b>	<b>84</b>
<b>2.5.4 自底向上方法 .....</b>	<b>59</b>	<b>3.9 小结 .....</b>	<b>87</b>
<b>2.6 任务分解结果的检验 .....</b>	<b>59</b>	<b>3.10 练习题 .....</b>	<b>87</b>
<b>2.7 校务通系统案例分析 .....</b>	<b>61</b>	<b>第4章 软件项目成本计划 .....</b>	<b>89</b>
<b>2.7.1 系统原型分析 .....</b>	<b>61</b>	<b>4.1 关于成本估算 .....</b>	<b>89</b>
<b>2.7.2 需求规格说明书 .....</b>	<b>62</b>	<b>4.1.1 项目规模与成本的关系 .....</b>	<b>89</b>
<b>2.7.3 系统WBS .....</b>	<b>63</b>	<b>4.1.2 成本估算的定义 .....</b>	<b>90</b>
<b>2.8 小结 .....</b>	<b>63</b>	<b>4.2 成本估算的过程 .....</b>	<b>90</b>
<b>2.9 练习题 .....</b>	<b>64</b>	<b>4.2.1 估算输入 .....</b>	<b>90</b>
<b>第3章 软件项目进度计划 .....</b>	<b>65</b>	<b>4.2.2 估算处理 .....</b>	<b>91</b>
<b>3.1 关于进度估算 .....</b>	<b>65</b>	<b>4.2.3 估算输出 .....</b>	<b>91</b>
<b>3.2 任务定义 .....</b>	<b>66</b>	<b>4.3 成本估算的方法 .....</b>	<b>91</b>
<b>3.3 任务关联关系 .....</b>	<b>66</b>	<b>4.3.1 代码行、功能点及对象点 .....</b>	<b>91</b>
<b>3.3.1 任务之间的关系 .....</b>	<b>66</b>	<b>4.3.2 类比估算法 .....</b>	<b>96</b>
<b>3.3.2 任务关联关系的依据 .....</b>	<b>66</b>	<b>4.3.3 自下而上估算法 .....</b>	<b>97</b>
<b>3.4 进度管理图示 .....</b>	<b>67</b>	<b>4.3.4 参数模型估算法 .....</b>	<b>98</b>
<b>3.4.1 甘特图 .....</b>	<b>67</b>	<b>4.3.5 专家估算法 .....</b>	<b>103</b>
<b>3.4.2 网络图 .....</b>	<b>68</b>	<b>4.3.6 猜测法 .....</b>	<b>104</b>
<b>3.4.3 里程碑图 .....</b>	<b>69</b>	<b>4.3.7 估算方法综述 .....</b>	<b>104</b>
<b>3.4.4 资源图 .....</b>	<b>69</b>	<b>4.4 估算的误差度 .....</b>	<b>105</b>
<b>3.5 任务资源估计 .....</b>	<b>69</b>	<b>4.5 成本预算 .....</b>	<b>107</b>
		<b>4.6 校务通系统案例分析 .....</b>	<b>108</b>
		<b>4.6.1 成本估算 .....</b>	<b>108</b>

4.6.2 成本预算 .....	109	7.4.2 沟通建议 .....	139
4.7 小结 .....	110	7.5 校务通系统案例分析 .....	139
4.8 练习题 .....	110	7.6 小结 .....	139
第5章 软件项目质量计划 .....	112	7.7 练习题 .....	140
5.1 质量概述 .....	112	第8章 软件项目风险计划 .....	141
5.1.1 软件质量 .....	113	8.1 风险概念 .....	141
5.1.2 软件质量与硬件质量的比较 .....	114	8.1.1 风险定义 .....	141
5.1.3 软件质量的重要性 .....	114	8.1.2 风险类型 .....	142
5.1.4 软件质量管理的发展过程 .....	114	8.1.3 风险的基本性质 .....	143
5.2 质量管理 .....	115	8.2 风险管理 .....	144
5.2.1 质量计划 .....	115	8.3 风险识别 .....	145
5.2.2 质量保证 .....	116	8.3.1 德尔菲方法 .....	145
5.2.3 质量控制 .....	116	8.3.2 头脑风暴法 .....	145
5.3 质量计划开发 .....	117	8.3.3 情景分析法 .....	146
5.3.1 质量成本 .....	117	8.3.4 风险条目检查表 .....	146
5.3.2 质量计划的方法 .....	118	8.3.5 其他方法 .....	152
5.3.3 质量计划的编写 .....	118	8.3.6 风险识别的结果 .....	152
5.4 质量体系 .....	120	8.4 风险评估 .....	152
5.5 软件质量改善的建议 .....	122	8.4.1 定性风险评估 .....	152
5.6 校务通系统案例分析 .....	122	8.4.2 定量风险分析 .....	153
5.7 小结 .....	125	8.4.3 风险评估的结果 .....	155
5.8 练习题 .....	125	8.5 风险应对计划 .....	156
第6章 软件项目人力资源计划 .....	127	8.5.1 回避风险 .....	156
6.1 相关概念 .....	127	8.5.2 转移风险 .....	157
6.2 项目组织结构 .....	127	8.5.3 损失控制 .....	157
6.2.1 职能型组织结构 .....	128	8.5.4 自留风险 .....	158
6.2.2 项目型组织结构 .....	128	8.5.5 风险规划的结果 .....	158
6.2.3 矩阵型组织结构 .....	129	8.6 风险管理建议 .....	159
6.3 责任分配矩阵 .....	131	8.7 校务通系统案例分析 .....	161
6.4 人员管理计划 .....	131	8.8 小结 .....	161
6.5 校务通系统案例分析 .....	132	8.9 练习题 .....	161
6.6 小结 .....	133	第9章 软件项目合同计划 .....	163
6.7 练习题 .....	133	9.1 项目采购 .....	163
第7章 项目沟通计划 .....	135	9.2 合同规划 .....	163
7.1 项目沟通管理概述 .....	135	9.2.1 关于合同 .....	164
7.2 沟通方式 .....	136	9.2.2 合同类型 .....	164
7.3 沟通渠道 .....	136	9.2.3 合同计划 .....	165
7.4 项目沟通计划 .....	137	9.3 软件外包 .....	166
7.4.1 沟通计划 .....	137	9.4 校务通系统案例分析 .....	167

## ■ 软件项目管理概述

### 0.1 项目与软件项目

无论是“项目”、“软件”还是“软件项目”已经越来越被大家所熟悉，而且普遍存在于我们生活或者社会的各个方面。而软件行业是一个极具挑战性和创造性的新行业，管理上没有很成熟的经验可供借鉴。

#### 0.1.1 项目及其特征

人类社会和日常生活中有很多的活动，但是有的活动我们称为“项目”，有的不能称为项目。所谓项目（project），就是为了创造一个唯一的产品或提供一个唯一的服务而进行的临时性的努力；是以一套独特而相互联系的任务为前提，有效地利用资源，为实现一个特定的目标所做的努力；项目是一个特殊的将被完成的有限任务，它是在一定时间内满足一系列特定目标的多项相关工作的总称。

从野餐活动到大型聚会，从阿波罗登月到微软的操作系统，从修建三峡到神州发射都是项目，一般说，工作活动包括日常运作和项目，它们虽然有共同点，例如它们都需要由人来完成、都受到有限资源的限制、都需要计划、执行、控制。但是项目是组织层次上进行的具有时限性和唯一性的工作，也许需要一个人，也许涉及成千上万的人，也许需要 100 小时完成、也许要 10 年 1 千人完成等等。“上班”、“批量生产”、“每天的卫生保洁”等是属于日常运作，不是项目。项目与日常运作的不同是：项目是一次性的，日常运作是重复进行的；项目是以目标为导向的，日常运作是通过效率和有效性体现的；项目是通过项目经理及其团队工作完成的，而日常运作是职能式的线形管理；项目存在大量的变更管理，而日常运作则基本保持持续的连贯性。下面给出项目所具有的特征。

1) 目标性。项目工作的目的在于得到特定的结果，即项目是面向目标的。其结果可能是一种产品，也可能是一种服务。目标贯穿于项目始终，一系列的项目计划和实施活动都是围绕这些目标进行的。例如，一个软件项目的最终目标可以是一个学生成绩管理系统，另外一个软件系统的目标可能是一个证券交易系统等等。

2) 相关性。项目的复杂性是固有的，一个项目里有很多彼此相关的活动，例如，某些活动在其他活动完成之前不能启动，而另一些活动则必须并行实施，如果这些活动相互之间不能协调地开展，就不能达到整个项目的目标。

3) 周期性。项目要在一个限定的期间内完成，是一种临时性的任务，有明确的开始点和结束点。当项目的目标达到时，意味着项目任务完成。项目管理的很大一部分精力是用来保证在预定时间内完成项目任务，为此而制定项目计划进度表，标识任务何时开始，何时结束。项目任务不同于批量生产。批量生产是相同的产品连续生产，取决于要求的生产量，当生产任务完成时，生产线才停止运行。这种连续生产不是项目。

4) 独特性。在一定程度上，项目与项目之间没有重复性，每个项目都有其独自的特点。每一个项目都是唯一的。如果一位工程师正在按照规范建造第五十栋农场式的住宅，其独特性一定很低，它的基本部分与已经造好的第四十九栋是相同的，如果说有特殊性也只是在于地基的土壤不同，使用了一个新的热水器，请了几位新木工等等。然而，如果你为新一代计算机设计操作系统时，你的工作必然

会有很强的独特性。这个项目以前没有做过，以前的经验能提供的帮助不多，而且会存在很大的风险和很多不确定性因素。

5) 约束性。每一项目都需要运用各种资源来作为实施的保证，而资源是有限的。所以，资源成本是项目成功实施的一个约束条件。

6) 不确定性。一个项目开始前，应当在一定的假定和预算基础上准备一份计划，但是，在项目的具体实施中，外部和内部因素总是会发生一些变化，因此项目也会出现不确定性。

### 0.1.2 项目群与项目、项目与子项目的关系

现在来确认一下关于项目群与项目、项目与子项目的关系。

项目群也称为大型项目（program），是通过协调来进行统一管理的一组相互联系的项目，它本身可能不是项目。许多大型项目通常包括持续运作的活动。一个大型项目可以理解为比项目高一级别的大项目，比如“863”计划、“星火计划”、“登月计划”、“阿波罗登月计划”等。以“863”计划为例，它的目标是跟踪世界先进水平，集中资源重点投入，争取在我国部分有优势的高科技领域有所突破，为我国在21世纪的经济发展和国防安全创造条件。这样的目标是战略性的，很难具体化，所以本身不是项目，但它可以通过一系列的具体项目去实施。

而子项目（subproject）是将项目分解成为更小单位，以便更好控制项目，项目中的某一阶段可以是一个单独的项目，也可以是一个子项目，一个子项目可以转包给外部机构的一个单元。在实际工作中，子项目的划分是很灵活的，可能有，也可能没有，视项目的需要而定。可以按照阶段划分，例如一期项目，二期项目；也可以按照项目的组成部分来划分。

### 0.1.3 软件项目

软件是计算机系统中与硬件相互依存的部分，它是包括程序、数据及其相关文档的完整集合。其中，程序是按事先设计的功能和性能要求执行的指令序列；数据是使程序能正常操纵信息的数据结构；文档是与程序开发、维护和使用有关的图文材料。

软件项目除了具备项目的基本特征之外，还有如下的特点：

1) 软件是一种逻辑实体，不是具体的物理实体，它具有抽象性。这使得软件与其他的诸如硬件或者工程类有很多的不同。

2) 软件的生产与硬件不同，开发过程中没有明显的制造过程，也不存在重复生产过程。

3) 软件没有硬件的机械磨损和老化问题，然而，软件存在退化问题，在软件的生存期中，软件环境的变化将导致软件失效率的提高。

4) 软件的开发受到计算机系统的限制，对计算机系统有不同程度的依赖。

5) 软件开发至今没有摆脱手工的开发模式，软件产品基本上是“定制的”，做不到利用现有的软件组件组装成所需要的软件。

6) 软件本身是复杂的。来自于应用领域实际问题的复杂性和应用软件技术的复杂性。

7) 软件的成本相当的昂贵。软件开发需要投入大量的、复杂的、高强度的脑力劳动，因此成本比较的高。

8) 很多的软件工作涉及社会的因素，比如许多的软件开发要受到机构、体系和管理方式等问题的限制。

软件项目是一种特殊的项目，它创造的唯一产品或者服务是逻辑载体，没有具体的形状和尺寸，只有逻辑的规模和运行的效果。软件项目不同于其他的项目，软件是一个新领域而且涉及的因素比较多，管理比较复杂。目前，软件项目的开发远远没有其他领域的项目规范，很多的理论还不能适应所有的软件项目，经验在软件项目中仍起很大的作用。软件项目是有相互作用的各个系统组成的，“系统”包括彼此相互作用的部分，软件项目中涉及的因素越多，彼此之间相互的作用就越大。另外变更

在软件项目中也是常见的现象，例如需求的变更、设计的变更、技术的变更、社会环境的变更等，所有这些都说明软件项目管理的复杂性。

项目的独特性和临时性决定了项目是渐进明细的，软件项目更是如此，因为软件项目比其他的项目有更大的独特性。“渐进明细”表明项目的定义会随着项目团队成员对项目、产品等理解认识的逐步加深而得到逐渐深入的描述。

软件行业是一个极具挑战性和创造性的行业，软件开发是一项复杂的系统工程，牵涉到各方面的因素。在实际工作中，经常会出现各种各样的问题，甚至面临失败。如何总结、分析失败的原因，得出有益的教训，是今后的项目中取得成功的关键。

#### 0.1.4 软件项目要素组成

简单地说，项目就是在既定的资源和要求的约束下，为实现某种目的而相互联系的一次性工作任务。一个软件项目的要素包括软件开发的过程、软件开发的结果、软件开发赖以生存的资源以及软件项目的特定委托人或者说是客户，它既是项目结果的需求者，也是项目实施的资金提供者。

#### 0.1.5 项目目标成功实现的制约因素

一个成功的项目应该是在工程允许的范围内满足成本、进度和客户满意的产品质量。所以，项目目标的成功实现受4个因素制约：项目范围、成本、进度计划和客户满意度，见图0-1。项目范围是为使客户满意，必须做的所有工作。项目成本就是完成项目所需要的费用。项目进度是安排每项任务的起止时间以及所需的资源等，是为项目描绘的一个过程蓝图。项目目标就是在一定时间、预算内完成工作范围，以使客户满意。客户能否满意要看交付的成果质量，只有客户满意才能意味着可以更快的结束项目，否则会导致项目的拖延，从而增加额外的费用。

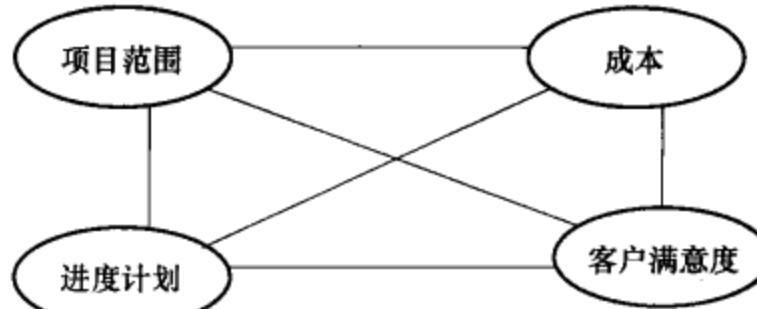


图0-1 项目目标成功实现的制约因素

### 0.2 项目管理

项目普遍存在于人们的工作和生活中，如何管理这些项目就是一项需要研究的任务。项目管理起源于美国，20世纪40~50年代主要应用于国防和军工项目，后来被广泛应用于工商、金融、信息等产业以及行政管理领域。目前，项目管理已经成为综合多门学科的新兴研究领域，其理论来自于项目管理的工作实践。所谓项目管理就是指把各种系统、方法和人员结合在一起，在规定的时间、预算和质量目标范围内完成项目的各项工作。

对一个组织的管理而言，主要包括三个部分“战略管理”、“运作管理”、“项目管理”，见图0-2。



图0-2 三种管理的关系

- 战略管理（strategy management）是从宏观上帮助企业明确和把握企业的发展方向。
- 运作管理（operation management）是对日常的、重复性工作的管理。
- 项目管理（project management）是对一次性的、创新性工作的管理。

项目是企业的最小赢利单位，项目管理自然成为构筑企业利润的基石。从这种意义上说，项目管理是企业的核心竞争力所在。由于项目管理具有效率高，反映灵敏的优点，所以更多的企业管理希望采取项目式管理的方式（management by project），从而可以对用户反应更及时，管理更高效，提高企业的管理质量。

### 0.2.1 项目管理背景

随着世界由工业时代进入信息时代，时空概念的根本改变加剧了项目的复杂性和可变性。项目涉及的范围和时间、空间跨度都在以空前的速度扩大。而随着行业竞争的加剧，项目只有在最少时间、最低花费的情况下完成才有意义。项目本身的复杂性和巨大风险以及在分工合作中个人经验的不确定性，使个人经验已无法确保项目的成功或按时完成。

项目管理是20世纪50年代后期发展起来的一种计划管理方法，它一出现就引起广泛关注。1957年美国杜邦公司把这种方法应用于设备维修，把维修停工时间由125小时锐减为78小时。1958年美国人在北极星导弹设计中，应用项目管理技术，竟把设计完成时间缩短了两年。由于项目管理在运作方式和管理思维模式上最大限度地利用了内外资源，从根本上改善了管理人员的工作程序，提高了效率，降低了风险，因此从20世纪60年代以来它被广泛运用到航空航天、国防、信息、建筑、能源、化工、制造、环保、交通运输、金融、营销、服务、法律等行业，以及国家和地方政府乃至联合国；它不仅适用于大公司也适用于小型企业。目前，在全球发达国家的政府部门和企业机构中，项目管理已成为运作的中心模式。

### 0.2.2 项目管理定义

项目管理是指一定的主体，为了实现其目标，利用各种有效的手段，对执行中的项目周期的各阶段工作进行计划、组织、协调、指挥、控制，以取得良好经济效益的各项活动的总和。通过项目各方干系人的合作，把各种资源应用于项目，以实现项目的目标，使项目干系人的需求得到不同程度的满足。要想满足或超过项目干系人的需求和期望，我们就需要在下面这些相互间有冲突的要求中寻求平衡：

- 1) 范围、时间、成本和质量。
- 2) 有不同需求和期望的项目干系人。
- 3) 明确表示出来的要求（需求）和未明确表达的要求（期望）。

“项目管理”有时被描述为对连续性操作进行管理的组织方法。这种方法，更准确地应该被称为“由项目实施的管理”，它是将连续性操作的许多方面作为项目来对待，以便对其可以采用项目管理的方法。虽然，对于一个通过项目实施管理的组织而言，对项目管理的认识显然是非常重要的。

项目管理是要求在项目活动中运用知识、技能、工具和技术，以便达到项目目标的活动。它是伴随着项目的进行而进行的，目的是为了确保项目能够达到期望的结果的一系列管理行为。正如Mulcahy所言“项目经理的工作是‘奇妙的’、‘伟大的’，但是也是非常有技巧的。”

项目管理类似导弹发射控制过程，需要一开始就设定好目标，然后在飞行中锁定目标，同时不断调整导弹的方向，使之不能偏离正常的轨道，最终击中目标。

软件项目管理是为了使软件项目能够按照预定的成本、进度、质量顺利完成，而对成本、人员、进度、质量、风险等进行分析和管理的活动。项目管理是项目能否高效、顺利进行的一项基础性的工作。

### 0.2.3 软件项目管理的特征及重要性

当前社会的特点是“变化”，而这种变化在信息产业中体现得尤为突出，技术创新速度越来越快，用户需求与市场不断变化，人员流动也大大加快。在这种环境下，企业需要应对的变化以及由此带来

的挑战大大增加，也给管理带来了很多问题和挑战。目前软件开发中面临很多的问题，例如：

- 1) 在有限的时间、资金内，要满足不断增长的软件产品质量要求。
- 2) 开发的环境日益复杂，代码共享日益困难，需跨越的平台增多。
- 3) 程序的规模越来越大。
- 4) 软件的重用性需要提高。
- 5) 软件的维护越来越困难等等。

因此，软件项目管理显得更为重要。软件项目管理的提出是在 20 世纪 70 年代中期的美国，当时美国国防部专门研究了软件开发不能按时提交、预算超支和质量达不到用户要求的原因，结果发现 70% 的项目是因为管理不善引起的，而非技术原因。于是软件开发者开始逐渐重视起软件开发中的各项管理。到了 20 世纪 90 年代中期，软件项目管理不善的问题仍然存在。据美国软件工程实施现状的调查，软件研发的情况仍然很难预测，大约只有 10% 的项目能够在预定的费用和进度下交付。

软件项目管理和其他项目管理相比有相当的特殊性。

1) 软件是纯知识产品，其开发进度和质量很难估计和度量，生产效率也难以预测和保证。与普通的项目不同，软件项目的交付成果事先“看不见”，并且难以度量。特别是很多应用软件项目已经不再是业务流程的“电子化”，而是同时涉及业务流程再造或业务创新。因此，客户在项目早期对到底要做什么样，确实很难说清楚，但这一点对于软件项目的成败恰恰又是至关重要的。与此矛盾的是，公司一般是市场营销人员负责谈判，其重点是迅速签约，而不是如何交付，甚者为了尽早签约而“过度承诺”。遇到模糊问题时也怕因为解释而节外生枝，所以避而不谈，而甲方为了保留回旋余地，也不愿意说得太清楚，更不愿意主动提出来（因为甲方还有最终验收的主动权）。等到项目经理一旦接手，所有这些没有说清楚的隐患和口头承诺都将暴露出来，并最终都由项目经理承担。

2) 项目周期长，复杂度高，变数多。IT 项目的交付周期一般都比较长，一些大型项目的周期可以达到 2 年以上。这样长的时间跨度内可能发生各种变化。软件系统的复杂性也导致了开发过程中各种风险的难以预见和控制。从外部来看，商业环境、政策法规变化会对项目范围、需求造成重大影响。例如，作者曾经从事的金融项目，临近上线时国家推出了“利息税”政策，造成整个系统的大幅变更。从内部来看，组织结构、人事变动等对项目的影响更加直接。有时，伴随着新的领导到任，其思路的变化，甚至对项目的重视程度的变化，都可能直接影响项目的成败。软件项目管理中有一个重要的生存法则：“不要相信任何人的口头承诺”。就是这个原因，即使是你绝对信赖的人，也可能发生人事变动，之后你无法保证继任的人员能够继续兑现承诺。

3) 软件需要满足一群人的期望。软件项目提供的实际上是一种服务，服务质量的好坏不仅仅是最终交付的质量，更重要的是客户的体验。实际上，项目中的“客户”不是一个人，而是一群人！他们可能来自多个部门，对项目的关注点不同，在项目中的利益（得与失）也不同。所以，当我们谈到满足“客户需求”的时候，实际的意思是“满足一群想法和利益各不相同的人的需求”。

有了项目管理，就有了管理改进的基础，无论刚开始的项目管理多么糟糕，只要有管理，就有了改进的可能性。所以，进行软件项目管理是必要的。像 Windows 这样的操作系统有 1500 万行以上的代码，同时有数千个程序员在进行开发，项目经理都有上百个。这样庞大的系统如果没有很好的管理，其软件质量是难以想象的。

#### 0.2.4 软件项目管理与软件工程的关系

软件工程可以分为三个部分，即软件工程可以包括三个重要的线索：一条线索是软件项目开发过程，一条线索是软件项目管理过程，另外一条线索是软件过程改进，如图 0-3 所示。

开发过程是软件人员生产软件的过程（例如需求分析\设计\编码\测试等），相当于机械流水线上的生产过程；管理过程是项目管



图 0-3 软件工程的三个线索

理者规划软件开发、控制软件开发的过程，相当于机械流水线上的管理过程；同理，过程改进相当于对软件开发过程和软件管理过程的“工艺流程”进行管理和改进，如果没有好的工艺，生产不出好的产品，它包括对开发过程和管理过程的定义和改进。

在现实软件项目中，大家更注意的线索是软件开发过程，经常忽略了软件管理过程和过程改进的线索，其实这两个线索很重要，甚至超过软件开发过程的重要性。随着软件的不断发展，软件规模的不断壮大，软件开发也会逐步向软件工厂化发展，软件项目开发过程就相当于软件工厂中生产车间的生产过程，而生产工艺的制定、生产内容、生产质量、生产时间、生产成本等工作都是项目管理的工作。软件开发过程的工作更多的是软件设计、编码等，项目管理的工作更多的是如何保证软件的成功，可能在有些人看来编码的人的工作更实在一些，项目管理的工作好像不实在。其实，这是一个误会，项目管理可以让一个项目获得高额的盈利，也可以让一个项目损失惨重，但是编码的人就不会有这个影响力。让软件工程成为真正的工程，就需要软件项目的开发、管理、过程等方面规范化、工程化、工艺化、机械化。

软件项目管理的根本目的是为了让软件项目尤其是大型项目的整个软件生命周期都能在管理者的控制之下，以预定成本按期、按质的完成软件并交付用户使用。而研究软件项目管理是为了从已有的成功或失败的案例中总结出能够指导今后开发的通用原则和方法，以避免前人的失误。

## 0.3 项目管理的范围

### 0.3.1 项目管理的五要素

项目管理的五要素有技术（technical）、方法（methodology）、团队建设（team building）、信息（information）、沟通（communication）。项目管理是技术，也是方法，是技巧，也是信息，当然也需要团队建设。其中沟通非常重要，项目经理主要的工作是沟通。沟通包括技术的沟通、管理的沟通、质量的沟通等很多方面。

### 0.3.2 从战略上看项目管理的三个关注点

从战略上看，有效的项目管理集中于3个P上：人员（people）、问题（problem）和过程（process）。

#### 1. 人员

事实上，“人的因素”非常重要，是项目最为宝贵的财富，软件行业就更是这样。以致于软件工程研究所专门开发了一个人员管理能力成熟度模型（PM-CMM），它为软件人员定义了招聘、人员选择、业绩管理、培训、报酬、专业发展、组织和工作计划以及团队精神、企业文化培养等关键实践域。

使用人必须先信任人、培养人，给他一个发展的空间，让他看到一个希望，让他感到工作的过程是一个自身价值升值的过程。需要建立一个祥和、友善、互助、向上的企业文化氛围。相互间的技术保密、妒忌是软件公司的大忌。首先，技术管理人员要有一个平常心，不要太多地被名利所左右。公司先解决好技术管理层的问题，公司如何制定提升和奖励政策，也会在这方面起一个引导作用。工作的安排要合适，任务要明确，多协作，少冲突，避免在同一领域展开员工间不必要的竞争，需要多人完成一项工作时要新老结合，高低结合。人的提升有多种途径，在用人的过程中，根据每个人不同的喜好和性格设计不同的发展路径，必要的时候可以和当事人交流，将公司对他的期望和他本人的努力统一起来。人的“提升”感重在于自身价值在集体中被承认，被认同，被同仁接受。

#### 2. 问题

项目经理的一个重要任务是发现问题和解决问题。明确该项目的目的和范围，选择合适的解决方案，定义技术和管理的约束，进行成本估算，有效的风险评估，适当地划分项目任务或给出意义明确

的项目进度等都是需要解决的问题。在项目的实施过程中又会出现很多的新问题，需要及时的发现和解决。

### 3. 过程

单纯注重项目管理技术本身，是无法对项目管理能力有实际提高的，因此在这里我们要引申出过程管理，过程管理也是项目管理的任务，下节将进行详细说明。

#### 0.3.3 从战术上看项目管理的三个关注点

软件项目管理的四大变量为：范围、质量、成本以及交期。项目管理需要在相互间具有冲突的要求中寻求平衡：1) 范围、成本、质量和交期；2) 具有不同需求和期望的项目相关人员；3) 明确表示出来的要求（需求）和未明确表达的要求（期望）。因此，从战术上看，项目管理主要关注在项目的范围（满足质量要求的产品需求）、成本、进度这三方面上，见图 0-4。

图 0-4 这个三角形的三个边是相互影响，任何一边发生变化都会影响其他两边的，例如，如果产品规格发生变化，那么产品的成本就要重新估算，项目的进度也要重新安排。如果要赶进度，就要对成本或者规格做出折中等。项目管理的作用是在项目目标之间做出一些权衡，在某一领域绩效的提高可能是以降低其他领域的绩效为代价的。具体的绩效平衡会因项目和组织的不同而不同。成功的项目管理需要积极的管理这些相互作用的目标。

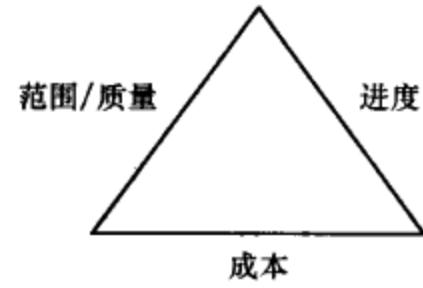


图 0-4 项目管理的三角形

## 0.4 项目管理知识体系

从前，有人认为项目管理是一种“意外的职业”。因为常常是人们在项目中先承担了项目责任，可能是从技术开发开始，然后随着项目经验的逐步提高，积累一定的技术管理等经验，最后顺理成章地当上了项目经理。但是管理一个项目的有关知识不是通过系统学习得来的，而是在实践中摸索出来的。摸索的过程可能会导致严重损失。近年来，在减小项目管理意外性方面已经有了很大进步。很多企业的决策者们日益认识到项目管理方法可以帮助他们在复杂的竞争环境中取得成功。项目管理成为热点，一位著名的管理学家明确指出：“项目管理是在当今急剧变化的时代中求得生存的关键”。为了减少项目管理的意外性，许多机构或者企业开始要求雇员系统地学习项目管理技术，努力成为经认证合格的项目管理人员。

PMP (Project Management Professional) 是项目管理专业人员资格的缩写，它是美国项目管理学会 (Project Management Institute, PMI) 开发并负责组织实施的一种专业资格认证。成为 PMP 是一个挑战，认证本身可以为个人的事业发展带来很多的好处。该项认证已经获得世界上 100 多个国家的承认，可以说是目前全球认可程度很高的项目管理专业认证，也是项目管理资格最重要的标志之一，在国际上已经树立了权威。在世界很多国家，特别是西方发达国家，PMP 已经被认为是合格项目管理的标志之一。

项目管理知识体系 (Project Management Body Of Knowledge, PMBOK) 是 PMI 组织开发的一套关于项目管理的知识体系。它是 PMP 考试的关键材料，它为所有的项目管理提供了一个知识框架。项目管理知识体系 (PMBOK 2004) 包括项目管理的九个知识领域、5 个标准化过程组及 44 个模块。九个知识领域分别是：项目集成管理 (Project Integration Management)，项目范围管理 (Project Scope Management)，项目时间管理 (Project Time Management)，项目成本管理 (Project Cost Management)，项目人力资源管理 (Project Human Resource Management)，项目沟通管理 (Project Communication Management)，项目风险管理 (Project Risk Management)，项目质量管理 (Project Quality Management)，项目采购管理 (Project Procurement Management)。其中 9 个领域包括的管理要素如图 0-5 所示。



图 0-5 项目管理知识领域

#### 0.4.1 项目管理 9 大知识领域

项目管理 9 大知识领域分布在项目进展过程中的各个阶段，它们的关系可以这样描述：

- 为了成功实现项目的目标，首先必须设定项目的工作和管理范围，即项目范围管理（what to do）。
- 为了正确实施项目，需要对目标进行分解，即对项目的时间、质量、成本三大目标分解，即项目时间管理（when）、项目成本管理（how much）、项目质量管理（how good）。
- 项目实施过程中，需要投入足够的人力、物力资源，即项目人力资源管理（People & Motivation）、项目采购管理（Partners）。
- 为了对项目团队中人员的管理，让大家目标一致的完成项目，需要沟通，即项目沟通管理（Understand & Be Understood）。
- 当然，项目在实施过程中会遇到各种风险，所以要进行风险管理，即项目风险管理。
- 最后项目管理一定要协调各个方面，不能只顾局部的利益和细节，所以需要集成管理，即项目集成管理。

项目管理 9 大知识领域具体描述如下：

##### 1) 项目集成管理

项目集成管理是项目成功的关键，它贯穿了项目的全过程。项目集成管理是在项目的整个生存

期内，协调项目管理其他各管理知识域，将项目管理的方方面面集成为一个有机整体，保证项目总目标的实现。项目集成管理从一个宏观的尺度将项目作为一个整体来考察。包括的过程如下：项目章程编制（develop the project charter）、初始项目范围编制（develop the preliminary project scope statement）、项目计划编制（develop the project management plan）、指导与管理项目执行（direct and manage project execution）、项目监控（monitor and control the project work）、集成变更控制（perform integrated change control）、项目结束（close the project）等过程。

项目集成管理的目标在于对项目中的不同组成元素进行正确高效的协调，它并不是所有项目组成元素的简单相加。

## 2) 项目范围管理

项目范围是为了交付具有特定属性和功能的产品而必须完成的工作。项目范围管理“是以保证项目包含所有需要完成的工作，以顺利完成项目所需要的所有过程”。范围管理主要是定义项目需要完成的工作，确保项目包含且只包含所有需要完成的工作。范围管理定义可以控制项目包含什么内容和不包含什么内容。包括的过程如下：范围规划（scope planning）、范围定义（scope definition）、任务分解结构定义（creating the WBS）、范围核实（scope verification）、范围控制（scope control）等过程。

## 3) 项目时间管理

很多的项目经理将按时提交项目作为一个最大的挑战之一，时间是灵活性最小的控制元素，进度是导致项目冲突的最主要的原因，尤其在项目的后期。所以项目管理者学习进度管理过程尤为重要。项目进度管理就是保证项目按时完成需要的一些管理过程，包括任务定义（activity definition）、任务排序（activity sequencing）、任务资源估计（activity resource estimating）、任务历时估计（activity duration estimating）、进度计划编制（schedule development）、进度计划控制（schedule control）等过程。

## 4) 项目成本管理

项目成本管理是在项目具体实施过程中，为了确保完成项目所花费的实际成本不超过预算成本而展开的管理活动。包括成本估算（cost estimating）、成本预算（cost budgeting）、成本控制（cost control）等过程。成本估算时估计完成项目需要的资源成本。成本预算（cost budgeting）是将总的估算成本分摊到各项任务中，以便建立项目跟踪的成本基线。成本控制（cost control）是控制成本预算的变更。

## 5) 项目质量管理

项目质量管理要求保证该项目能够兑现它关于满足各种需求的承诺。它涵盖了与决定质量工作的策略、目标和责任的全部管理功能有关的各种活动。项目质量管理过程包括如下3个过程：质量规划（quality planning）、质量保证（quality assurance）、质量控制（quality control）等过程。

## 6) 项目人力资源管理

项目人力资源管理包括了最有效管理人力资源的过程，它要求充分发挥参与项目的人员的作用，包括人力资源规划（human resource planning）、人力资源获取（acquiring the project team）、团队建设（developing the project team）、团队管理（managing the project team）等过程。这里的人力包括项目中所有的干系人：赞助商、供应商、客户、项目团队成员、支持人员等等。人力资源规划是定义项目的角色、职责、汇报关系等。人力资源获取是招募项目需要的人员并分配到相应的工作中。团队建设是开发个人和团队的技能和增强项目性能的过程。团队管理是跟踪团队成员性能、激励团队成员热情、及时反馈和解决问题、从而增强项目的性能的过程。

## 7) 项目沟通管理

项目沟通管理包括为了确保项目信息及时准确的生成、收集、发布、存储和部署的过程。主要有4个过程，分别是沟通规划（communications planning）、信息分发（information distribution）、绩效报告（performance reporting）、项目干系人管理（managing stakeholder）。沟通规划是确定项目人员的沟通需

求和需要的信息，即确定谁需要什么信息，什么时候需要，如何获取这些信息。信息分发及时的提供项目人员需要的信息。项目绩效报告包括收集和发布项目的性能信息，如项目的状态，项目的进展报告，项目的预测等。项目干系人管理主要是通过沟通管理满足项目相关人员的需求和期望，同时解决问题。

### 8) 项目风险管理

项目风险管理是决定采用什么方法和如何规划项目风险的活动，是指对项目风险从识别到分析乃至采取应对措施等一系列过程。它包括将积极因素所产生的影响最大化和使消极因素产生的影响最小化两方面内容。风险管理的主要过程是：风险管理规划（risk management planning）、风险识别（risk identification）、定性风险分析（qualitative risk analysis）、定量风险分析（quantitative risk analysis）、风险应对计划（risk response planning）和风险监控（risk monitoring and control）。

### 9) 项目采购管理

项目采购管理包括从执行组织之外获取货物和服务的过程。为了满足项目的需求，项目组织需要从外部获取某些产品，这就是采购。采购的意义是广义的，可能是采购物品，也可能是采购服务（例如软件开发等），还包括收集有关产品的信息，进行择优选购。采购一般是通过合同进行的。项目采购管理包括如下的过程：采购计划编制（planning purchases and acquisitions）、合同计划编制（planning contracting）、供方反馈获取（requesting seller responses）、供方选择（source selection）、合同管理（contract administration）、合同收尾（closing the contract）。

## 0.4.2 项目管理知识体系的标准化过程组

按照项目管理生命周期，项目管理知识体系又分为5个标准化过程组，也称为项目管理生命周期的5个阶段，见图0-6，他们是启动过程组、计划过程组、执行过程组、控制过程组、收尾过程组。每个标准化过程组有一个或多个过程组成。它们正如导弹的发射过程，从设定目标，锁定目标，执行目标，控制目标，到最后的达到目标的目的。它们的关系定义如下：

1) 启动过程组：启动过程组主要是确定一个项目或一个阶段可以开始了，并要求着手实行。定义和授权项目或者项目的某个阶段。在这一过程中最重要的是确定项目章程和项目初步范围说明书。项目章程是在客户与项目经理达成共识后建立的，主要包括项目开发人、粗的成本估算和进度里程碑等信息。项目初步范围说明书包含了范围说明书涉及到的所有内容，同时还包含了初步的WBS分解、假设约束、风险、开发人员、范围、交付物、粗进度里程碑、粗成本费用估算、验收准则和项目边界等诸多内容。这些做好后防止客户在软件提交后提出无理的要求。

2) 计划过程组：计划过程组是为完成项目所要达到的商业要求而进行的实际可行的工作计划的设计、维护，确保实现项目的既定商业目标。计划基准是后面跟踪和监控的基础。

3) 执行过程组：执行过程组根据前面制定的基准计划，协调人力和其他资源，去执行项目管理计划或相关的子计划。执行过程则存在两个方面的输入，一个是根据原来的基准来执行，另外一个就是要根据监控中发现的变更来执行。因为主要变更必须要得到了整体变更控制批准后才能够执行。

4) 控制过程组：通过监督和检测过程确保项目达到目标，必要时采取一些修正措施。集成变更控制是一个重要的过程。

5) 收尾过程组：收尾过程组是取得项目或阶段的正式认可并且有序地结束该项目或阶段。应提交给客户，发布相关的结束报告，并且更新组织过程资产并释放资源。

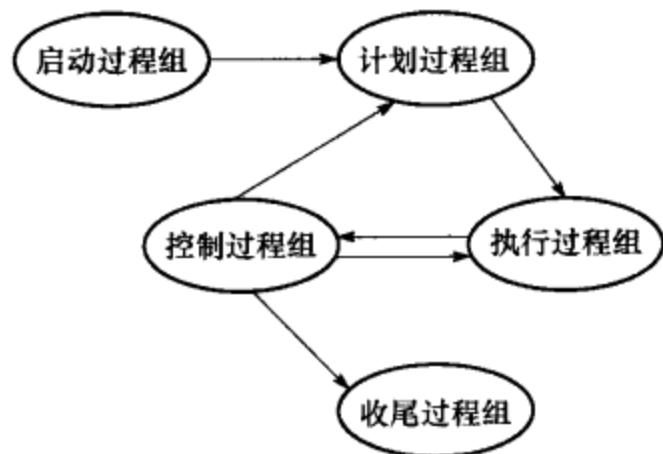


图0-6 项目管理的5个过程组

各个过程组通过其结果进行连接，一个过程组结果或输出是另一个过程组的输入。其中，计划过程组、执行过程组和控制过程组是核心管理过程组。

表0-1是PMBOK的5个标准化过程组、9个知识领域、44个模块之间的关系。

**表0-1 PMBOK的5个标准化过程组、9个知识领域、44个模块之间的关系**

过程组 知识领域	启动过程组	计划过程组	执行过程组	控制过程组	收尾过程组
项目集成管理	项目章程编制 初始项目范围编制	项目计划编制	指导与管理项目执行	项目监控 集成变更控制	项目结束
项目范围管理		范围规划 范围定义 任务分解结构定义		范围核实 范围控制	
项目时间管理		任务定义 任务排序 任务资源估计 任务历时估计 进度计划编制		进度计划控制	
项目成本管理		成本估算 成本预算		成本控制	
项目质量管理		质量规划	质量保证	质量控制	
项目人力资源管理		人力资源规划	人力资源获取 团队建设	团队管理	
项目沟通管理		沟通规划	信息分发	绩效报告	项目干系人管理
项目风险管理		风险管理规划 风险识别 定性风险分析 定量风险分析 风险应对计划		风险监控	
项目采购管理		采购计划编制 合同计划编制	供方反馈获取 供方选择	合同管理	合同收尾

## 0.5 过程管理与软件项目管理的关系

### 0.5.1 软件过程定义

所谓过程，简单来说就是我们做事情的一种固有的方式。我们做任何事情都有过程存在，小到日常生活中的琐事，大到我们的工程项目。对于做一件事，有过经验的人对完成这件事的过程会很了解，他会知道完成这件事需要经历几个步骤，每个步骤都完成什么事，需要什么样的资源和什么样的技术等等，因而可以顺利地完成工作。没有经验的人对过程不了解，就会有无从着手的感觉。下面两个图可以形象的说明过程在软件开发中的地位。如果项目人员将关注点只放在最终的产品上（如图0-7所示），不关注期间的开发过程，那么不同的开发队伍或者个人可能就会采用不同的开发过程，结果导致开发的产品质量是不同的，有的质量高，有的质量差，完全依赖个人的素质和能力。

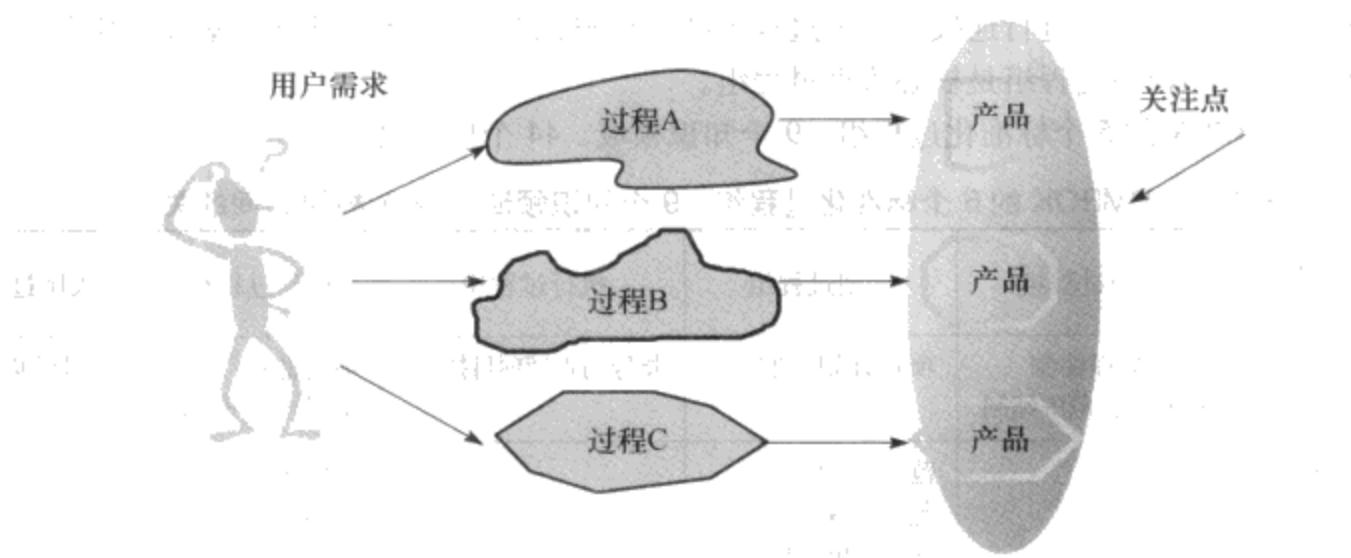


图 0-7 关注开发的结果

反之，如果将项目的关注点放在项目的开发过程，见图 0-8，不管谁来做，也不管什么需求，都采用统一的开发过程，也就是说，企业的关注点在过程。经过同一企业过程开发的软件，产品的质量是一样的。可以通过不断提高过程的质量，来提高产品的质量。这个过程是公司能力的体现，它是不依赖于个人的。也就是说，产品的质量依赖于企业的过程能力，不依赖于个人能力。

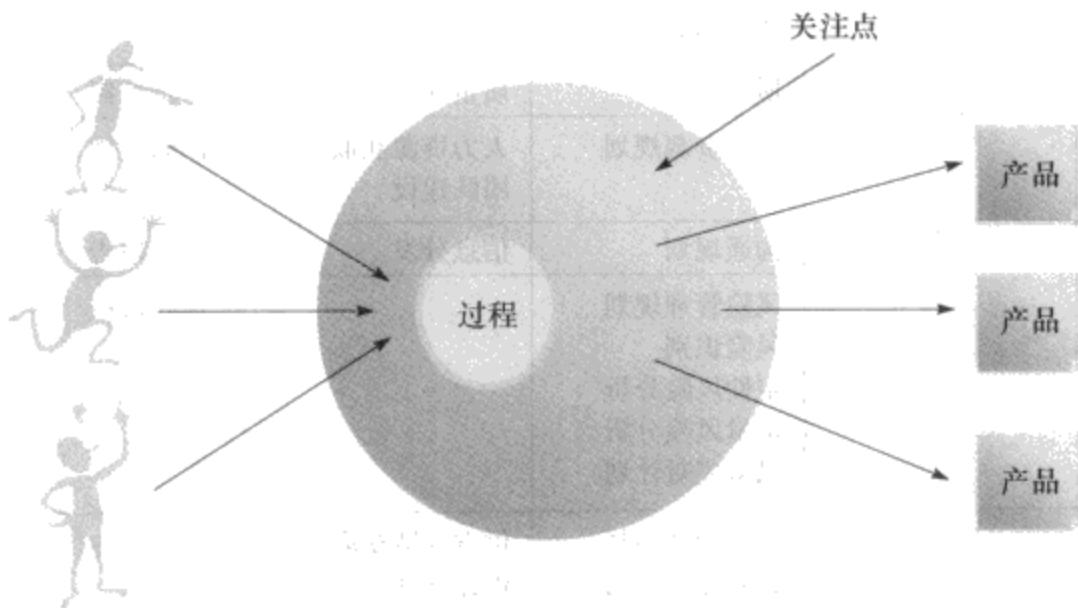


图 0-8 关注开发的过程

对于软件过程的理解，绝对不能简单地理解为软件产品的开发流程，因为我们要管理的并不只是软件产品开发的活动序列，而是软件开发的最佳实践。它包括：流程、技术、产品、活动间关系、角色、工具等，是软件开发过程中的各个方面的因素的有机结合。因此，在软件过程管理中，首先要进行过程定义，将过程以一种合理的方式描述出来，并建立起企业内部的过程库，使过程成为企业内部可以被重用的共享资源。对于过程，要不断地进行改进，以不断地改善和规范过程，以帮助提高企业的生产力。如果将一个软件生产类比于一个工厂的生产，那么生产线就是过程，产品按照生产线的规定过程进行生产。

软件开发的风险之所以大，是由于软件过程能力低，其中关键的问题在于软件开发组织不能很好地管理其软件过程，为此必须强调和加强软件开发过程的控制和管理。软件项目的开发过程主要有系统调研、需求分析、概要设计、详细设计、编码、测试、实施与维护等。对于不同的软件项目，虽然大体上过程相同，但不同的项目其每一个过程所包含的一系列具体的开发活动（子过程）千差万别，而且不同的项目组采用不同的开发技术、使用不同的技术路线，其开发过程的侧重点也不一样。因此项目经理在软件项目开发前，根据所开发的软件项目和项目组的实际情况，建立起一个稳定、可控的软件开发过程模型，并按照该过程来进行软件开发是项目成功的基本保证。

软件过程是极其复杂的过程。我们知道，软件是由需求驱动的，有了用户应用的实际需求才会引发开发一个软件产品。软件产品从需求的出现到最终的产品出现，要经历一个复杂的开发过程。软件产品在使用时要根据需求的变更进行不断的修改，这称为软件维护。我们把用于从事软件开发及维护的全部技术、方法、活动、工具，以及他们之间的相互变换统称为软件过程。由此可见，软件过程的外延非常之大，包含的内容非常之多。对于一个软件开发机构来说，做过一个软件项目，无论成功与否，都能够或多或少地从中总结出一些经验。做过的项目越多，其经验越丰富，特别是一个成功的开发项目是很值得总结的，从中可以总结出一些完善的过程，我们称之为最佳实践（best practices）。最佳实践开始是存放在成功者的头脑中的，很难被机构内部共享和重复利用，发挥其应有的效能。长期以来，这些本应从属于机构的巨大的财富被人们所忽视，这无形中给机构带来了巨大的损失，当人员流动时这种企业的财富也随之流失，并且也使这种财富无法被其他的项目再利用。过程管理，就是对最佳实践进行有效的积累，形成可重复的过程，使我们的最佳实践可以在机构内部共享。过程管理的主要内容包括过程定义与过程改进。过程定义是对最佳实践加以总结，以形成一套稳定的可重复的软件过程。过程改进是根据实践中对过程的使用情况，对过程中有偏差或不切合实际的地方进行优化的活动。通过实施过程管理，软件开发机构可以逐步提高其软件过程能力，从根本上提高软件生产能力。

### 0.5.2 过程管理在软件项目中的作用

上面我们已经看到了过程在软件开发中的重要性。对于软件这种产品来讲，软件过程具有非常重要的意义。对一件家具，它的质量好坏主要有两方面的因素。一是用于生产这件家具的材料的质量要好，否则很难有好的家具。再就是生产的加工工艺要好。早期的家具是手工制造为主，那么由于工匠的手艺不同，产品的质量自然参差不齐，由于技术的不断发展，材料上得到了进一步的提高，同时在产品的加工上，更多地引入了高技术含量的木工机械，所以产品的加工能力和质量的稳定性都得到了很大程度的提高。在软件这种产品的生产上，我们说有一定的特殊性。首先，软件产品没有物理的存在实体，它是完全的逻辑的高度聚合体，那么在质量因素的构成上，材料质量的因素就没有了，那么，在生产过程中唯一影响产品质量的就是产品的生产工艺，这个生产工艺在软件工程中的术语就是软件过程。软件过程管理对于软件产业的发展非常重要。软件产业的发展基础不能永远是零，软件产业发展中的重要问题就是要注重循序渐进地积累，不单是积累技术实践，更为重要的是积累我们所欠缺的管理实践，积累项目中的各个环节的实践经验和项目管理的实践经验，这样才能保证我们的生产力持续地发展，满足业务发展的需要。

软件过程管理，将帮助软件组织将过程资产进行有效管理，使之可以被复用在实际项目中，并结合从项目中获取的过程的实际应用结果来不断地改进过程。这样软件组织将能够有能力改变自身的命运，将它从维系在一个或几个个体身上变成维系在企业中的管理上。过程管理能够让软件组织直观感觉到的一个最明显的转变就是软件项目中的所有成员的位置可以替换。

### 0.5.3 过程管理与项目管理的关系

过程管理，顾名思义，就是对过程进行管理，这种管理的目的是要让过程能够被共享、复用，并得到持续的改进。在软件行业，要管理的是软件过程。过程管理与项目管理在软件组织中是两项最为重要的管理，项目管理用于保证项目的成功，而过程管理用于管理最佳实践。但这两项管理并不是相互孤立的，而是有机地紧密地结合的。图 0-9 中展现的是过程管理和项目管理的基本关系。过程管理的成果即软件过程可以在项目管理中辅助于项目管理的工作，在项目的计划阶段，项目计划的最佳参考是过去的类似项目中的实践经验，这些内容通过过

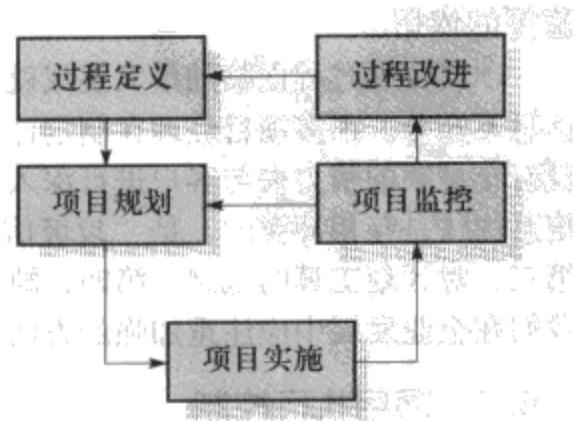


图 0-9 项目管理与过程管理的关系

程管理都成为了过程管理的工作成果，这些成果对于一个项目的准确估算和合理计划非常有帮助，合理的计划是项目成功管理的基础。在项目计划的执行过程中，计划将根据实际情况不断地得到调整，直到项目结束时，项目计划才能被真正稳定下来。这份计划及其变更历史将是过程管理中的过程改进的最有价值的参考。在国外的成熟的软件组织内部，每个项目的开发完成后必须提供一个《软件过程改进建议》的文档，这是从软件开发项目的过程中提炼出来的对软件过程改进的建议。过程的改进就是注重从项目的实际经验中不断地将最佳实践提炼出来。

## 0.6 软件项目管理过程

软件项目管理不同于其他的项目管理，有很多的特殊性。软件是一个特殊的领域，远远没有建筑工程等领域的规范化。软件目前有很大的发展空间，经验在项目管理中占有很重要的作用，理论和标准还在发展中，它体现了软件“软”的特色。

为实现项目目标，使软件项目获得成功，需要对软件项目的范围、可能的风险、需要的资源、实现的任务、成本，以及进度的安排等做到心中有数。而软件项目管理可以提供这些信息，它贯穿于项目的始终。

在本书中，将软件项目管理分为4个阶段：项目初始、项目计划、项目执行控制、项目结束。每个阶段有更多的过程或者程序。

### 0.6.1 项目初始

软件项目管理的第一个阶段是确定项目的目标范围，包括开发和被开发双方的合同（或者协议），软件要完成的主要功能，以及这些功能的量化的范围，项目开发的阶段周期等。软件的限制条件，性能，稳定性等都必须明确的说明；必须满足客户的要求。

软件企业实施项目管理的挑战，可以说是贯穿于项目的整个生命周期。项目初始过程面临的第一挑战是“项目目标含糊，充满冲突”。项目的项目干系人，如项目发起人、成果使用者、负责单位等之间对需求理解不一致，对项目的目标设定不一致。初始过程面临的第二个挑战是“交流语言不规范，缺乏沟通技巧和工具”。以上两方面的挑战导致了在项目初始过程中，很难使项目目标被清晰定义及一致理解。除此之外，对项目目标一致性重视程度不够，也是项目初始过程中普遍存在的一个问题。很多项目管理者低估了达成项目目标一致性的难度，在这方面投入的精力不够，往往简单地认为已经达到一致。因此，很多项目其实是在目标没有定义清楚的情况下匆忙启动的。

### 0.6.2 项目计划

项目计划是建立项目行动指南的基准，包括对软件项目的估算、风险分析、进度规划、人员的选择与配备、产品质量规划等，它指导项目的进程发展。规划建立软件项目的预算，提供一个控制项目成本的尺度，也为将来的评估提供参考，它是项目进度安排的依据。最后，形成的项目计划书将作为跟踪控制的依据。软件项目计划是一个用来协调所有其他计划，以指导项目执行和控制的可操作的文件。它体现了对客户需求的理解，是开展项目活动的基础，是软件项目跟踪与监控的依据。

项目计划过程面临的最大挑战就是计划的准确性差。产生这个问题的原因是多方面的。首先，是信息不充分。很多项目经理在制定计划，尤其是制定一个新的项目计划时没有认真地去挖掘项目信息，没有花足够的精力去与客户进行深入交流。由于没有真正使项目目标达到一致，因此项目人员获取的信息往往是互相冲突的，是“垃圾信息”，对制定计划没有有效的帮助。第二，缺乏科学的估算方法。第三，对计划工具的抵制。第四，缺乏对数据的统计积累。这也是一个非常具有挑战性的方面，也是我们在企业发展中应注重加强的方面。

### 0.6.3 项目执行控制

一旦建立了基准计划就必须按照计划执行，包括按计划执行项目和控制项目，以使项目在预算内、

按进度、使顾客满意地完成。在这个阶段，项目管理过程包括测量实际的进程，并与计划进程相比较。同时发现计划的不当之处。为了测量实际的进程，掌握实际上已经开始或结束的是哪些任务，已经花了多少钱，这些都很重要。如果实际进程与计划进程的比较显示出项目落后于计划、超出预算或是没有达到技术要求，就必须立即采取纠正措施，以使项目能恢复到正常轨道，或者更正计划的不合理之处。

项目执行控制过程面临的挑战是由于计划不准确、关键路径不能锁定，从而导致里程碑目标不能保证项目目标，项目实施的时间压力增大；导致资源调动配置不合理，成本上升。在时间和成本的双重压力下，公司的质量管理很容易流于形式。而在项目控制过程中，由于受项目时间和资源的限制，项目管理者必须能够准确及时地了解各方面的变化，以及由此带来的连锁反应，并做出相应的系统调整。然而，在没有有效数据积累和信息不充分的情况下，这方面的难度很大。

#### 0.6.4 项目结束

项目管理的最后环节就是软件项目的结束过程，前面介绍了项目的特征之一是它的一次性。有起点也有终点，进入项目结束期的主要工作是适当地做出项目终止的决策，确认项目实施的各项成果，进行项目的交接和清算等，同时对项目进行最后评审，并对项目进行总结。

在项目结束过程中，关于时间、质量、成本和项目范围的冲突在这个过程中集中爆发出来。这些冲突主要表现在三个方面：一是客户与项目团队之间，项目团队可能认为已经完成了预定任务，达到了客户需求，而客户并不这样认为；二是项目团队与公司之间，项目团队可能认为自己已经付出了艰苦的努力，已经尽到了责任，然而公司却因为项目成本上升和客户满意度不高并没有获得利润；三是项目成员之间，由于缺乏科学合理的评价体系，项目完成后的成绩属于谁、责任属于谁的问题往往造成团队成员之间互相不理解。

### 0.7 本书的组织结构

本书分为四篇，代表软件项目管理的4个主要阶段，即项目初始、项目计划、项目执行控制、项目结束。在项目计划篇中，又分10章，分别是范围计划、时间计划、成本计划、质量计划、人力资源计划、沟通计划、风险计划、合同计划、配置管理计划、集成计划。见图0-10。

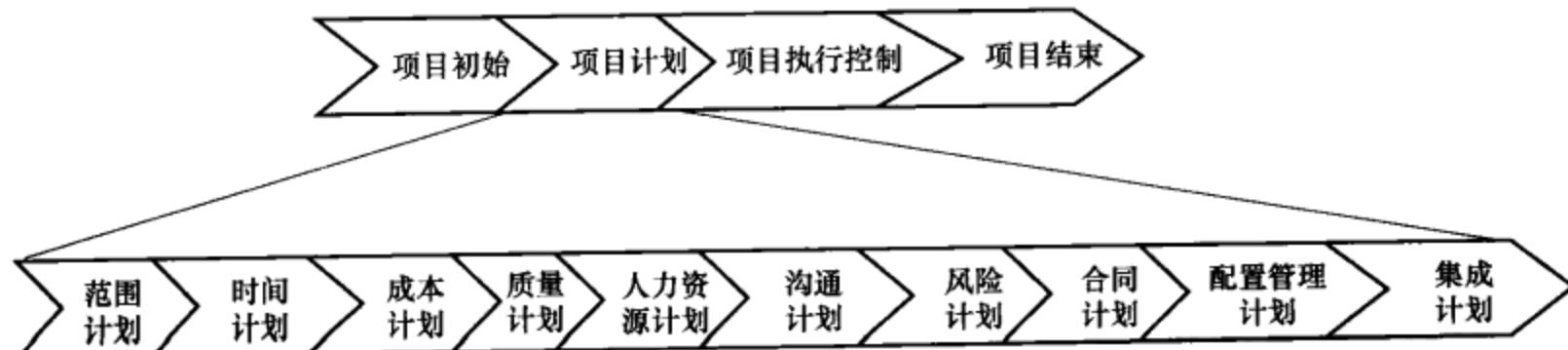


图0-10 软件项目管理路线图

### 0.8 小结

本章讲述了软件项目管理的概念、特点、过程及其重要性。项目管理的范围从战术上看包括产品范围、进度、成本的三角形；从战略上看，包括对人、问题、过程的管理。过程管理在软件项目管理中有着重要的作用，通过不断地优化和规范过程，可以帮助企业提高软件生产能力。软件项目管理的过程主要包括：项目初始、项目计划、项目执行控制和项目结束。本章同时也阐述了项目管理九大知识体系，它是项目管理的核心体系。

记住：对软件项目的理解越深刻，项目开发和管理的经验越多，就越能管理好项目。项目管理是渐进明细的过程。它是灵活性和实践性很强的一门学科，不可死记硬背，项目管理没有唯一的标准，

只有最适合特定项目的管理方法。

## 0.9 练习题

### 一、问答题

1. 项目集成管理是什么？
2. 项目管理九大知识领域是什么？
3. 项目成本管理包括哪些过程？
4. 项目管理的五个过程组是什么？
5. 判断下面活动哪些是项目？
  - 上课
  - 野餐活动
  - 集体婚礼
  - 社区保安
  - 开发微软的操作系统
  - 每天的卫生保洁
  - 神州飞船计划

### 二、判断题

1. 项目开发过程中可以无限制地使用资源。（ ）
2. 项目具有暂时的特征。（ ）
3. 项目管理核心的三角形是范围、进度、风险。（ ）
4. 过程管理与项目管理在软件组织中是两项最为重要的管理，项目管理用于保证项目的成功，而过程管理用于管理最佳实践。（ ）
5. 运作管理是从宏观上帮助企业明确和把握企业发展方向的管理。（ ）

# 第一篇

# 项目初始

本书将按照前章介绍的项目管理的路线图讲述软件项目管理的过程，首先从路线图第一阶段即项目初始阶段开始（如下图所示）。



# 第1章

## ■ 软件项目初始过程

这章主要介绍项目初始阶段的主要过程，包括项目的来源、项目的授权、项目的生存期等。在项目实施过程中，项目的利益应该高于一切，所谓项目利益是因项目的成功而给各项目干系人（stakeholder）带来可以分享的利益。因此，确定实施一个项目是需要多方斟酌和考虑的。

### 1.1 项目立项

在项目选择过程中，关键是对项目的定义有明确的描述，应包括明确项目的目标、时间表、项目使用的资源和经费，而且得到执行该项目的项目经理和项目发起人的认可。这个阶段称为“立项阶段”。

立项是要解决做什么的问题。需要确定开发的项目，关注点是效益和利润。这个问题展开就是前期需要投入多少？能否盈利？什么时候能够盈利？能否持久的盈利？整个立项报告的内容都将围绕这些核心内容展开。

企业确定开始某个项目时，一般会下达一个立项的文件，主要内容是遵照的合同或相关协议，例如项目的大致范围、项目结束时间和一些关键时间、指定项目经理和部分项目成员等等。

项目一旦确定，就具有明确的起始日期和终止日期。项目经理的角色不是永久性的，而是暂时的。作为一个项目经理，你的责任是明确目标，规划达到目标的步骤，然后带领你的团队按计划朝着目标前进。

在立项阶段，产品负责人会进行自造 - 购买（make or buy）决策，确定待开发产品的哪些部分应当“采购”、“外包开发”或“自主研发”。自造 - 购买分析是最初定义项目范围时使用的决定自造还是购买的方法。在项目的进展过程中，也需要自造 - 购买分析方法。

除了需要考虑自造或者购买的初始成本，还要考虑后续的大量费用。例如，一个公司准备租一台设备。那么就需要评估租赁设备后续费用与购买设备后续费用的比较情况，以及每月的维护费、保险费和设备管理费等。

例如，图 1-1 显示了决定自己开发一个软件还是从一个软件公司购买软件的数学分析结果。如果选择自己开发软件的策略，公司需要花费 25 000 美元，根据历史信息，维护这个软件每个月需要的费用是 2 500 美元。

如果选择购买软件公司的策略，需要 17 000 美元，同时软件公司为每个安装的软件进行维护的费用是每月 2 700 美元。自己开发软件和购买软件的费用之差是 8 000 美元，而支持维护的费用之差是每月 200 美元。

自行开发费用与购买费用之差除以每个月的 200 美元——即 8 000 美元除以 200 美元，为 40 月。如果这个软件的使用期为 40 个月之内，公司可以考虑

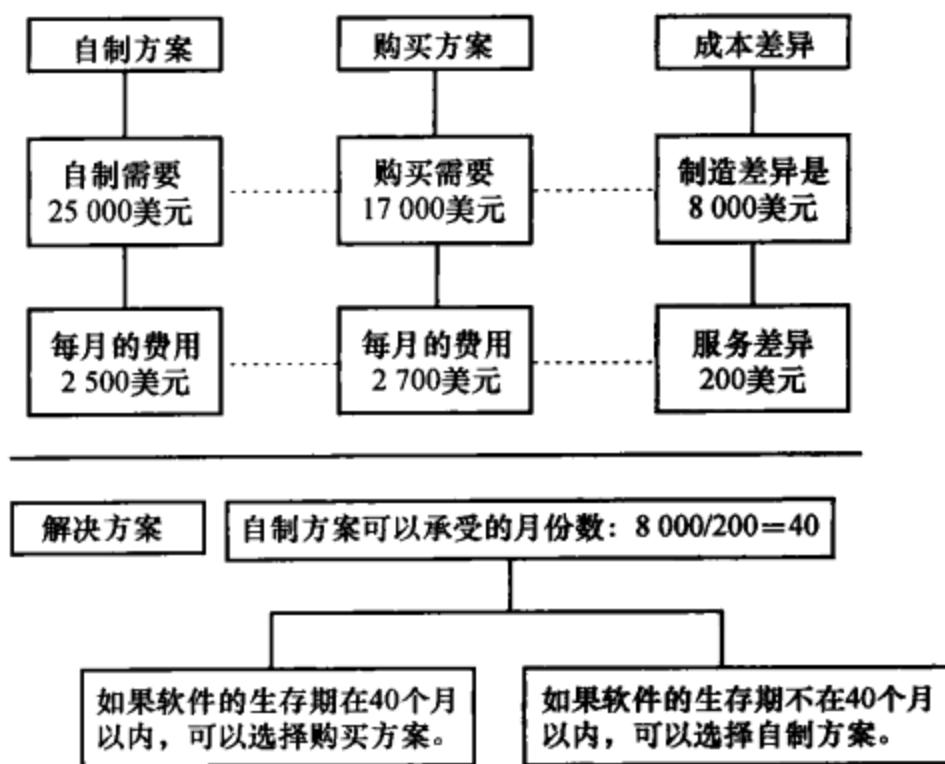


图 1-1 自造 - 购买决策过程

购买软件。否则应该自行开发软件。

一个企业选择自制还是购买的依据有很多。表1-1列出了一些常用的选择依据。

你可能曾经历过，购买需要组装和配置的硬件有很多种方式。有些情况下，对于一个项目来说，使用组装好的硬件是比较合适的，在另外一些情况下，现场组装硬件可以节省成本。

所以，一个项目或者自行开发或者是外包给别人，当项目外包的时候，就存在甲乙方之间的责任和义务的关系。

甲方即需方（有时也称为买方）是对所需要的产品或服务进行“采购”，这覆盖了两种情况，

一种为自身的产品或资源进行采购，另一种是为顾客进行采购（与顾客签定合同的一部分）。“采购”这个术语是广义的，其中包括软件开发委托、设备的采购、技术资源的获取等方面。

乙方即供方（有时也称为卖方）是为顾客提供产品或服务。“服务”这个术语是广义的，其中包括为客户开发系统、为客户提供技术咨询、为客户提供专项技术开发服务及为客户提供技术资源（人力和设备）的服务。

总之，一个项目的来源或者是合同项目或者是内部项目，本章主要介绍合同项目，内部项目可以参照合同项目的一些做法完成。

## 1.2 合同项目

作为合同项目，需要明确甲乙双方的任务。企业在甲（需）方合同环境下的关键要素是提供准确、清晰和完整的需求、选择合格的乙（供）方并对采购对象（‘采购对象’包括产品、服务、人力资源等）进行必要的验收。企业在乙方合同环境下的关键要素是了解清楚甲方的要求并判断企业是否有能力来满足以这些需求。作为软件开发商更多是乙方的角色。

### 1.2.1 甲方初始过程

甲方在初始阶段的主要任务是：招标书定义、乙方选择、合同签署。以下分别对这三个过程进行详细介绍。

#### 1. 招标书定义

启动一个项目主要是由于存在一种需求，项目的需求可能源自企业内部的需要，也可能是为客户开发的软件项目中的一部分，通过寻找合适的软件开发商，将部分软件外包给其他的开发商。招标书定义主要是甲方的需求定义，也就是甲方定义采购的内容，软件项目采购的是软件产品，需要定义采购的软件需求，即提供完整清晰的软件需求和软件项目的验收标准。必要的时候明确合同的要求。最后，潜在的乙方可以拿到这个招标文件。

招标书定义过程如图1-2所示。

甲方在招标书定义过程中的具体活动描述如下：

- 1) 定义采购需求并对采购需求进行评审。
- 2) 根据采购需求确定采购商务条件（如甲乙双方的职责、控制方式、价格等）。
- 3) 制订采购对象的验证、检验标准与方式。
- 4) 收集和汇集其他相关采购资料（如技术标准附件、产品提交清单）。
- 5) 项目决策者负责认可采购需求、验收标准和相关资料。
- 6) 根据上述信息编写招标书（招标文件），必要时可以委托招标公司进行招标。

表1-1 自造-购买决策过程的常用选择依据

自制的理由	购买的理由
自制成本少	购买成本低
可以采用自制的技巧	不会自制
工作可控	工作量小
可以获得知识产权	购买更有益
学习新的技能	转移风险
有可用的开发人员	有很好的供货商
核心项目工作	项目可以将注意力放在其他工作上

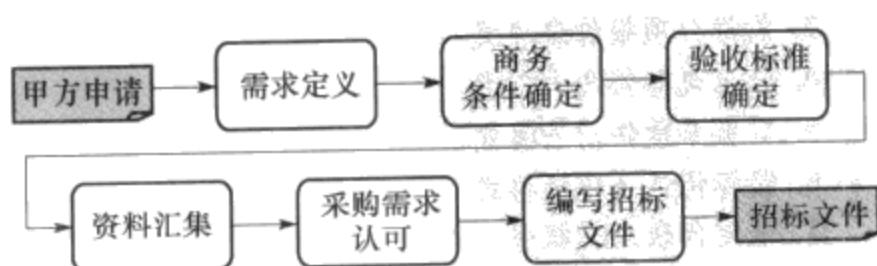


图1-2 招标书定义过程

招标书主要包括的三部分内容：技术说明、商务说明和投标说明。技术说明主要对采购的产品或者委托的项目进行详细的描述。商务说明主要包括合同条款。投标说明主要是对项目背景、标书的提交格式、内容、提交时间等作出规定。招标书是投标人编写投标书的基础，也是签订合同的基础，必须小心谨慎，力求准确完整。如果合同条款存在漏洞，在合同执行过程中，可能双方会发生争议，直接影响了合同的顺利进行，甚至可能造成巨大的经济损失。

招标书一般要明确投标书的评估标准，评估标准用来对投标书进行排序和打分，是选择乙方的依据。包括客观和主观的评定标准，客观标准是事先规定好的明确的要求，例如“乙方需要达到 CMM 3 级以上的要求”，主观标准比较模糊，例如“乙方应该具备同类技术的相关经验”，评估标准一般包括：

- 1) 价格：包括产品以及产品提交后所发生的附属费用。
- 2) 对需求的理解：通过乙方提交的投标书，评定乙方是否完全理解甲方的需求。
- 3) 产品的总成本：乙方所提供的产品是否有最低的总成本。
- 4) 技术能力：乙方是否具备保证项目所需要的技术手段和知识。
- 5) 管理能力：乙方是否具备保证项目成功的管理手段。
- 6) 财务能力：乙方是否具备必要的资金来源。

国际上，招标文件的类型主要有：投标邀请（Invitation For Bidding, IFB）、建议书提交邀请（Request For Proposal, RFP）、报价邀请（Request For Quotation, RFQ）、谈判邀请（Invitation For Negotiation, INF）。

招标书没有统一的格式，可繁可简。例如：下面是某一招标书的主要目录。

- 第一章 投标邀请
- 第二章 投标人须知前附表
- 第三章 投标人须知
  - 一、说明
  - 二、招标文件
  - 三、投标文件的编制
  - 四、投标文件的密封和递交
  - 五、开标与评标
  - 六、授予合同
- 第四章 合同专用条款
- 第五章 合同通用条款
- 第六章 合同格式
- 第七章 XXXXX 网络软件系统规划设计要求与目标
- 第八章 附件（投标文件格式）
  - 1. 投标书格式
  - 2. 开标一览表格式
  - 3. 投标分项报价表格式
  - 4. 技术规格偏离表格式
  - 5. 商务条款偏离表格式
  - 6. 投标保证金保函格式
  - 7. 法定代表人授权书格式
  - 8. 资格证明文件格式
  - 9. 履约保证金保函格式
  - 10. 投标人情况表格式
  - 11. 投标人财务状况表格式
  - 12. 投标人 2006/2007 年的财务报表
  - 13. 投标人专业技术人员一览表格式

14. 投标人近二年已完成的与招标内容相同或相似的项目一览表格式
15. 投标人正在承担的与招标内容相同或相似的项目一览表格式
16. 投标人资产目前处于抵押、担保状况格式
17. 投标人近三年结束正在履行的合同引起仲裁或诉讼的格式

### 第九章 评标标准

招标书编写好后，招标书可以发给（或者卖给）潜在的乙方，邀请他们参加投标，乙方如果认为可以参与竞标，可以提交投标书。

## 2. 乙方选择

招标文件确定后，就可以通过招标的方式选择乙方，招标文件应该对乙方的要求做明确的说明，获得招标文件的乙方根据招标文件的要求，编写项目投标书，并提交给甲方，甲方根据招标文件确定的标准对乙方资格认定，并对其开发能力资格进行确认，最后选择出最合适的乙方。乙方选择过程如图 1-3 所示。

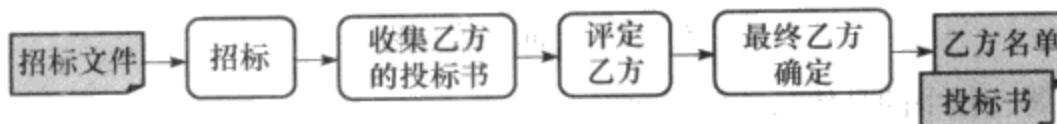


图 1-3 乙方选择过程

甲方在乙方选择过程中的具体活动描述如下：

- 1) 将招标文件发给（或者卖给）具备竞标条件的乙方。
- 2) 组织项目竞标，并获取竞标单位的投标书。
- 3) 根据招标文件的标准和竞标单位的竞标过程以及乙方提交的投标书，确定竞标单位的排名。
- 4) 确定最终选择的乙方名单。

为了选择到合适的乙方，甲方应该让更多的潜在的乙方参与投标，展开竞争，以便获得价格最合理、质量最优的产品。

招标的方式也有很多种，例如有公开招标、有限招标、多方洽谈和直接谈判等。

1) 公开招标是招标信息在社会上公开发布，使一切潜在供应商都获得平等的参与竞争的机会。这种方式强调公平、公开、公正，最具有透明度。供应商之间的竞争可以使甲方可以以最优最低的价格获得产品。这种招标方式的缺点是成本高，比较花时间。

2) 有限招标是招标信息在有限的范围内发布，通常是直接向筛选合格的潜在的供应商发出邀请，这种招标方式比公开招标方式节省成本和时间，但是，不一定能获得最好的产品。

3) 多方洽谈是甲方不明确发出招标信息，而是选择几个潜在的合格的供应商分别进行洽谈，从中选择合适的供应商。这种方式透明度更低。但是对于一些特殊的情况也可能获得最有利的合同。

- 4) 直接谈判是直接与一家供应商谈判并签订合同，只适应一些特殊的项目。

招标之后需要进行评标，评标的目的是从众多的投标人中挑选出能以最合理价格、最好的服务项目的乙方。在此过程中，项目经理需要与专家一起，根据评估标准和相关策略，对所有的投标书进行评估、选择、通过合同谈判，最终确定供应商，并签署合同。

在评标过程中主要分两个阶段，第一阶段是初评阶段，采用筛选系统将一部分不满足评估标准中最低资格要求的投标书筛选出去。在第二阶段主要进行细评工作，对通过初评的投标书的各个方面进行量化打分，按照分值将投标人排序，以此决定进行合同谈判的顺序，或者直接与得分最高的投标人签署合同。

## 3. 合同签署

如果甲方选择了合适的乙方（软件开发商），而且被选择的开发商也愿意为甲方开发满足需求的软件项目，那么为了更好的管理和约束双方的权利和义务，以便更好的完成软件项目，甲方应该与乙方（软件开发商）签订一个具有法律效力的合同。签署之前需要起草一个合同文本。双方就合同的主要条款进行协商，达成共识，然后按指定模板共同起草合同。双方仔细审查合同条款，确保没有错误

和隐患，双方代表签字，合同生效。

合同签署过程就是正式签署合同，使之成为具有法律效力的文件，同时，根据签署的合同，分解出合同中甲方的任务，并下达任务书，指派相应的项目经理的过程。合同签署过程如图 1-4 所示。

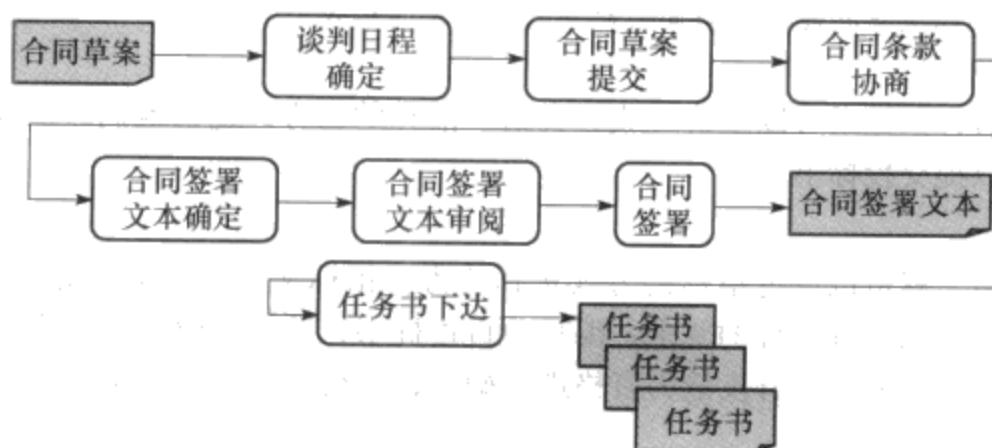


图 1-4 合同签署过程

甲方在合同签署过程中的具体活动描述如下：

- 1) 制订合同草案。
- 2) 确定项目定义，确定甲乙方权利和义务，并将结果反映到合同条款中。
- 3) 确定项目的验收和提交方式（如验收标准、产品介质、包装和复制要求），并将结果反映到合同条款中。
- 4) 确定合同其他有关条款，并将结果反映到合同条款中。
- 5) 对制订的采购合同草案进行评审。
- 6) 根据评审结果对合同草案进行修改并确认，形成最终合同草案。
- 7) 确定谈判日程和谈判所涉及的人员。
- 8) 在谈判日程所规定的谈判时间前向乙方提供合同草案。
- 9) 按谈判日程和谈判要点与乙方讨论并形成合同签署文本。
- 10) 项目决策者审阅合同签署文本。
- 11) 根据甲方项目决策者审阅意见签署或终止合同谈判。
- 12) 将合同签署文本（无论是否经双方签署）及合同相关文档存档保存。
- 13) 根据合同条款，分解出甲方所需执行的活动或任务，编写任务书，确定项目经理。
- 14) 项目经理对任务书进行确认。

在签署合同的时候，甲方会同时将工作任务说明（Statement Of Work, SOW）作为合同附件提交给乙方。工作任务说明是甲方描述的实现开发约定所要执行的所有任务。

### 1.2.2 乙方初始过程

项目的选择是项目型企业业务能力的关键核心。项目选择过程，是指从市场上获得商机到与客户签订项目合同的过程。过程开始于收集项目商机，进行简单评估，确定可能的目标项目，初步选择适合本企业的项目，然后对项目进一步分析，与客户进行沟通，制定项目方案和计划，通常还需要与客户进行反复交流，参加竞标，直到签订合同才算完成项目的选择过程。

作为乙方的软件企业可能会在项目开发过程中将项目中的一部分外包给另外一个软件公司，它同样需要选择一个合适的乙方。这时，这个软件企业就既是甲方的角色，也是乙方的角色。

乙方在初始阶段的主要任务是：项目分析、竞标、合同签署。以下分别对这三个过程进行详细介绍。

#### 1. 项目分析

项目分析是乙方分析用户的项目需求，并据此开发出一个初步的项目规划的过程，作为下一步能力评估和可行性分析之用。项目分析过程如图 1-5 所示。

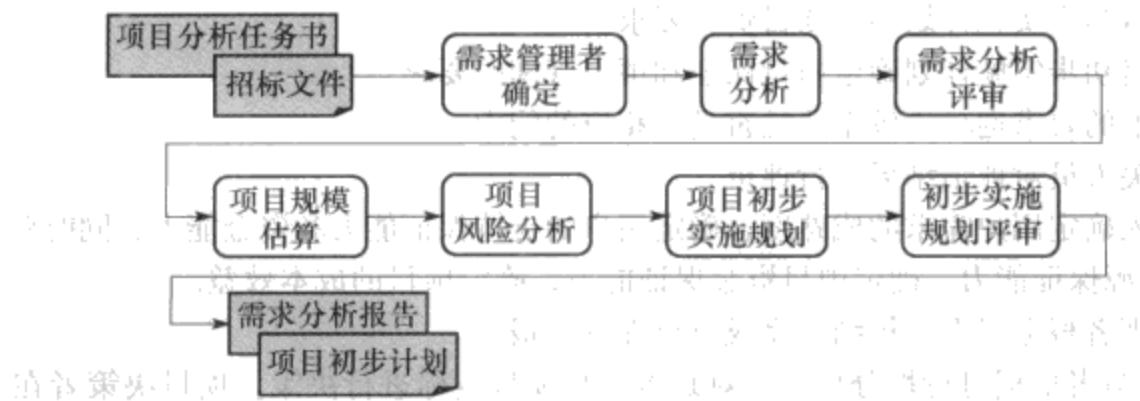


图 1-5 项目分析过程

乙方在项目分析中的具体活动描述如下：

- 1) 供方确定需求管理者。
- 2) 需求管理者负责组织人员分析项目需求，并提交需求分析结果。
- 3) 邀请用户参加对项目需求分析结果的评审。
- 4) 项目管理者负责组织人员根据输入和项目需求分析结果确定项目规模。
- 5) 项目管理者负责组织人员根据需求分析结果和规模以及估算结果，对项目进行风险分析。
- 6) 项目管理者负责组织人员根据项目输入、项目需求和规模要求，分析项目的人力资源要求、时间要求及实现环境要求。
- 7) 项目管理者根据分析结果制定项目初步实施规划，并提交合同管理者评审。
- 8) 合同管理者负责组织对项目初步实施规划进行评审。

## 2. 竞标

竞标过程是乙方根据招标文件的要求进行评估，以便判断企业是否具有开发此项目的能力，并进行可行性分析，可行性分析是判断企业是否应该承接此软件项目，项目是否可行。首先判断企业是否有能力完成此项目，另外判断企业通过此项目是否可以获得一定的回报。如果项目可行，企业将组织人员编写项目投标书，参加竞标。具体过程如图 1-6 所示。

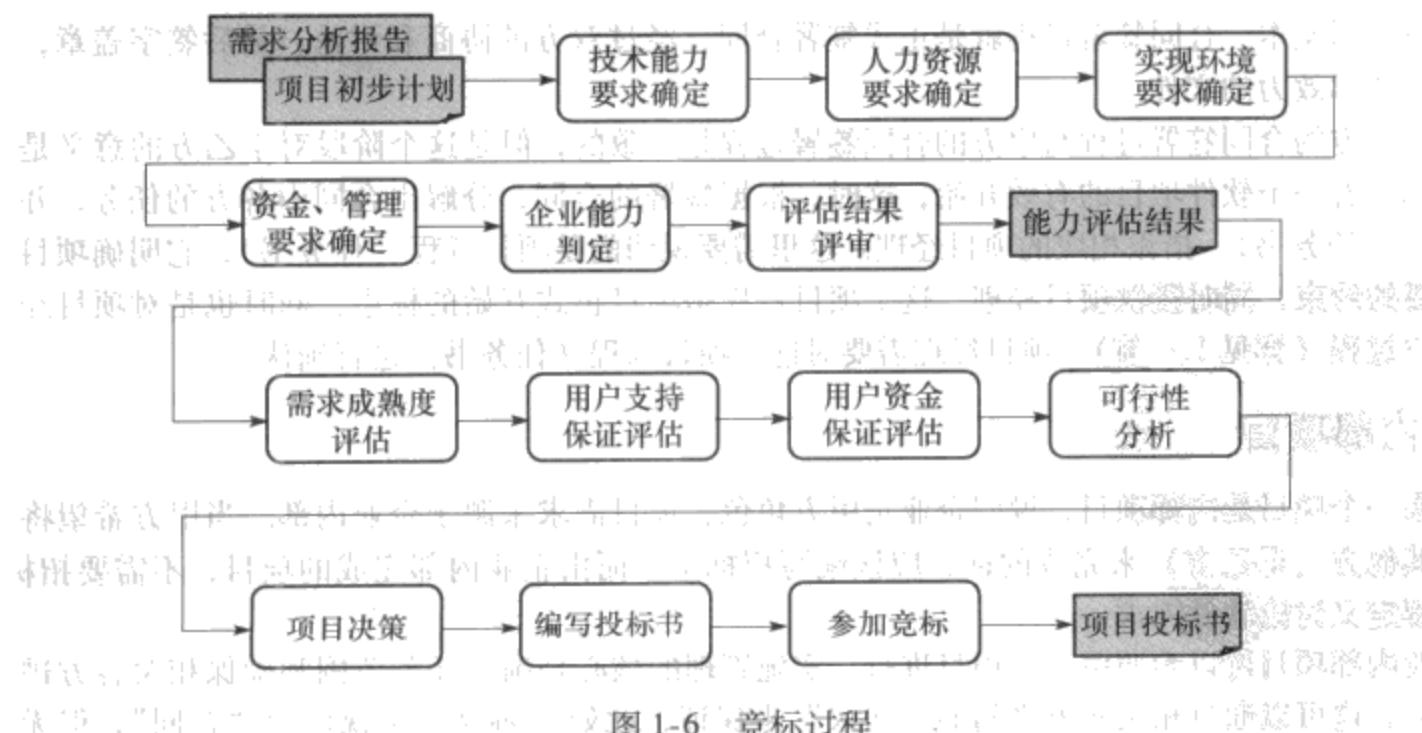


图 1-6 竞标过程

乙方在竞标过程中的具体活动描述如下：

- 1) 根据项目需求分析报告确定项目技术能力要求。
- 2) 根据项目初步实施计划确定项目人力资源要求。
- 3) 根据项目需求分析报告确定项目实现环境要求。

- 4) 根据项目初步实施计划确定项目资金要求。
- 5) 根据项目初步实施计划确定质量保证和项目管理的要求。
- 6) 根据以上的要求逐项比较企业是否具有相应的能力。
- 7) 组织有关人员对评估结果进行评审。
- 8) 根据输入确定用户需求的成熟度，确定用户的支撑保证能力和资金能力，同时确定企业技术能力，确定人力资源保证能力，确定项目资金保证能力，确定项目的成本效益。
- 9) 合同管理者根据以上分析结果完成可行性分析报告。
- 10) 项目决策者根据可行性分析报告对是否参与项目竞标进行决策。项目决策者在进行项目决策时应主要考虑以下几个方面：
  - 技术要求：技术要求是否超出公司的技术能力。
  - 完成时间：用户所要求的完成时间是否合理，公司是否有足够的保证资源。
  - 经济效益：可能的合同款项是否能覆盖所有的成本并有收益。
  - 风险分析：项目的风险和风险控制方式。
- 11) 如果乙方决定参与竞标，应该组织相关人员编写投标书。

投标文件主要有两种类型，一个是建议书（proposal），另外一个是报价单（quotation）。建议书是乙方根据甲方提出的产品的性质、目标、功能等，提交的完整的技术方案和报价等。报价单主要是乙方根据甲方提出的产品特定型号、标准和数量等要求提交必要的报价材料等。

一般说，如果乙方竞标一个软件开发项目而不是一个软件产品的话，这个过程的关键是编写并提交建议书。项目建议书是指在项目初期为竞标或签署合同而提交的文档，它是在双方对相应问题有共同认识的基础上，清晰地说明项目的目的以及操作方式，可以决定项目有无足够吸引力或是否可行。它是乙方描述甲方需求，并提出解决方案的文档，通过建议书可以展示乙方对项目的认识程度和解决问题的能力，也是甲方判断乙方能否成功完成任务的重要依据。

### 3. 合同签署

对于合同，甲乙双方可能都要准备合同文本，当然，一般是甲方提供合同的框架结构，起草主要内容，乙方提供意见；有时，乙方可能根据甲方的要求起草合同文本，甲方审核；当然有时双方可以同时编写合同文本。合同签署过程就是正式签署合同，经过双方的协商和讨论，最后签字盖章，使之成为具有法律效力的文件。

作为乙方的合同签署过程与甲方的合同签署过程是一致的，但是这个阶段对于乙方的意义是重大的，它标志着一个软件项目的有效开始，这时，根据签署的合同，分解出合同中各方的任务，并下达项目章程（任务书），指派相应的项目经理。这里需要说明的是项目章程（任务书），它明确项目的目标、必要的约束，同时授权项目经理。这个项目章程是项目正式开始的标志，同时也是对项目经理的有效授权过程（详见 1.4 节）。项目经理需要对这个项目章程（任务书）进行确认。

## 1.3 内部项目

如果一个项目是内部项目，说明企业是甲方角色，项目需求来源于企业内部。当甲方希望将项目委托给其他方（即乙方）来完成的话，应该编写招标书，而由企业内部完成的项目，不需要招标了，这时需要定义讨论软件需求。

企业内部项目可以参照甲乙方项目进行，实施管理的核心是确定任务范围和确保相关各方进行有效的配合，这可以通过相关各方之间的“协议”来保证，此处“协议”可视作为“合同”，但无特别的商业约束。对于项目承接方（乙方）的能力也不做任何认定要求，因为内部项目往往包括研究和了解新技术的成分，若承接方不具有相应的能力，将通过培训解决。

尽管内部项目和合同项目本质是一致的，都存在甲方和乙方的问题，但是由于利益等关系不同，内部项目中甲方和乙方没有具有法律约束的合同等，在项目进行过程中，对项目范围、成本、进度、质量等方面管理没有合同项目执行的严格。

## 1.4 项目授权

启动一个项目的重要原因是满足商业目标，提供一个合理的投资回报等，项目发起人组织相关人员提出需求，确定项目管理者及其相关成员，编写必要的项目说明书。

当选择了一个项目之后，就需要对这个项目进行授权和初始化，以便确认相关的人知晓这个项目。这就需一个文档化的输出，这个文档可以有很多不同的形式，一个最主要的形式是项目章程。

### 1.4.1 项目章程

**项目章程**（project charter）是指项目执行组织高层批准的一份以书面签署的确认项目存在的文件，包括对项目的确认、对项目经理的授权和项目目标的概述等。严格说，项目章程包括对开始一个项目或者项目阶段的正式授权，但是通常而言，在每个项目阶段都进行一次授权的做法并不多见。

项目章程是一个正式的文档，它正式的认可一个项目的有效性，并指出项目的目标和管理方向。它授权项目经理来完成项目，从而保证项目经理可以组织资源用于项目活动。项目章程通常由项目发起人、出资人或者高层管理人员等签发。

项目章程和项目目标类似，但章程更加正式，叙述也更加完整详尽，更符合公司的项目视图和目标。项目目标使这种描述更加具有特定性，并且加上了截止日期。而项目章程使目标形式化，就好像是到达目的地的一张通行证。总之，项目章程正式的授权项目。有的项目没有明确的项目章程，但是合同也可以作为项目章程。

章程不仅清楚地定义了项目，说明了它的特点和最终结果，还指明了项目权威。项目权威通常是项目的发起人、项目经理和团队领导（如果需要的话），章程可以详细规定了每个人的角色，以及相互交流信息的方式。

当然不同企业的做法是不一样的，有不同的形式，例如有的企业采用一个简单的协议，有的是一个很长的文档，或者使用合同作为项目章程。

建立项目章程时，你可以把任何想要加入的信息都包括进去。一般说来，包括下列要素：

- 项目的正式名称。
- 项目发起人及联系方式。
- 项目经理及联系方式。
- 项目目标。
- 关于项目的业务情况（项目的开展原因）。
- 项目的最高目标和可交付成果。
- 团队开展工作的一般性描述。
- 开展工作的基本时间安排（详细的时间安排在项目计划中列举）。
- 项目资源、预算、成员以及供应商。

无论采用哪种形式，这个过程正式的授权项目开始，任命项目经理，说明项目的背景，来源等，一定的假设或者约束等。例如，表1-2是一个项目章程的例子。

表1-2 IT升级项目的项目章程

项目题目：IT升级项目	项目结束时间：2008.5.15
项目开始时间：2008.3.10	
项目经理：赵强， Zhaoqiang@abc.com	
Project Objectives：根据新的企业标准对企业所有人员的软件硬件进行升级，新的标准见附件。升级可能影响服务器和一些计算机以及一些网络的软硬件。软硬件的费用300万元，人工成本为20万。	
建议方式方法：	
<ul style="list-style-type: none"> <li>• 升级数据库目录</li> <li>• 做详细的成本估算，然后上报</li> <li>• 获取软硬件报价</li> <li>• 尽可能用内部的人员参与项目</li> </ul>	

(续)

人员	角色	职责
赵强	项目经理	规划、监控项目
王立	质量经理	负责项目的质量
江明	技术经理	负责技术
章溢	系统支持	负责产品的所有系统、网络
韩斌	采购经理	负责采购软硬件

签字: XXX, XXX

注释:

XXXXXXX

项目章程类似项目的授权书，相当于对项目的正式授权，表明项目可以有效地开始了。项目章程授权项目，它建立了项目经理的责任心、发起人的主人翁意识以及项目团队的团队意识。项目章程可以解决很多令人头痛的问题，确立负责人，帮助你更加自信地快速向目标前进。

#### 1.4.2 项目经理的责任和权利

项目经理是项目组织的核心和项目团队的灵魂，对项目进行全面的管理，他的管理能力，经验水平、知识结构、个人魅力都对项目的成败起着关键的作用。同时作为团队的领导者，他的管理素质、组织能力、知识结构、经验水平、领导艺术等都对团队管理的成败有着决定性的影响。在一个特定的项目中，项目经理要对项目实行全面的管理，包括制定计划，报告项目进展，控制反馈，组建团队，在不确定环境下对不确定性问题进行决策，在必要的时候进行谈判及解决冲突等。其中组建团队是项目经理的首要责任，一个项目要取得好的成绩，一个关键的要素就是项目经理应该具备把各方人才聚集在一起，组建一个有效的团队的能力。在团队建设中，要确定项目所需人才，从各有关职能部门获得人才，定义成员任务和角色，把成员按任务组织起来形成一个高效的团队。要建立并使团队有效运行，项目经理需要起关键的作用。

项目经理是沟通者，团队领导者，决策者，气氛创造者等多个角色的综合。以身作则与有威信是相辅相成的。规范制度的权威性主要还是靠项目经理自己，只有坚持以身作则，才能将自己优秀的管理思想在整个项目中贯穿下去，取得最后的成功。项目经理关系到一个项目的成败，是自己的责任就要敢于承担。有相应的权利就必然有相应的责任。如果不负责任，项目管理中就可以不再需要项目经理了。当然负责归负责，退一步而言并不是要项目经理负项目中所有的责任。但只有项目经理敢于负责，才能使得责任对应的个人有勇气站出来！而这样也将使项目朝更快更好的方向发展。项目经理的职责可以总结为如下：

1) 开发计划。项目经理的首要任务就是计划、计划、再计划。计划安排项目工作，使各项目工作形成有机整体。完善合理的计划对于项目的成功至关重要。项目经理要在对所有的合同、需求等熟知、掌握的基础上，明确项目目标，并就该目标与项目客户达成一致，同时告知项目团队成员、然后为实现项目目标制定基本的实施计划（成本、进度、产品质量），在项目的实施过程当中，还要根据项目 的实际进展情况，在必要的时候，调整各项计划方案。

2) 组织实施。同有关部门联络，选择并确定项目团队的职责。确定职能专业部门和其他项目参与者之间的分工，有效的调用项目团队和每个成员。项目经理组织实施项目主要体现在两个方面：其一，设计项目团队的组织结构图，对各职位的工作内容进行描述，并安排合适的人选；以及对项目所需的人力资源进行规划、开发。其二，对于大型项目，项目经理应该决定哪些任务由项目团队完成，哪些由承包商完成。

3) 项目控制。项目实施过程中，项目经理要时时监视控制项目的运行，积极预防，防止意外的发生。及时解决出现的问题，同时要预测可能的风险和问题，保证项目在预定的时间、资金、资源下顺利完成。

既然项目经理担负着确保项目成功的重大责任，那么就必须赋予项目经理一定的权利，保证项目得以顺利实施。实际上，尽管项目经理对项目负有主要职责，但由于大多数项目资源（如人力资源）

不直接受项目经理的控制，职权与职责并不是统一在项目经理一个人身上，因此，项目经理地位最主要的一个特点是“责任大于权利”，项目经理在管理项目中会更多的依赖个人权利。通常项目经理主要有以下三方面权利：

- 1) 制定项目的有关决策。项目在实施过程当中必然会面临各种各样的决策，而制定决策是项目经理所拥有的最主要的权利，这也是项目经理最基本、最重要的权利。
- 2) 挑选项目成员的权利。项目组织是一个临时性的组织，项目成员为了一个共同的目标而团结在一起。项目启动后，项目经理有权根据自己的判断、自己的方式选择项目成员，组建项目团队。
- 3) 对项目获得的资源进行再分配。上级组织将资源划拨给项目组织，项目经理有权决定这些资源的具体使用，根据项目具体工作要素的情况进行资源再分配。

### 1.4.3 项目经理的能力

专职的项目经理是必需的，他是团队的灵魂人物。项目经理的责任重在管理，同时也应具备相应地技术能力。一般说，作为项目经理应该具备基本的计算机及网络的应用能力、对IT新技术的接受（了解）能力、较强的自我更新能力等。同时，也应该具备沟通能力、协调能力、项目控制能力、资源管理与控制能力、服务意识与能力，当然也应该具备个人人格魅力等。

经调查，作为一个理想的项目经理，它应具备的特点如下：

- 对项目的目标有透彻的理解。
- 具备了解项目成员需要的能力。
- 对细节问题头绪清楚。
- 对项目有强烈的责任心。
- 能应付挫折和失望。
- 具备很好的谈判技巧。
- 面向成果，注重实际。
- 有成本意识和基本的经营技巧。
- 有政治头脑——知道该做什么，不该做什么。
- 能忍受模糊不清的煎熬。

通过项目经理职责，我们可以看出项目经理在项目中所担当的角色，其中重要的角色应该是组织协调能力。计算机信息系统涉及的领域较多，知识面较广，技术复杂度有时候是难以预料的，项目经理不可能对每个领域都知道，不可能对所有的技术都了解，这就要求项目经理要调动项目团队的积极性，激发各个专业人才的工作潜能，做好团队内部沟通和纵横向沟通。

当前，在政府机关和企事业单位软件项目管理中存在主要问题是，没有专门项目管理专业的人员来担任项目经理。被任命的项目经理主要是因为他们能够在技术上独当一面，而管理方面特别是项目管理方面的知识比较缺乏。因此，项目经理接受系统的项目管理知识培训是非常必要的。

## 1.5 初始项目范围

在项目的初期，一般会开发初始的项目范围说明书，说明项目所需要完成的工作和所需要提交的成果。由于项目具有渐进明晰的特性，这个初始的范围说明书也需要不断的完善。

一个初始的项目范围说明，相当于确定初始的项目需求说明书，对项目需求进行初步的描述，将来编写需求规格书的时候，可以在此基础上进行详细的描述。

## 1.6 生存期模型

在一个项目生存期中，每一项任务都通过一个或者多个过程的方式来完成的，在生存期中所有这些相关过程的组合，称为软件生存期过程。建立软件生存期过程可以使用两个基本的程序，第一个程序是软件开发生存期全局模型的选择，这个全局过程模型称为一个生存周期模型；第二个程序是特定

阶段及其任务所执行的一些单个过程的选择。

在生存期模型中定义软件过程非常重要，人和过程是保证项目成功的两个最关键因素。由好的人按好的过程进行项目开发，才能最大限度地保证项目的成功。一个好的过程可以保证差的人做出来的东西不至于太差，但不能确保做出精品。通过过程可以实现一种规范化、流水线化、工业化的软件开发。软件的生产过程不存在绝对正确的过程形式，可以肯定的是不同的软件开发项目应当采用不同的或者说是有针对性的软件开发过程，而真正合适的软件开发过程是在软件项目的开发完成才能明了的。因此项目开发之初只能根据项目的特点和开发经验进行选择，并在开发过程中不断的调整。

软件项目生存期模型的基本特征是：

- 1) 描述了开发的主要阶段。
- 2) 定义了每一个阶段要完成的主要过程和活动。
- 3) 规范了每一个阶段的输入和输出。

一定规模的软件企业，会有生存期模型库，模型库中有各种生存期模型的具体说明，项目经理可以根据项目的具体情况选择适合本项目的生存期模型。这里我们介绍在软件开发中可能用到的主要生存期模型：V 模型，瀑布模型，原型模型，增量模型，螺旋式模型，快速应用开发模型，渐近式阶段模型等。

选择具体的 IT 项目生命周期，需要按照项目的特点、项目管理的方式来确定。例如，软件开发项目的生命周期划分方式有多种，它们具有不同的特点，在实际运用时应该根据具体情况予以裁剪。在项目的开始阶段，成功完成项目的概率很低，风险和不确定性很高，随着项目的发展，完成的概率越来越高，直到最后完全明确。项目生命周期中一个阶段向另一个阶段过渡的过程都是一个风险过滤的阀门，因而，这些阶段的结束（或开始）就是项目控制节点，就是项目里程碑。

### 1.6.1 瀑布模型

瀑布模型（waterfall model）是一个经典的模型，也称为传统模型（conventional model），是一个理想化的生存期模型，见图 1-7。它要求项目所有的活动都严格按照顺序执行的，一个阶段的输出是下一阶段的输入。在很多的标准中都有明确定义瀑布模型，而且是软件工程经常涉及的模型。这个模

型没有反馈，一个阶段完成后，一般就不返回了。尽管实际的项目中要经常返回上一阶段。虽然瀑布模型是一个比较“老”的模型，甚至有些过时，但在一些小的项目中还是经常用到的。

瀑布模型有如下特点：

- 简单、易用、直观。
- 开发进程比较严格，一个进程顺着一个进程进行。
- 模型中没有反馈过程。
- 模型执行过程中需要严密控制。
- 允许基线和配置早期接受控制。
- 一个新的项目不适合瀑布模型，除非在项目的后期。
- 用户直到项目结束才能看到产品的质量，用户不是渐渐地熟悉系统。
- 不允许变更或者限制变更。

瀑布模型使用指南可以从三个方面说明：

- 1) 开发前，要进行概念开发和系统配置的开发，概念开发主要是确定系统级的需求，提交一份任务陈述。系统配置开发需要确定软件和硬件的情况。
- 2) 开发中，需进行需求过程、设计过程、实施过程。
- 3) 开发后，需进行安装过程、支持过程、维护过程、抛弃过程等。

瀑布模型比较适合下列情况的项目：在项目开始前，项目的需求已经被很好的理解，也很明确，

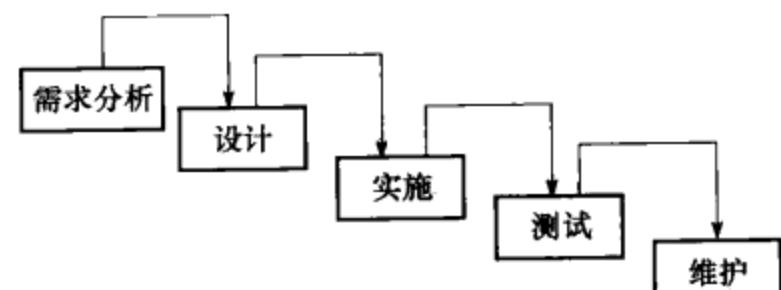


图 1-7 瀑布模型

而且项目经理很熟悉为实现这一模型所需要的过程，同时解决方案在项目开始前也很明确。很多的短期项目可以采用瀑布模型。

### 1.6.2 V 模型

V 模型是瀑布模型的一种变种，见图 1-8，同样需要一步一步进行，前一阶段任务完成之后才可以进行下一阶段的任务。这个模型强调测试的重要性，它将开发活动与测试活动紧密地联系在一起。每一步都将比前一阶段进行更加完善的测试。

实验证明，一个项目 50% 以上的时间花在测试上。一般，大家对测试存在一种误解，认为测试是开发周期的最后一个阶段。其实，早期的测试对提高产品的质量，缩短开发周期起着重要作用。V 模型也正好说明了测试的重要性，它是与开发并行的，这个模型体现了全过程的质量意识。

V 模型的特点如下：

- 简单易用，只要按照规定的步骤一步一步执行即可。
- V 模型强调测试过程与开发过程的对应性和并行性，例如单元测试对应详细设计，集成测试对应概要设计，系统测试对应需求分析。

使用 V 模型，要求开发的全过程是严格按照顺序进行的，一个阶段的输出是下一个阶段的输入。同时，注意图 1-8 中虚线对应过程的并行考虑，例如需求分析阶段，应该有确认测试的准备；概要设计阶段应该有集成测试的准备；详细设计阶段应有单元测试的准备等。

V 模型比较适合下列情况的项目：项目的需求在项目开始前很明确，解决方案在项目开始前也很明确，项目对系统的性能安全很严格。类似的项目如：航天飞机控制系统，公司的财务系统等。

### 1.6.3 原型模型

原型模型是在需求阶段快速构建一部分系统的生存期模型，见图 1-9。用户可以通过试用原型提出原型的优缺点，这些反馈意见可以作为进一步修改系统的依据。开发人员对开发的产品有时与客户不一致，因为开发人员更关注设计和编码实施，而客户更关注于需求。因此，如果开发人员快速构造一个原型将会很快与客户就需求达成一致。

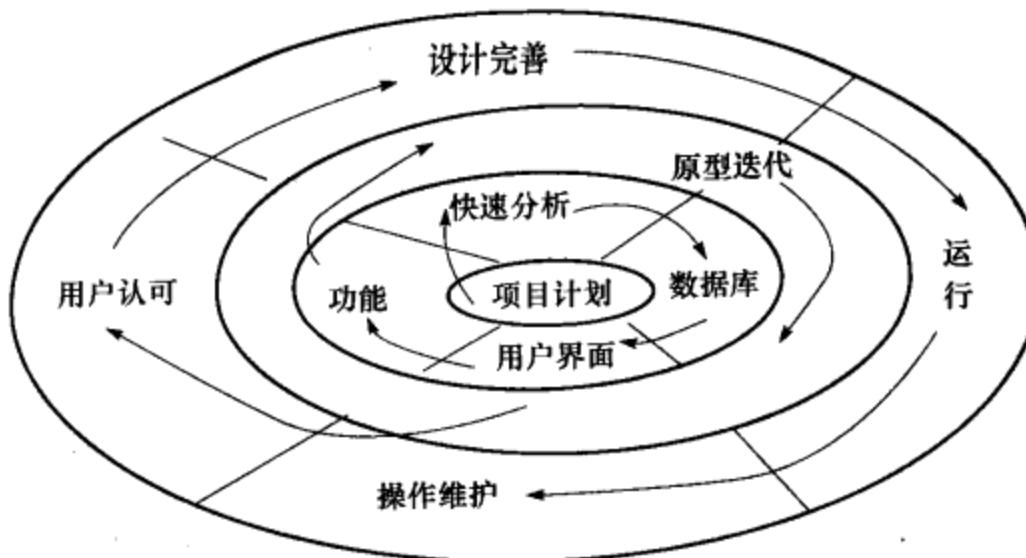


图 1-9 原型模型

原型模型有如下特点：

- 需求定义之前，需要快速构建一个原型系统。
- 用户根据快速构建的原型系统的优缺点，给开发人员提出反馈意见。
- 根据反馈意见修改软件需求规格，以便系统可以更正确地反映用户的需求。
- 可以减少项目的各种假设以及风险等。

使用原型模型的基本指南是：

- 用户和开发人员根据初始需求共同开发一个项目规划。
- 用户和开发人员利用快速分析技术共同定义需求和规格。
- 设计者构建一个原型系统。
- 设计者演示这个原型系统，用户来评估性能并标识问题。
- 用户和设计者一起来解决标识的问题，循环这个过程，直到用户满意为止。
- 详细设计可以根据这个原型进行。
- 原型可以用代码或者工具来实施。

当项目的需求在项目开始前不明确，或者需要减少项目的不确定性的时候，可以采用原型方法。类似的项目如：需要明确系统的界面，验证一些技术的可行性等。

#### 1.6.4 增量式模型

增量式模型（incremental life cycle model）是由瀑布模型演变而来的。该模型假设需求可以分段，成为一系列增量产品，每一增量可以分别的开发。首先构造系统的核心功能，然后逐步增加功能和完善性能的方法就是增量式模型。增量式生存期模型见图 1-10。

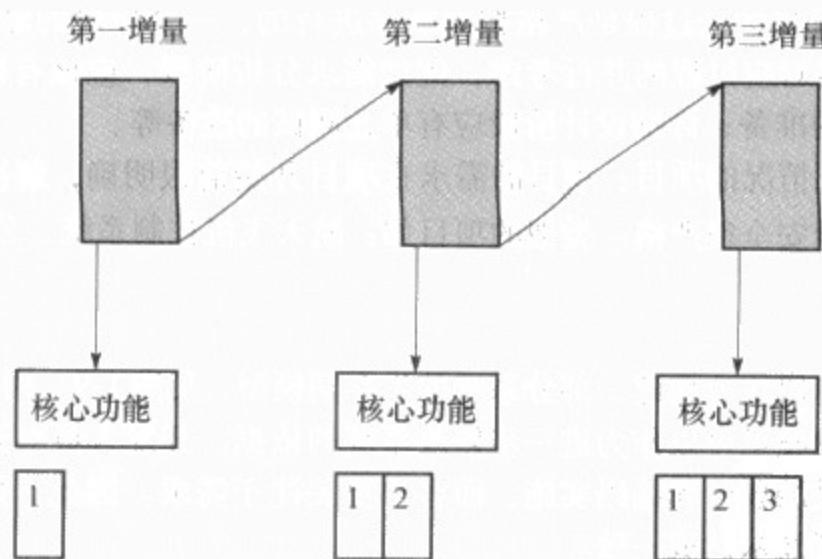


图 1-10 增量式模型

增量式模型的特点如下：

- 可以避免一次性投资太多带来的风险，将主要的功能或者风险大的功能首先实现，然后逐步完善，保证投入的有效性。
- 可以更快地开发出可以操作的系统。
- 可以减少开发过程中用户需求的变更。
- 一些增量可能需要重新开发（如果早期开发的需求不稳定或者不完整）。

使用增量式模型时，首先构建整个系统的核心部分，或者是具有高风险的部分功能，这部分功能对项目的成功起到重要作用。通过测试这些功能以决定它们是否是项目所需要的，这样可以排除后顾之忧，然后逐步地增加功能和性能，循序渐进，避免一步走的太快，增加功能的时候应该高效而且符合用户的需要。

增量式模型比较适合下列情况的项目：

- 项目开始时，明确了大部分的需求，但是需求可能会发生变化的项目。
- 对于市场和用户把握不是很准，需要逐步了解的项目。
- 对于有庞大和复杂功能的系统进行功能改进时需要一步一步实施的项目。

### 1.6.5 螺旋式模型

螺旋式模型（spiral model）是针对风险比较大的项目而设计的一种模型。设计这个模型的目的主要是克服瀑布模型的缺点，尤其是应对变化的灵活性上很有优势。它是通过一系列瀑布模型的不断循环来逐步规避风险的。螺旋式模型见图 1-11。每个循环步骤包括如下四个阶段：

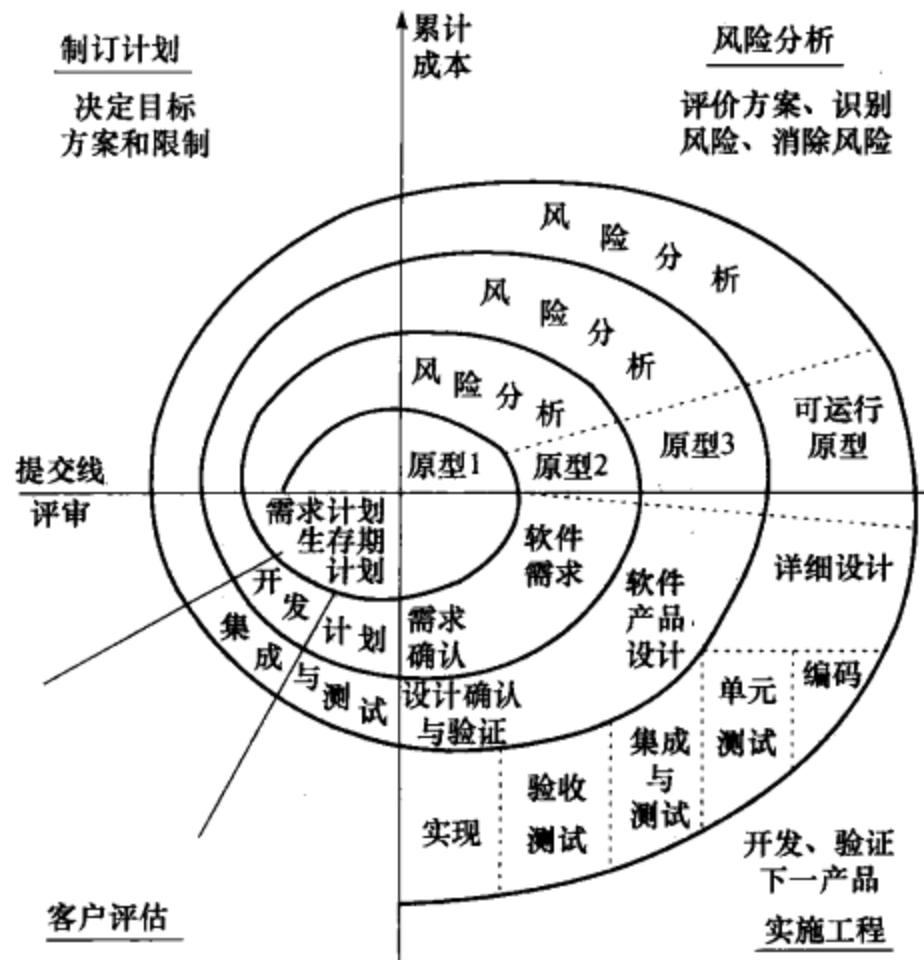


图 1-11 螺旋式模型

- 1) 制定计划——确定软件目标、需求和选定实施方案，弄清项目开发的限制条件，确定下步可选的方案。
- 2) 风险分析——评估所选方案，考虑如何识别和消除风险，进行原型开发。
- 3) 实施工程——实施软件开发、编码、测试等。
- 4) 客户评估——评价开发工作，提出修正建议，规划下一阶段的任务。

螺旋式模型提供了多个系统构造，为用户提供了几个可选的机会，因此需要精心的策划。

螺旋式模型的特点如下：

- 表现为瀑布模型的多次迭代。
- 可以将每个阶段进行更细的划分。
- 可以进行灵活设计。
- 通过风险管理进行驱动。
- 用户可以更早看到产品。
- 用户可以不断对产品进行评估。
- 用户可以与开发人员进行紧密的合作。
- 项目的投资不用一次性投入。

- 可以给开发人员更多的反馈信息。

螺旋式模型的使用指南如下：

- 采用最低成本开发对项目有用的一部分。
- 允许设计的变动。
- 选择比较小的步伐循序渐进。
- 需求规格处于可修改状态。
- 项目中有很多风险，注意风险的控制。

螺旋式模型适合的项目类型是：

- 项目中风险是主要的制约因素。
- 项目中的不确定因素和风险限制了项目的进度。
- 用户对自己的需求不是很明确。
- 需要对一些基本的概念进行验证。
- 可能发生一些重大的变更。
- 项目规模很大。
- 项目中采用了新技术。

### 1.6.6 渐近式阶段模型

项目规模越大对项目的管理人员要求越高，参与的人员越多，需要协调沟通的渠道越多，周期越长，开发人员也容易疲劳，将大项目拆分成几个小项目，可以降低对项目管理人员的要求，减少项目的管理风险，而且能够充分地将项目管理的权利下放，充分调动人员的积极性，目标会比较具体明确，易于取得阶段性的成果，使开发人员有成就感。项目采用阶段性版本发布，减少项目组成员的挫折感，提高大家的士气。

对于软件项目来讲，可以将大的项目划分成几个小项目来做，将周期长的项目划分成几个明确的阶段。“化繁为简，各个击破”是自古以来解决复杂问题的一个法宝。例如一个5年完成的项目可以分成5个阶段，每年提交一个版本，无形中的感觉是工作时间缩短了，工作量变小了，尽管实际中的工作量没有减少。开发过程中反复和阶段提交是比较合理的过程，渐进式阶段模型恰恰体现了这些特征。它也是近来比较流行的生存期模型。渐进式阶段模型见图1-12和图1-13。

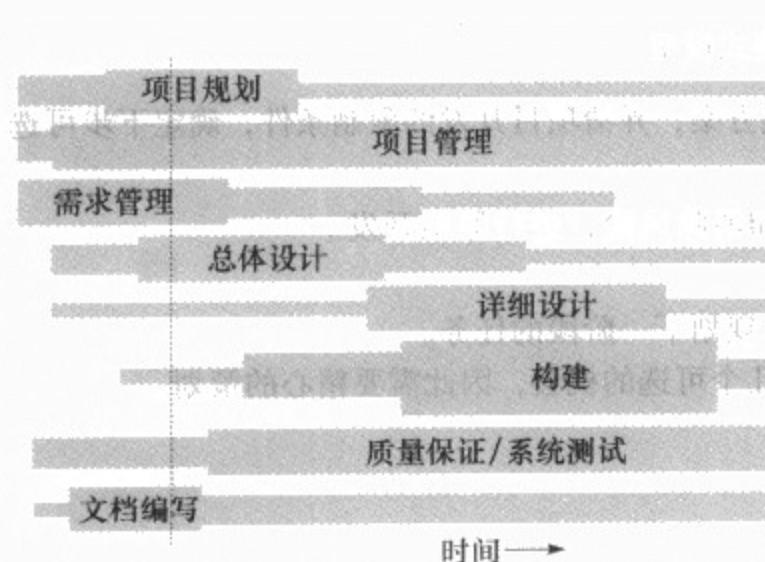


图 1-12 渐进式模型

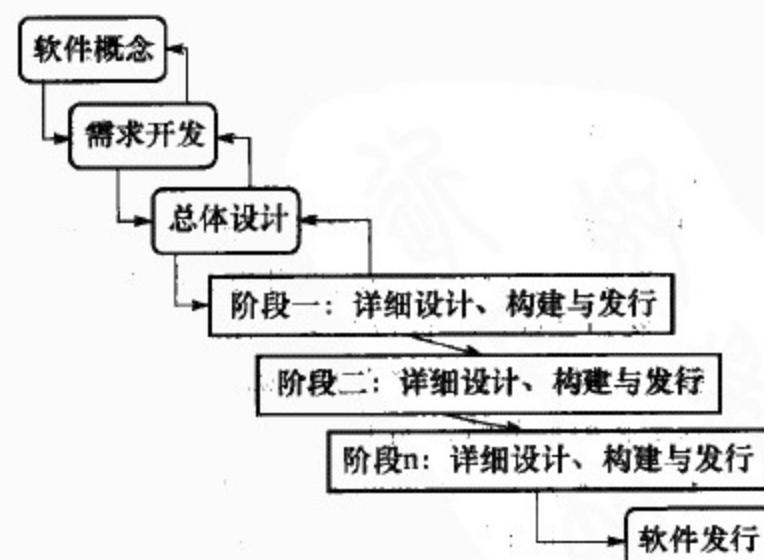


图 1-13 阶段式模型

渐进式阶段模型体现渐进式过程和阶段提交的模式，从图1-12可以看出“项目规划”、“项目管理”、“需求管理”、“总体设计”、“详细设计”、“构建”、“质量保证/测试”、“文档编写”等过程都是贯穿项目始终的，只是各个阶段的任务量不同而已，“项目规划”开始任务多，而后每个阶段的工作量逐渐变少了，“项目管理”是从始至终都有的，而“需求管理”开始任务多，然后逐步变少，“总体设计”也是在需求快结束的时候开始，然后逐步减少。著名的80/20的黄金规则可以应用到这里，在完成80%的需求开发后

就可以开始总体设计，完成80%的总体设计后，就要开始详细设计，以此类推。而且这个模型也强调将项目中开发的软件分阶段完成，每个阶段可以提交不同版本的产品，而不是一次性完成。这样项目经理可以提早得到明确的进度报告，而不是类似“完成80%”、“还剩20%”等模棱两可的报告。产品的动态结果比任何书面报告更能精确反应项目状况。

渐进式阶段模型具有如下的特点：

- 阶段式提交一个可运行的产品，而且每个阶段提交的产品是独立的系统，如图1-14。
- 关键的功能更早出现，可以提高开发人员和客户的信心。
- 通过阶段式产品提交，可以早期预警问题，避免后期发现问题的高成本。
- 通过阶段式提交可以运行的产品，来有力的证明项目的实际进展，减少项目进展报告的负担。
- 阶段性完成可以降低估计失误，因为通过阶段完成的评审，可以重新估算下一阶段的计划。
- 阶段性完成均衡了弹性与效率，提高开发人员的效率和士气。

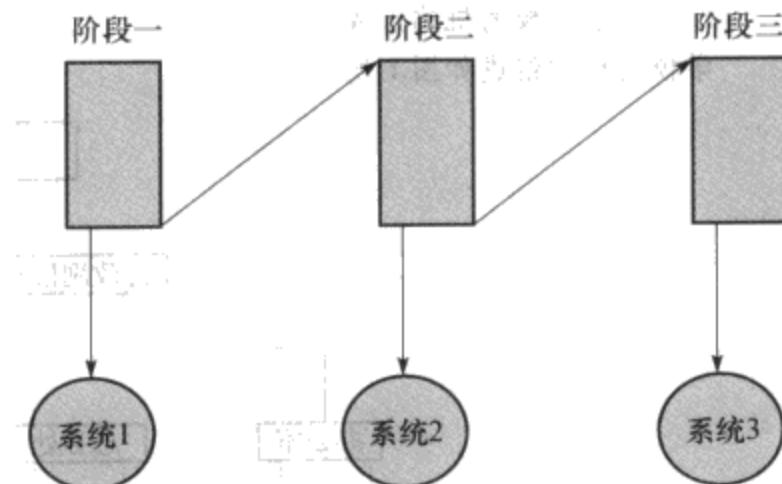


图1-14 渐进式系统

选择渐进式阶段模型的项目团队先进行软件概念分析，汇集、分析需求，再完成架构设计。这些工作通过积极的风险管理与精心规划，朝着消除风险的目标前进。在每个实施阶段，进行详细设计、实施、测试、修正等，而且每个阶段都建立可能推出的产品。

从本质上讲，渐进式阶段模型可以适合任何规模的项目，但是需要不断提交新的版本，因此渐进式阶段模型主要适合中型或大型项目，是目前软件开发中常采用的模型。采用这个模型可以随时看到项目的未来。

作者曾主管过的一个软件开发项目，该项目前期投入了5人做需求，时间达3个多月，进入开发阶段后，投入了15人，时间达10个月之久，陆续进行了3次封闭开发，在此过程中经历了需求的裁剪、开发人员的变更、技术路线的调整，项目组成员的压力极大，大家疲惫不堪，产品上市时间拖期达4个月。项目完工后总结下来的一个致命的教训就是应该将该项目拆成3个小的项目来做，进行阶段性版本化发布，以缓解市场上的压力，减少项目组成员的挫折感，提高大家的士气。

## 1.7 校务通系统案例分析

某教育部门（甲方）希望所管辖的学校有一个现代化的信息交流平台，即校务通系统，为此他们提出了《校务通系统》的需求，希望委托软件公司为其开发这样的软件项目。《校务通系统》是对学校教务和教学活动进行综合管理的平台系统，是一个学校和地区教育信息化的基础信息平台。目的是共享学校各种资源、提高学校的工作效率、规范学校的工作流程、便利校内外的交流。针对这个项目，甲方采取多方洽谈的招标方式。经过多方沟通和不懈的努力，北京XXX公司（乙方）获得了这个项目的开发权。双方经过多次的协商和讨论，最后签署项目开发合同。

### 1.7.1 甲方招标需求

由于本项目的甲（卖）方采取了多方洽谈的招标方式，所以没有明确的招标书，只编写了工作任务说明（SOW）与乙（买）方谈判。SOW如下：

#### 校务通管理平台信息系统业务需求

校务通管理平台信息系统是对学校教务和教学活动进行综合管理的平台系统，是一个学校和地区教育信息化的基础信息平台。满足学校管理层、教师、学生、家长等日常工作、学习、管理、咨询等工作。

的是共享学校各种资源、提高学校的工作效率、规范学校的工作流程、便利校内外的交流。

### 一、整体要求：

1. 系统要求提供教师工作平台和学生工作平台。
2. 系统要求有严格的权限管理，权限要在数据方面和功能方面都要体现。
3. 系统要求有可扩充性，可以在现有系统的基础上，通过前台就可加挂其他功能模块。

### 二、一般学校的机构组成

学校的机构组成如图 1 所示。

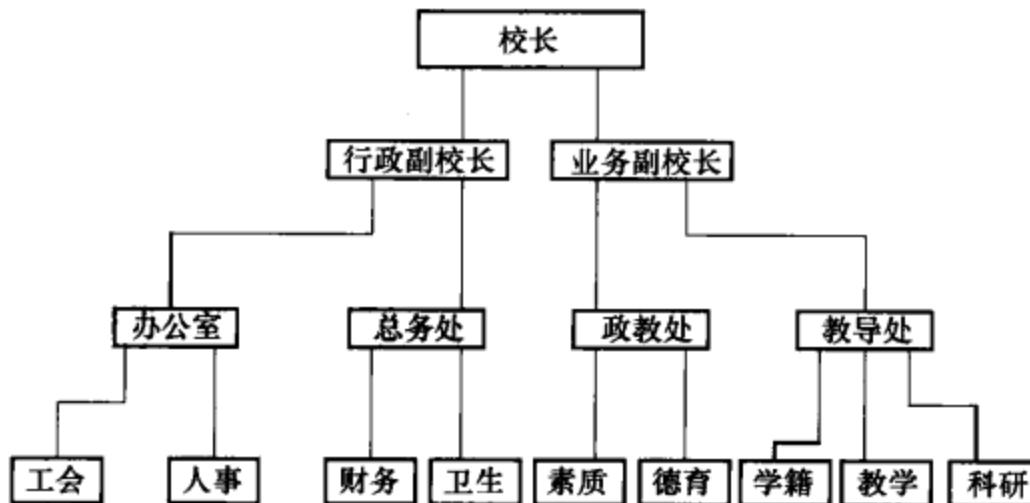


图 1 一般学校的机构组成

### 说明：

1) 可能每个学校机构不尽相同，但基本框架相似。这里需要指出的是关于学科教研室的设置，有几种情况：

- 每个年级设立学科教研室，如：初一数学教研室、初二数学教研室。
- 每个级部设立学科教研室，如：初中数学教研室、高中数学教研室。
- 每个学校设立学科教研室，如：××学校数学教研室、××学校数学教研室。

所以，对于教研室最好不要在机构中体现，在教师基本情况中体现即可。

2) 对于学校组织机构和人员的设置应遵循以下原则：

#### 组织机构设置：

××学校为一级，各处（室）、单位、级部为二级，各年级为三级，各班级为四级。

#### 人员设置：

各人员均设置在相应的处（室）、单位、级部和年级，即人员的设置最低到年级。

#### 3) 机构的日常业务

**办公室：**各类通知的上传下达、工作安排、日程管理、教师档案管理

**总务处：**固定资产管理、教学仪器的使用管理、图书的借阅管理、卫生值日的安排和检查

**政教处：**学生德育教育的管理和评定、学生家长和学校的沟通、学生大型活动的安排

**教务（导）处：**学生学籍的管理、教师教学的管理

**班主任：**班级学生学籍的管理、学生日常管理、学生成绩评价，学生学期评定、学生毕业鉴定。

**任课教师：**学生的教学（备课、考试），学生考试成绩评价分析。

### 三、系统功能描述

#### (一) 通用功能

对于每个教师，登录系统后，都应提供如下功能

- 1) 电子课表：系统根据学校总排课的情况和该教师的任课情况自动生成电子课表备该教师查阅。
- 2) 会议通知和公告：系统根据该教师的权限，自动列出该教师需要查阅的会议通知和公告，同时若具备起草和发布通知和公告，则系统提供相应功能。发送通知和公告应可自由设定相应的权限组。如全体学生、全体老师、一年级全体老师等。

- 3) 日程安排：该日程安排应可分级设定，教师登录后可看到与自己有关的日程，同时能对自己的日程进行安排，日程安排同时需要设置自动提醒功能。
- 4) 个人日记：系统可为每个用户设置一个用于个人记事的功能
- 5) 通讯录：系统自动从教师基本信息和学生基本信息中抽取通讯记录，形成公共通讯录用于用户查询使用，同时应给用户提供一个个人通讯录，该通讯录应能够录入、修改、删除、检索。
- 6) 教师答疑：系统自动抽取在学生平台提出的需该教师回答的问题，由教师进行解答，并记录相应状态。
- 7) 家庭作业：教师课利用此功能对学生进行作业布置和批改。

## (二) 学校日常业务管理功能

### 1. 招生管理

本功能完成各学校从招生到入学的全部过程。其业务流程图如图2所示。

说明：

1) 对于招生工作，首先是报名，系统需要提供报名功能，有的学校还要组织招生考试，但某些招生不存在入学考试，如小学的招生和某些中学的招生等。如不需考试则根据报名审查情况录入新生基本信息（参照所提供的“招生录入”），录入信息包括姓名、性别、考籍号、总分、考生来源、考生类型。下划线加粗部分为必录项。

2) 符合入学条件的学生全部录入或根据考试情况转入完毕后，首先要根据性别和分数进行分班，分班原则为：每班男生、女生比例要基本一致，各班各分数段的人数要基本一致。自动分班后，一定要提供手工调整的功能。

3) 分班结束后，转入正常教学工作前一定要保证，各班的升学工作已经结束。

### 4) 统计查询

可按入学总分统计查询，可按男女生查询，按学生来源统计。对于总分可按任意分数段统计。如每10分一个分数段，或每1分一个分数段。

### 2. 学生日常管理

学生日常管理应包括学生档案管理，学生考勤管理，学生奖惩管理、学生变动管理

#### 1) 学生档案管理

本模块应完成学生基本档案信息的录入过程

具体内容请参照所提供的资料完善，需要加入照片。很多项目应有下拉框选择，日期时间加一个统一的控件。

另：学籍卡片附后。

#### 2) 学生考勤管理

能完成正常考勤工作即可。

#### 3) 学生奖惩管理

本功能将合并到后期的学生素质评价中。

#### 4) 学生变动管理

应包括升学、降级、留级、转学、转班、休学、复学、辍学、退学、开除、死亡等。对于升学要考虑用户可操作性，不能对每个学生逐个进行升学处理。

### 3. 教务管理

教务管理主要完成以下功能：教师日常管理、年级、班级设置、学科设置、年班级课程设计、排课表、考试、评价

- 1) 关于教师档案，比照所提供的资料设计，现在也没有一个成形的东西。关于论文、奖惩、培训都放在

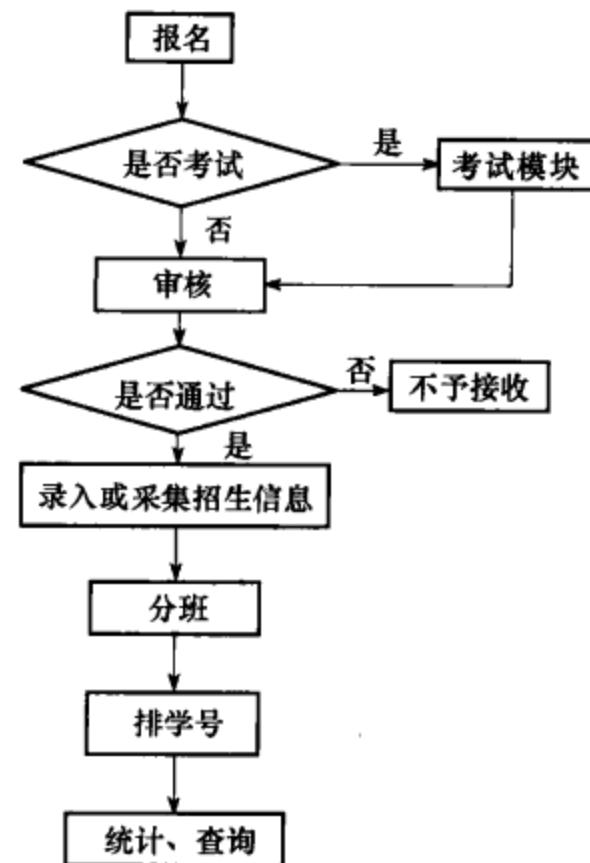


图2 学校招生管理功能的业务流程图

后期教师评价统一处理。

2) 对于年班级设置、学科设置、年班级课程设计基本按照我们讨论结果。需指出的是：年、班级设置中班级设置中增加一个班级级别字段，用来后期对班级的评价，有些学校每学期会对每个班级打分，评出星级班级。学科设置，由于前面讲到的机构问题，最好不要和学科教研室联系。

3) 对于排课表的设计要遵照以下原则：

- 确定每个班级的课程类型、每门课的任课教师、每门课的周课时数和每周上课的天数。
- 确定学校每天课时数。
- 确定每门课在节次上的限制。
- 确定每门课的场地限制。
- 每个任课老师在兼顾前面的情况下，每天上课时间要交错开。
- 其他因素，如哺乳期教师不能安排在开头或者结尾几节课等。

4) 考试管理

学校考试管理功能的业务流程图如图 3 所示。

对于考试是各个学校比较重视的一个方面，因为考试成绩在现阶段很多方面起着主导作用，现将考试需求明确如下：

- 对于考试类型用户应可以自行维护，考试类型维护中应增加权重（即所占比例）一项。
- 考试科目和考试时间可根据实际情况设置即可。
- 考场安排应遵循以下原则：
  - 对不需要安排考场的考试如单元考试等可跳过。
  - 对需要安排考场的考试首先要确定考生数量、考生来源（班级或学校）、考场数量，每个考场的考生数量，课桌排列方式。
  - 根据以上情况按照相同来源的考生前后左右不能相邻的原则分考场和考号。
- 考试成绩录入，应能够按照每个班级、学科录入，对于成绩单的生成，也可考虑数据导入。
- 对于考试成绩应能够按照班级排序查询或者按照参加此类考试的全部学生排序查询。可以按照优秀率、及格率、平均分、标准分统计。可以按照分数段统计，分数段可自由设定。

5) 评价

本模块完成对教师、班级、学生的评价。

教师教学评价：

对于教师评价，暂按如下操作：

- 评价工作通常每学期一次，期末产生结果。
- 设定一指标体系框架，指标内容可由用户自行维护，指标体系要分级如：中学 - 师德素质 - 爱岗敬业。
- 指标应包括，具体指标内容，权重，分数、备注。
- 对于教师的论文情况、奖惩、培训情况都可作为指标维护。
- 对每个教师所涉及的指标录入相应的分数和备注。
- 通过指标的组合形成一个评价公式（评价方案）。
- 根据评价方案产生评价结果。

班级星级评价

班级评价比照教师评价操作，对于评价应该有一个分数转换对照表，如 90 分以上为优秀、A 级、五星级等。

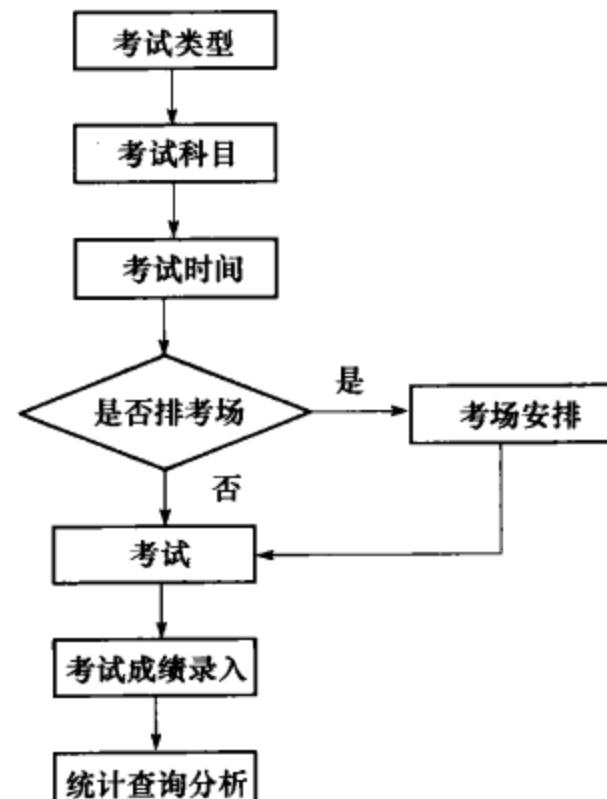


图 3 学校考试管理功能的业务流程图

**学生成绩评价:**

学生成绩评价比照以上操作。但需要考虑,如何把各种考试成绩加到评价系统指标体系中。

**4. 教师备课系统**

此功能提供给每一个授课老师一个计算机备课的功能,包括备课素材的准备、组织以及备课笔记的生成和存档。

**5. 资源库系统**

应提供一个标准的资源库解决方案。

**6. 网上考试功能****7. 聊天室**

挂一个功能比较完善的成品即可。

**8. 论坛**

找一个比较有特色的成品即可。

**1.7.2 乙方建议书**

乙方为了争取项目,编写了项目建议书(即投标书)。由于本项目建议书与需求规格说明书(见第2章的案例分析)有部分重叠,暂略。

**1.7.3 项目合同**

双方经过多次的协商和讨论,最后签署项目开发合同。合同文本如下:

合同登记编号:

--	--	--	--	--	--	--	--	--	--	--	--

**技术开发合同**

项目名称: 校务通管理系统

委托人(甲方): XXXXX省教育委员会

研究开发人(乙方): 北京科力拓技术发展有限公司

签订地点: 北京市

签订日期: 2003年4月10日

有效期限: 2003年4月10日至2003年12月16日

北京技术市场管理办公室

根据《中华人民共和国合同法》的规定,合同双方就校务通管理系统开发项目的技术开发(该项目属于\_\_\_\_/\_\_\_\_计划),经协商一致,签定本合同。

**一、标的物的内容、范围及要求:**

根据甲方的要求,乙方完成校务通软件系统的研制开发。

1. 根据甲方要求进行系统方案设计,要求建立B/S结构的,基于的Sqlserver数据库、NT服务器和J2EE技术的三层架构体系的综合服务软件系统。

2. 配合甲方,在与整体系统相融合的基础上,建立系统运行的软硬件环境。

3. 具体需求见SOW

**二、应达到的技术指标和参数**

1. 系统应满足并行登录,并行查询的速度要求。其中主要内容包括(1)保证100人以上可以同时登录系统(2)所有查询速度应在10秒以内(3)保证数据的每周备份(4)工作日期间不能当机(5)出现问题应在10分钟内恢复。

2. 系统的主要功能是应满足双方认可的需求规格,不可以随意改动。

### 三、研究开发计划

1. 第一阶段：乙方在合同签订后 7 个工作日内，完成合同内容的系统设计方案。
2. 第二阶段：完成第一阶段的系统设计方案之后，乙方于 50 个工作日内完成系统基本功能的开发。
3. 第三阶段：完成第一和第二阶段的任务之后，由甲方配合乙方于 3 个工作日内完成系统在 XXX 信息中心的调试、集成。

### 四、研究开发经费、报酬及其支付或结算方式

1. 研究开发经费是指完成本项目研究开发工作所需的成本。报酬指本项目开发成果的使用费和研究开发人员的科研补贴。

2. 本项目研究开发经费和报酬（人民币大写）：XXX 万元整。

3. 支付方式：分期支付。

本合同签订之日起生效，甲方在五个个工作日内应付乙方合同总金额的 50%，计人民币 XXX.00 元（人民币大写 XXX 元整），验收后甲方在五个个工作日内付清全部合同余款，计人民币 XXX.00（人民币大写 XXX 元整）。

五、利用研究开发经费购置的设备、器材、资料的财产权属：/。

### 六、履行的期限、地点和方式。

本合同自 2003 年 8 月 16 日至 2003 年 11 月 16 日在北京履行。

### 本合同的履行方式：

#### 甲方责任

1. 甲方全力协助乙方完成合同内容。
2. 合同期内甲方为乙方提供专业性接口技术支持。

#### 乙方责任：

1. 乙方按甲方要求完成合同内容。
2. 乙方愿提供在实现功能的前提下，进一步予以完善。
3. 乙方在合同商定的时间内保证系统正常运行。
4. 乙方在项目验收后提供一年免费维护。
5. 未经甲方同意，乙方不得向第三方提供本系统中涉及专业的技术内容和所有的系统数据。

### 七、技术情报和资料的保密。

本合同中的相关专业技术内容和所有的系统数据，归甲方所有，未经甲方同意乙方不得提供给第三方。

### 八、技术协作的内容：

见系统设计方案。

### 九、技术成果的归属和分享

专利申请权：/。

技术秘密的使用权、转让权：/。

### 十、验收的标准和方式：

研究开发所完成的技术成果，达到了本合同第二条所列技术指标，按国家标准，采用一定的方式验收，由甲方出具技术项目验收证明。

### 十一、风险的承担

在履行本合同的过程中，确因在现有水平和条件下难以克服的技术困难，导致研究开发部分或全部失败所造成的损失，风险责任由甲方承担 50%，乙方承担 50%。

本项目风险责任确认的方式：双方协商。

### 十二、违约金和损失赔偿额的计算：

除不可抗力因素外（指发生战争、地震、洪水、飓风或其他人力不能控制的不可抗力事件），甲乙双方须遵守合同承诺，否则视为违约并承担违约责任：

1. 如果乙方不能按期完成软件开发工作并交给甲方使用，乙方应向甲方支付延期违约金。每延迟一周，乙方向甲方支付合同总额 0.5% 的违约金，不满一周按一周计算，但违约金总额不得超过合同总额的 5%；
2. 如果甲方不能按期向甲方支付合同款项，甲方应向乙方支付延期违约金。每延迟一周，甲方向乙方

支付合同总额 0.5% 的违约金，不满一周按一周计算，但违约金总额不得超过合同总额的 5%。

### 十三、解决合同纠纷的方式：

在履行本合同的过程中发生争议，双方当事人和解或调解不成，可采取仲裁或按司法程序解决。

1. 双方同意由北京市仲裁委员会仲裁。

2. 双方约定向北京市人民法院起诉。

### 十四、名词和术语解释

如有，见合同附件。

### 十五、其他

1. 本合同一式 6 份，具有同等法律效力。其中正式两份，甲乙双方各执一份；副本 4 份，交由乙方。

2. 本合同未尽事宜，经双方协商一致，可在合同中增加补充条款，补充条款是合同的组成部分。

## 1.7.4 乙方项目授权书

签署合同之后，乙方授权项目经理正式管理这个项目，即授权项目。为此编写项目章程，这个项目的项目章程见表 1-3。

表 1-3 校务通项目的项目章程

项目名称	校务通系统	项目标识	QTD-SCHOOL		
下达人	项目委员会	下达时间	2003 年 4 月 10 日		
项目经理	韩万江	项目计划提交时限	2003 年 4 月 14 日		
送达人	姜岳尊，孙泉，郭天奇，岳好，杨炎泰				
项目目标	1. 为 xxx 提供基于 B/S 结构的校务管理系统； 2. 为 xxx 提供多平台的交流。				
项目范围	项目性质	公司外部项目，属软件开发类			
	项目组成	见项目输入			
	项目要求	见项目输入			
	项目范围 特殊说明	无			
项目输入	1. 《校务通管理系统实施方案建议书》 2. 合同及其附件				
项目用户	XXXX 教育委员会				
与其他项目关系	无				
项目限制	完成时间	预计完成时间为：2003 年 6 月 20 日			
	资金	见项目输入 1 第 6 章			
	资源	依据批准的项目计划			
	实现限制	B/S 结构，开发平台为：Windows NT、IIS Server、Sql server、J2EE			

## 1.7.5 生存期模型

针对本项目的开发特点，参考企业的生存期模型说明和软件过程体系，决定采用增量式模型如图 1-15 所示，理由如下：

1) 校务通系统的全部功能分成通用功能和日常业务管理功能两大类，因此可以先基于通用功能作出一个最小的使用版本，再逐步添加其余的功能。这样一来，用户可以先试用最小版本的同时，提出更多明确的需求，这有助于下一阶段的开发，大大减小了开发的风险。

2) 在校务通系统需求中，要求系统有可扩充性。若使用增量模型，可以保证系统的可扩充性。用户明

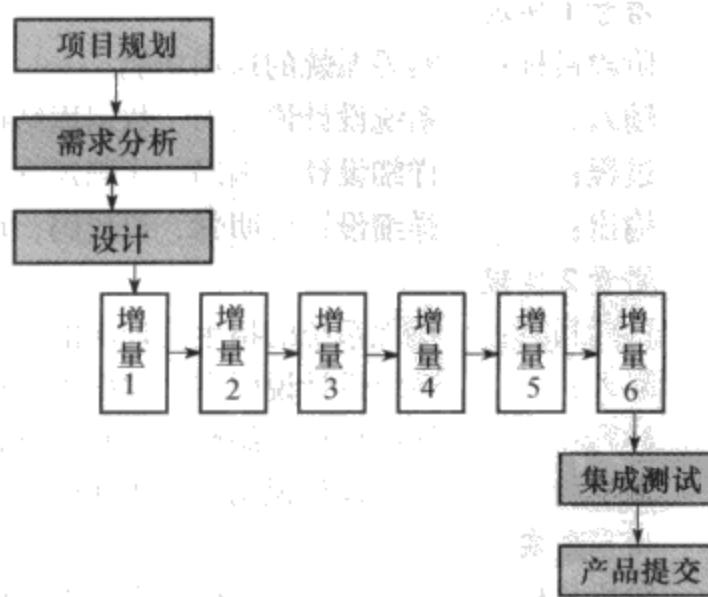


图 1-15 项目生存期模型

确了需求的大部分，但也存在不很详尽的地方。如：“关于教师档案，比照所提供资料设计，现在也没有一个成形的东西”，资源库系统只提到“应提供一个标准的资源库解决方案”。这样只有等到一个可用的产品出来，通过客户使用，然后进行评估，评估结果作为下一个增量的开发计划，下一个增量发布一些新增的功能和特性，直至产生最终完善的产品。

3) “系统要求有可扩充性，可以在现有系统的基础上，通过前台就可加挂其他功能模块”——也说明用户可能会增加新的需求。

4) 对一个管理方式已经比较成熟的学校，要完全舍弃原有的管理方式，用校务通系统替代全部管理，这是不实际的。所以，可以从最基础的做起，逐步扩充其应用，所以选用增量模型来开发校务通系统。

5) 本项目具备增量式模型的其他特点：

- 项目复杂程度为中等；
- 预计开发软件的成本为中等；
- 产品和文档的再使用率会很高；
- 项目风险较低。

生存期中的各阶段定义如下：

#### 项目规划阶段

阶段目标： 根据合同和初步的需求分析确定项目的规模、时间计划和资源需求。

输入： 合同文本、SOW

过程： 项目规划，计划确认

输出： 项目计划

#### 需求分析阶段

阶段目标： 确定客户的需求

输入： 项目计划，SOW

过程： 需求获取，需求分析，需求控制

输出： 原型系统，需求规格

#### 设计阶段

阶段目标： 总体系统结构设计

输入： 原型系统，需求规格

过程： 总体设计

输出： 系统设计说明书、数据库结构定义

#### 增量1实现

阶段目标： 实现系统的通用功能

输入： 系统设计说明书、数据库结构定义

过程： 详细设计，编码，代码走查，代码评审，单元测试

输出： 详细设计说明书，源代码，可运行版本-1

#### 增量2实现

阶段目标： 实现系统的招生管理功能

输入： 系统设计说明书、数据库结构定义

过程： 详细设计，编码，代码走查，代码评审，单元测试

输出： 详细设计说明书，源代码，可运行版本-2

#### 增量3实现

阶段目标： 实现系统的学生成绩管理功能

输入： 系统设计说明书、数据库结构定义

过程： 详细设计，编码，代码走查，代码评审，单元测试

输出： 详细设计说明书，源代码，可运行版本 -3

#### 增量 4 实现

阶段目标： 实现系统的教务管理功能

输入： 系统设计说明书、数据库结构定义

过程： 详细设计，编码，代码走查，代码评审，单元测试

输出： 详细设计说明书，源代码，可运行版本 -4

#### 增量 5 实现

阶段目标： 实现系统的教师辅助功能

输入： 系统设计说明书、数据库结构定义

过程： 详细设计，编码，代码走查，代码评审，单元测试

输出： 详细设计说明书，源代码，可运行版本 -5

#### 增量 6 实现

阶段目标： 实现系统的聊天室/论坛功能

输入： 系统设计说明书、数据库结构定义

过程： 详细设计，编码，代码走查，代码评审，单元测试

输出： 详细设计说明书，源代码，可运行版本 -6

#### 集成测试

阶段目标： 通过集成环境下的软件测试

输入： 测试计划、测试案例

过程： 集成测试，系统测试

输出： 系统软件包，测试报告，产品说明书

#### 产品提交

阶段目标： 产品可投入使用

输入： 系统软件包

过程： 产品提交

输出： 验收报告

**注：**生存期模型中的过程定义可以参照企业的质量保证体系并结合项目的具体特点来决定，由于公司的流程已覆盖到了项目开发、管理的所有方面，包括从最开始的合同到最后软件的产品提交，都有相应的过程规定，基本上已形成一种工业化的软件开发，所以，为形成一个良好的软件开发环境奠定了基础。

## 1.8 小结

本章介绍了项目初始阶段的过程以及提交的文档，项目可以是合同项目，也可以企业内部项目。项目立项后便进入项目的初始阶段。这个阶段可能会产生的主要输出是：项目章程、项目建议书、项目的初始范围说明、生存期模型等。

## 1.9 练习题

### 一、名词解释

1. 项目章程

2. 项目建议书

### 二、选择题

1. 项目建议书是哪个阶段开发的文档？（ ）

- A. 项目执行阶段      B. 项目结尾阶段      C. 项目初始阶段      D. 项目计划阶段

2. 项目章程（ ）。  
A. 明确了项目经理                           B. 确定了项目的质量标准  
C. 明确了团队的纪律                       D. 定义了项目需求
3. 对于风险比较大的项目，最好选择（ ）生存期模型。  
A. 瀑布模型                                  B. 原型                                      C. 螺旋模型                                  D. V 模型
4. 开发项目建议书的目的是为了（ ）。  
A. 验收    B. 竞标或者签署合同                      C. 编写计划                                  D. 跟踪控制项目

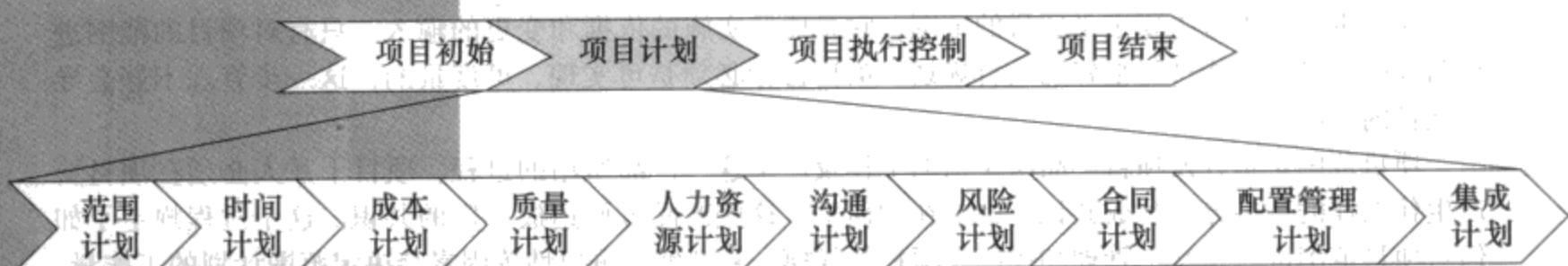
### 三、判断题

1. 项目经理是一个综合的角色。（ ）
2. 软件项目都是需要签署合同的。（ ）
3. 增量式模型可以避免一次性投资太多带来的风险。（ ）
4. 项目章程类似一个项目授权书。（ ）
5. 在甲乙合同项目中，提出需求的一方是乙方。（ ）
6. 如果是内部项目，在项目初始阶段可以不提交招标书。（ ）

## 第二篇

# 项目计划

这篇进入本书路线图第二阶段即项目计划阶段（如下图所示），这个阶段是很重要的阶段，项目计划是项目成败的关键。这篇中，我们将分为10章，分别讲述范围计划、时间计划、成本计划、质量计划、人力资源计划、沟通计划、风险计划、合同计划、配置管理计划以及集成计划等。



# 第2章

## ■ 范围计划

项目计划活动的第一项计划活动就是估算：需要多长时间、需要多少工作量、以及需要多少人员。此外，我们还必须估算所需要的资源（硬件及软件）和可能涉及到的风险。

项目管理过程中最重要也是最困难的方面之一是确定项目的范围，项目成功要素中很多是与范围相关的。范围计划是项目计划中很重要的一部分，是项目管理核心三角形的一边。这章开始项目计划的第一个计划，进入本书路线图的范围计划，见图 2-1。

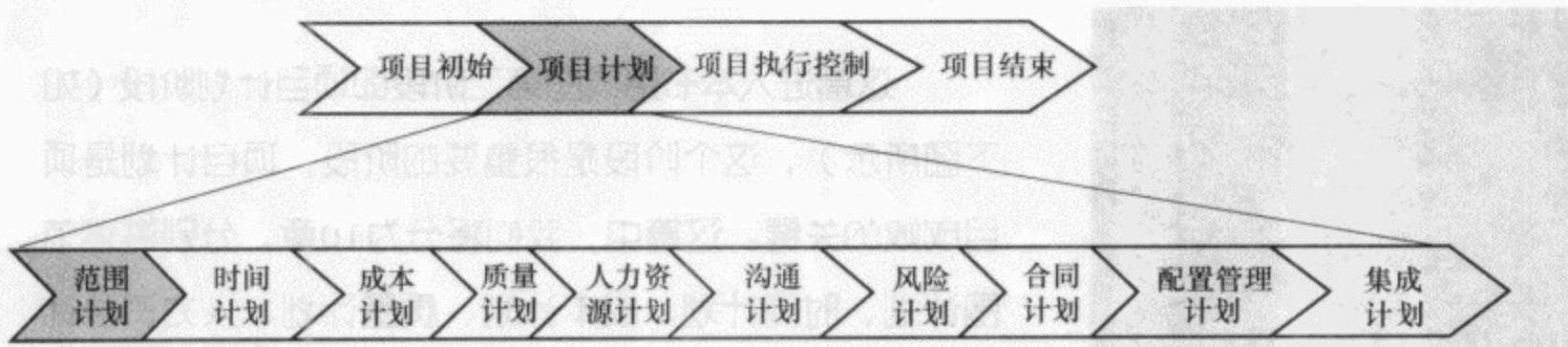


图 2-1 路线图——项目计划 - 范围计划

项目范围管理是指对项目包括什么与不包括什么的定义与控制过程。这个过程用于确保项目组和项目干系人对作为项目结果的项目产品以及生产这些产品所用到的过程有一个共同的理解。确定项目范围是制订软件开发计划的根据，它包括对功能、性能、接口和可靠性的确定。通过项目范围管理，明确项目管理的目标与边界。

在范围管理中首先要定义项目的范围，它是项目实施的依据和变更的输入，只有对项目的范围进行明确的定义，才能进行很好的项目规划。项目目标必须是可实现、可度量的，这一步管理不好会导致项目的最终失败。

项目范围是指开发项目产品所包括的工作及产生这些产品所用的过程。项目干系人必须在项目要产生什么样的产品方面达成共识，也要在如何生产这些产品方面达成一定的共识。这个过程规定了如何对项目范围进行定义、管理和控制。项目的规模、复杂度、重要性等因素会决定范围计划的工作量，不同的项目，范围计划的情况可以不同，对于大型的项目进行项目规划可以花费一定的时间。而软件项目的范围首先从项目的需求开始。

### 2.1 关于软件需求

启动软件项目的原因是软件需求的存在。无论采用何种软件生存期模型，软件需求是软件开发过程的基础。需求是一个软件项目的开端，也是项目建设的基石。有资料表明，软件项目中 40%~60% 的问题都是在需求分析阶段埋下的隐患。软件开发中返工开销占开发总费用的 40%，而其中 70%~80% 的返工是由需求方面的错误所导致的。在以往失败的软件项目中，80% 是由于需求分析的不明确而造成的。因此一个项目成功的关键因素之一就是对需求分析的把握程度。而项目的整体风险往往表现在需求分析不明确、业务流程不合理，所以，需求管理是项目管理的重要一环。

软件需求是指用户对软件的功能和性能的要求，就是用户希望软件能做什么事情，完成什么样的功能，达到什么样的性能。软件人员要准确理解用户的要求，进行细致的调查分析，将用户非形式的需求陈述转化为完整的需求定义，再由需求定义转化到相应形式的需求规格说明的过程。对于软件项目的需求，首先要理解用户的要求，要澄清模糊的需求，与用户达成共识。

软件需求包括三个不同的层次：业务需求（business requirement）、用户需求（user requirement）、功能需求（functional requirement）。最后确定软件需求规格（software requirement specification, SRS），它们的关系如图 2-2 所示。

业务需求反映了组织机构或客户对系统、产品高层次的目标要求，由管理人员或市场分析人员确定，它们在项目视图与范围文档中予以说明。

用户需求描述了用户通过使用本软件产品必须要完成的任务，一般是用户协助提供。用户需求可以在用例（use case）或场景（scenario）说明中予以说明。

功能需求定义了开发人员必须实现的软件功能，使得用户通过使用此软件能完成他们的任务，从而满足了业务需求。对一个复杂产品来说，功能需求也许只是系统需求的一个子集。

软件需求规格充分描述了软件系统应具有的外部行为，它描述了系统展现给用户的行为和执行的操作等。它包括产品必须遵从的标准、规范和合约；外部界面的具体细节；非功能性需求（例如性能要求等）；设计或实现的约束条件及质量属性。所谓约束是指对开发人员在软件产品设计和构造上的限制。质量属性是通过多种角度对产品的特点进行描述，从而反映产品功能。多角度描述产品对用户和开发人员都极为重要。

软件需求规格说明在开发、测试、质量保证、项目管理以及相关项目功能中都起了重要的作用。

用户需求必须与业务需求一致。用户需求使需求分析者能从中总结出功能需求，以满足用户对产品的期望从而完成其任务，而开发人员则根据软件需求规格来设计软件，以实现必要的功能。

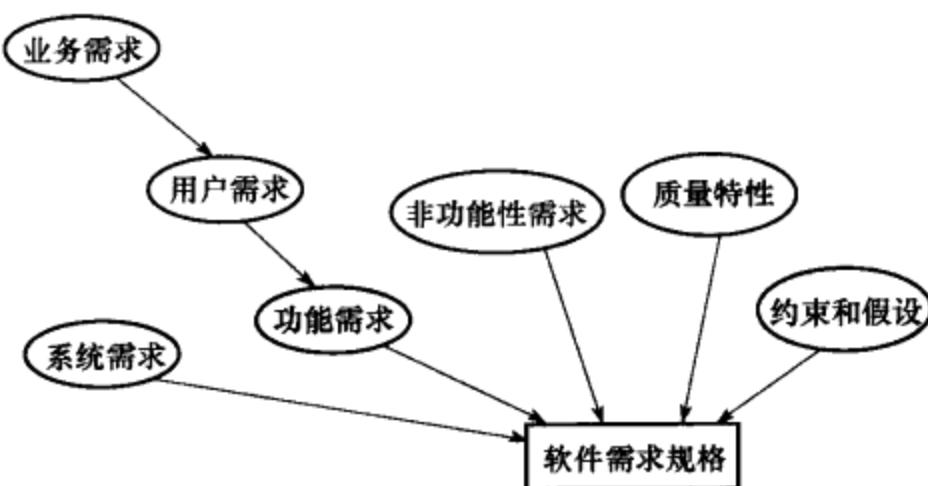


图 2-2 软件需求的层次

## 2.2 需求管理过程

需求管理过程是保证软件需求以一种形式描述一个产品应该具有的功能、性能等。对于软件的设计、实施以及产品验证，需求是它们的基本信息源，由此可见，需求中的错误对项目的成本和进度具有负面影响。需求上出现问题很少是源于需求开发技术，而更多的是软件人员对需求理解上的错误和忽略，源于需求工作的复杂性、细腻性以及任务的繁多。为了获得一个完整的需求，必须整合来自不同的信息源，因此开发需求是一个复杂的、细致的过程。需求过程的一个重要活动是系统化需求，并建立相应的文档。

有效的需求管理能获得多方面的好处，最大的好处是在开发后期和整个维护阶段的返工的工作量可以大大减少。Boehm(1981)发现要改正在产品付诸应用后所发现的一个需求方面的缺陷比在需求阶段改正这个缺陷要多付出 68 倍的成本。近来很多研究表明这种缺陷导致成本放大因子可以高达 200 倍。

20 世纪 80 年代中期，形成了软件工程的子领域——需求工程（RE）。需求工程是一个不断反复的需求定义、文档记录、需求演进的过程，并最终在验证的基础上冻结需求。需求工程是指应用已证实有效的技术、方法进行需求分析，确定客户需求，帮助分析人员理解问题并定义目标系统的所有外部特征的一门学科。它通过合适的工具和记号系统地描述待开发系统及其行为特征和相关约束，形成需求文档，并对用户不断变化的需求演进给予支持。软件需求工程是一门分析并记录软件需求的学科，它把系统需求分解成一些主要的子系统和任务，把这些子系统或任务分配给软件，并通过一系列重复的分析、设计、比较研究、原型开发过程把这些系统需求转换成软件的需求描述和一些性能参数。

软件需求工程的管理分为需求开发和需求管理，如图 2-3 所示。需求开发是对需求进行调查、收

集、分析、评价、定义等所有活动，主要包括需求获取、需求分析、需求规格编写和需求验证等过程。需求管理是对需求进行一些维护活动，保证在客户和开发方之间能够建立和保持对需求的共同理解，同时维护需求与后续工作成果的一致性，并控制需求的变更，即需求变更管理过程，它主要任务是需求评审、需求跟踪、需求变更控制。

### 2.2.1 需求获取

软件需求是整个软件开发项目的最关键的一个输入，和传统的生产企业相比较，软件的需求具有模糊性、不确定性、变化性和主观性的特点。

需求获取是通过与用户的交流，对现有系统的观察及对任务进行分析，从而开发、捕获和修订用户的需求。需求获取的主要任务是和用户方的领导层、业务层人员的访谈式沟通，目的是从宏观上把握用户的具体需求方向和趋势，了解现有的组织架构、业务流程、硬件环境、软件环境、现有的运行系统等等具体情况和客观的信息，建立起良好的沟通渠道和方式。

需求获取需要执行的活动如下。

1) 了解客户方的所有用户类型以及潜在的类型。然后，根据他们的要求来确定系统的整体目标和系统的工作范围。

2) 对用户进行访谈和调研。交流的方式可以是会议、电话、电子邮件、小组讨论、模拟演示等形式。需要注意的是，每一次交流一定要有记录，对于交流的结果还可以进行分类，便于后续的分析活动。例如，可以将需求细分为功能需求、非功能需求（如响应时间、平均无故障工作时间、自动恢复时间等）、环境限制、设计约束等类型。

需求调查形式通常会采用如下方式：

- Q&A List 邮件提问。这是向用户调查需求主要的方式。对软件产品需求不明确的问题，整理归纳成 Q&A 列表，通过邮件传递给用户。Q&A 列表能够详细记录问题从不明确到清晰的整个过程，但是比较费时，需求提问的进度取决于用户是否能及时答复邮件。
- 电视电话会议访谈。电视电话会议访谈是一种自由的、开放的获取需求的方式，可以深入探究用户对某些问题的回答，从而得到更准确的信息；对不同的被访谈者可以及时调整问话方式。但是需要注意沟通方式。
- 需求专题讨论会。需求专题讨论会即所谓的头脑风暴（brain storm），它指的是在一段短暂但紧凑的时间段内，把所有与需求相关的人员集中到一起，围绕产品或者项目的目标进行自由讨论，各抒己见，最后统一归纳出初步的需求。这个方式的效率非常高，往往看上去不相关的意见经过综合可以得出很好的主意；可以立即得出结果。但是需要有经验的人组织才能保证成功。
- 自行搜集需求。客户不能提供明确的需求，需要我们自己调查相关的行业标准、同类标准，总结出功能、非功能需求，需求点可以通过前两种方式与委托方确认。

3) 需求分析人员对收集到的用户需求做进一步的分析和整理：

- 对于用户提出的每个需求都要知道“为什么”，并判断用户提出的需求是否有充足的理由；
- 将那种以“如何实现”的表述方式转换为“实现什么”的方式，因为需求分析阶段关注的目标是“做什么”，而不是“怎么做”；
- 分析由用户需求衍生出的隐含需求，并识别用户没有明确提出来的隐含需求（有可能是实现用户需求的前提条件），这一点往往容易忽略掉，经常因为对隐含需求考虑得不够充分而引起需求变更。

4) 需求分析人员将调研的用户需求以适当的方式呈交给用户方和开发方的相关人员。大家共同确

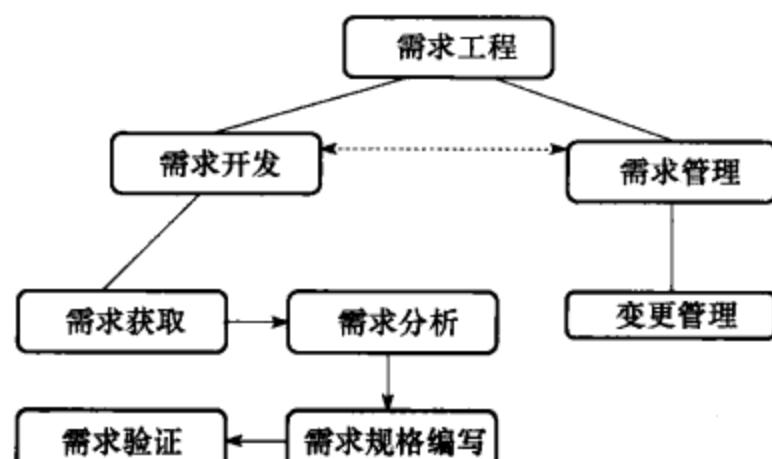


图 2-3 需求管理过程

认需求分析人员所提交的结果是否真实地反映了用户的意图。

需求分析人员在这个任务中需要执行下述活动：

- 明确标识出那些未确定的需求项（在需求分析初期往往有很多这样的待定项）；
- 使需求符合系统的整体目标；
- 保证需求项之间的一致性，解决需求项之间可能存在的冲突。

以上只是一些用得比较多的，比较成功的形式，需求分析员可以按照实际情况采取适合的形式。

进行需求获取的时候应该注意如下问题：

1) 识别真正的客户。识别真正的客户不是一件容易的事情，项目总是要面对多方的客户，不同类型客户的素质和背景都不一样，有的时候没有共同的利益，例如，销售人员希望使用方便，会计人员最关心的是销售的数据如何统计，人力资源关心的是如何管理和培训员工等等。有时他们的利益甚至有冲突，所以必须认识到客户并非政治上平等的，有些人比其他人对项目的成功更为重要，清楚地认识影响项目的那些人，对多方客户的需求进行排序，如果只是局外人来参与项目，可以暂缓考虑其需求。

2) 正确理解客户的需求。客户有时并不十分明白自己的需要，可能提供一些混乱的信息，而且有时会夸大或者弱化真正的需求，所以需要我们既要懂一些心理知识，也要懂一些社会其他行业的知识，了解客户的业务和社会背景，有选择地过滤需求，理解和完善需求，确认客户真正需要的东西。弄清客户的真正需要是什么，例如除了表面的需求，客户个体其实还有隐含的“需要”。举个生活中的例子，买衣服的人都会谈颜色、款式和面料等方面的需求，但买衣服隐含的需要可能是“御寒”，可能是“漂亮”，也可能是“体面”，而且这些需要不会直接说出来。

3) 具备较强的忍耐力和清晰的思维。进行需求获取的时候，应该能够从客户凌乱的建议和观点整理出真正的需求，不能对客户需求的不确定性和过分要求失去耐心，甚至造成不愉快，要具备好的协调能力。

4) 说服和教育客户。需求分析人员可以同客户密切合作，帮助他们找出真正的需求，通过说服引导等手段，也可以通过培训来实现；同时要告诉客户需求可能会不可避免地发生变更，这些变更会给持续的项目正常化增加很大的负担，使客户能够认真对待。

5) 需求获取阶段一般需要建立需求分析小组，进行充分交流，互相学习，同时要实地考察访谈，收集相关资料，进行语言交流，必要时可以采用图形表格等工具。

## 2.2.2 需求分析

需求分析是开发人员对系统需要做什么的定义过程。从系统分析的经验来看，这个过程往往是个循序渐进的过程，一次性对系统形成完整的认识是困难的。只有不断地和客户领域专家进行交流确认，方能逐步明了用户的需求。从系统开发的过程得知，系统需求分析时犯下的错误，会在接下来的阶段被成倍的放大，越是在开发的后期，纠正分析时犯下的错误所花费的代价越是昂贵，也越发影响系统的工期和系统的质量。

需求分析也称为需求建模，是为最终用户所看到的系统建立一个概念模型，是对需求的抽象描述，并尽可能多的捕获现实世界的语义。需求分析的任务就是借助于当前系统的逻辑模型导出目标系统的逻辑模型，解决目标系统的“做什么”的问题。见图 2-4。

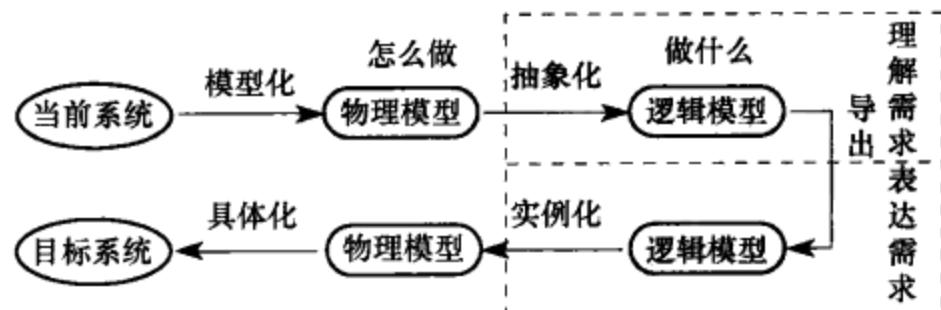


图 2-4 需求分析模型

需求是与技术无关（technology independent）的。在需求阶段讨论技术是没有任何意义的。那只会让你的注意力分散。技术的实现细节是在后面的设计阶段才需要考虑的事情。在很多情形下，分析用户需求是与获取用户需求并行的，主要通过建立模型的方式来描述用户的需求，为客户、用户、开发方等不同参与方提供一个交流的渠道。这些模型是对需求的抽象，以可视化的方式提供一个易于沟通的桥梁。需求分析与需求获取有着相似的步骤，区别在于分析用户需求时使用模型来描述，以获取用户更明确的需求。

分析用户需求应该执行下面的活动：

- 1) 以图形表示的方式描述系统的整体结构，包括系统的边界与接口。
- 2) 通过原型、页面流或其他方式向用户提供可视化的界面，用户可以对需求做出自己的评价。
- 3) 以模型描述系统的功能项、数据实体、外部实体、实体之间的关系、实体之间的状态转换等方面的内容。

需求分析的基本策略是采用脑力风暴、专家评审、焦点会议组等方式进行具体的流程细化、数据项的确认，必要时可以提供原型系统和明确的业务流程报告、数据项表，并能清晰地向用户描述系统的业务流设计目标。用户方可以通过审查业务流程报告、数据项表以及操作开发方提供的原型系统，来提出反馈意见，并对可接受的报告、文档签字确认。

软件需求分析人员不强求都是全才，更不可能强求行业方面的专家。客户表达的需求，不同的分析人员可能有不同的理解。如果分析人员理解错了，可能会导致以后的开发工作劳而无功。所以，进行需求分析的时候，要求开发人员的知识面要宽一些！

需求不明确是软件开发过程中经常遇到的问题，这类问题往往表现在需求范围未界定、需求未细化、需求描述不清楚、需求遗漏、需求互相矛盾等多个方面。在软件开发过程各阶段中，需求不明确所造成的浪费是最大的，必须尽早尽可能解决。确定用户需求是件非常困难的事情，我们常常从以下几个方面着手处理需求不明确问题：

1) 让用户参与开发。提供一个协作开发环境，让用户参与开发过程。如果条件不允许，至少应该在每次迭代的需求分析和系统测试阶段，让客户能够参与开发。在选择参与开发过程的用户时，一方面，要尽可能争取精通业务或计算机技术的用户参与。另一方面，如果开发的产品要在不同规模、不同类型的企业应用，应该选择具有代表性的用户参与。

虽然尽早的邀请用户参与，仍然避免不了项目进行中用户的需求变更请求。对于开发过程存在的需求变动，我们要求用户填写变更申请单发送给项目配置管理员，然后一起讨论实施变更的可行性及实施后所带来的影响，无论那种变更都需要对相应的文档实施同步变更（包括需求规格说明书、详细设计文档、安装手册、操作手册等）。但是对于无法实现或是变更会带来巨大的影响而将导致进度的延期，这时，我们将变更报告提交给用户或邀请用户进行协调会议，讨论变更取舍问题或是项目进度变更问题。当然，仅仅让用户参与是不够的，应该采取一定的激励措施，提高用户参与的积极性。

2) 开发用户界面原型。用户通常不善于精确描述自己的业务需求，系统分析员需要借助白板、白纸等沟通方式，帮助用户清楚表述需求。然后，开发一个用户界面原型，以便用户确认需求。

3) 需求讨论会议。对于用户分布广、用户量大的项目，要全面收集用户需求，往往很困难，通常采取需求研计会议方式进行需求确认。通过在会议前几周调查各地、各部门用户需求意见，然后集中各地或各部门的用户代表，举办一次需求研讨会，通过会议方式收集需求。本方法适合于具有一定信息系统使用经验的用户。

4) 强化需求分析与评审。首先，需求分析是项目成功的基础，需要引起足够的重视，并分配充足的时间和人力，要让有经验的系统分析员负责，切忌让项目新手或程序员负责。其次，要进行需求评审，尽可能让用户参与需求评审，不要让需求评审流于行式。第三，也是最重要的一点，通过评审的需求规格说明书，要让用户方签字，并作为项目合同的附件，对双方都具有约束力。

### 2.2.3 需求规格编写

对于软件这个特殊领域的业务需求，项目经理在管理需求的时候，一定要求需求分析的人员首先获取用户的真正要求，即使是双方画的简单的流程草案也很重要，然后根据获取的真正的要求采取适当的方法编写需求规格。

软件需求规格的编制是为了使用户和软件开发者双方对该软件的初始规定有一个共同的理解，使之成为整个开发工作的基础。需求分析完成的标志是提交一份完整的软件需求规格说明书（SRS）。建立了需求规格文档，才能描述要开发的产品，并作为项目演化的指导。如果没有需求规格文档，那么需求就隐含地确定和影响项目的内容和项目的成功。需求规格说明书（SRS）以一种开发人员可用的技术形式、陈述了一个软件产品所具有的基本特征和性质，以及期望和选择的特征和性质。对项目来说，需求规格

文档 SRS 和工作陈述 SOW(Statement of Work) 是很关键的两个文档，SRS 的编写可以参照甲方提供的 SOW 的有关信息进行，SRS 为客户提供和开发者之间建立一个约定，准确地陈述了要交付给客户什么。

软件产品范围是指软件产品所包含的特征或功能，而软件需求说明书正是对软件产品范围正式书面的界定，是软件项目管理过程必须的基础性文档。

需求规格相当于软件开发的图纸，一般说，软件需求规格说明书的格式可以根据项目的具体情况采用不同的格式，没有统一的标准。

下面是一个可以参照的软件需求规格模板。

## 1. 导言

### 1.1 目的

[说明编写这份项目需求规格的目的，指出预期的读者。]

### 1.2 背景

说明：

- a. 待开发的产品的名称。
- b. 本项目的任务提出者、开发者、用户及实现该产品的单位。
- c. 该系统同其他系统的相互来往关系。

### 1.3 缩写说明

[缩写]

[缩写说明]

列出本文件中用到的外文首字母组词的原词组。

### 1.4 术语定义

[术语]

[术语定义]

列出本文件中用到的专门术语的定义。

### 1.5 参考资料

[编号]《参考资料》[版本号]

列出相关的参考资料。

### 1.6 版本更新信息

具体版本更新记录下表所示。

修改编号	修改确认日期	修改后版本	修改位置	修改方式 (AMD)	修改内容概述	修改请求号

修改方式：A - 增加 M - 修改 D - 删除

## 2. 任务概述

### 2.1 系统定义

本节描述内容包括：

- 项目来源及背景。
- 项目要达到的目标，如市场目标、技术目标等。
- 系统整体结构，如系统框架、系统提供的主要功能，涉及的接口等。
- 各组成部分结构，如果所定义的产品是一个更大的系统的一个组成部分，则应说明本产品与该系统中其他各组成部分之间的关系，为此可使用一张方框图来说明该系统的组成和本产品同其他各部分的联系和接口。

### 2.2 应用环境

本节应根据用户的要求对系统的运行环境进行定义，描述内容包括：

- 设备环境
- 系统运行硬件环境
- 系统运行软件环境

- 系统运行网络环境
- 用户操作模式
- 当前应用环境

### 2.3 假定和约束

列出进行本产品开发工作的假定和约束。例如经费限制、开发期限等，列出本产品的最终用户的特点，充分说明操作人员、维护人员的教育水平和技术专长，以及本产品的预期使用频度等重要约束。

## 3. 需求规定

### 3.1 对功能的规定

本节依据合同中定义的系统组成部分分别描述其功能，描述应包括：

- 功能编号
- 所属产品编号
- 优先级
- 功能定义
- 功能描述

### 3.2 对性能的规定

本节描述用户对系统的性能需求，可能的系统性能需求有：

- 系统响应时间需求
- 系统开放性需求
- 系统可靠性需求
- 系统可移植性和可扩展性需求
- 系统安全性需求
- 现有资源利用性需求

#### 3.2.1 精度

说明对该产品的输入、输出数据精度的要求，可能包括传输过程中的精度。

#### 3.2.2 时间特性要求

说明对于该产品的时间特性要求，如对：

- a. 响应时间；
- b. 更新处理时间；
- c. 数据的转换和传送时间；
- d. 计算时间等的要求。

#### 3.2.3 灵活性说明对该产品的灵活性的要求，即当需求发生某些变化时，该产品对这些变化的适应能力，如：

- a. 操作方式上的变化；
- b. 运行环境的变化；
- c. 同其他系统的接口的变化；
- d. 精度和有效时限的变化；
- e. 计划的变化或改进；

对于为了提供这些灵活性而进行的专门设计的部分应该加以标明。

### 3.3 输入输出的要求

解释各输入输出数据类型，并逐项说明其媒体、格式、数值范围、精度等。对软件的数据输出及必须标明的控制输出量进行解释并举例，包括对硬拷贝报告（正常结果输出、状态输出及异常输出）以及图形或显示报告的描述。

### 3.4 数据管理能力要求

说明需要管理的文卷和记录的个数、表和文的大小规模，要按可预见的增长对数据及分量的存储要求作出估算。

### 3.5 故障处理要求

列出可能的软件、硬件故障以及对各项性能而言所产生的后果和对故障处理的要求。

### 3.6 其他要求

如用户单位对安全保密的要求，对使用方便的要求，对可维护性、可补充性、易读性、可靠性、运行环境可转换性的特殊要求等。

### 4. 运行环境规定

#### 4.1 设备

列出该产品所需要的硬件环境。说明其中的新型设备及其专门功能，包括：

- a. 处理器型号及内存容量；
- b. 外存容量、联机或脱机、媒体及其存储格式，设备的型号及数量；
- c. 输入及输出设备的型号和数量，联机或脱机；
- d. 数据通信设备的型号和数量；
- e. 功能键及其他专用硬件。

#### 4.2 支持软件

列出支持软件，包括要用到的操作系统、编译程序、测试软件等。

#### 4.3 双方签字

需求方（甲方）：

开发方（乙方）：

日期：

有一种非常危险的思想认为在项目的需求分析阶段，开发方与客户方在各种问题的基本轮廓上达成一致即可，具体细节可以在以后填充。实际上许多软件项目失败的最主要的原因就是需求阶段对问题的描述不够细致，导致后来预算超出或者时间进度达不到要求。正确的做法是：在项目需求分析阶段，双方必须全面地尽可能细致地讨论项目的应用背景、功能要求、性能要求、操作界面要求、与其他软件的接口要求，以及对项目进行评估的各种评价标准。并且，在需求分析结束以后，双方还要建立可以直接联系的渠道，以尽早地对需求变动问题进行沟通。

## 2.2.4 需求验证

需求规格提交后，开发人员需要与客户对需求分析的结果进行验证，以需求规格说明为输入、通过符号执行、模拟或快速原型等途径，分析需求规格的正确性和可行性。

验证需求的正确性及其质量，就能大大减少项目后期的返工现象。在项目计划中应为这些保证质量的活动预留时间并提供资源。从客户代表方获得参与需求评审的赞同（承诺），并尽早且以尽可能低的成本，通过非正式的评审逐渐到正式评审来找出其存在的问题。

验证需求包括以下几个方面：

1) 需求的正确性。开发人员和用户都进行复查，以确保将用户的需求充分、正确的表达出来。只有进行一番调查研究，才能知道某一项需求是否确实正确。每一项需求都必须准确地陈述其要开发的功能。做出正确判断的参考是需求的来源，如用户或高层的系统需求规格说明。若软件需求与对应的系统需求相抵触则是不正确的。只有用户代表才能确定用户需求的正确性，这就是一定要有用户的积极参与的原因。没有用户参与的需求评审将导致评审者凭空猜测。

2) 需求的一致性。一致性是指与其他软件需求或高层（系统，业务）需求不相矛盾。在开发前必须解决所有需求间的不一致部分。验证没有任何的冲突和含糊的需求，没有二义性。

3) 需求的完整性。验证是否所有可能的状态、状态变化、转入、产品和约束都在需求中描述，不能遗漏任何必要的需求信息。遗漏需求将很难查出。注重用户的任务而不是系统的功能将有助于你避免不完整性。如果知道缺少某项信息，用 TBD（“待确定”）作为标准标识来标明这项缺漏。在开始开发之前，必须解决需求中所有的 TBD 项。

4) 需求的可行性。验证需求是否实际可行，每一项需求都必须是在已知系统和环境的权能和限制范围内可以实施的。为避免不可行的需求，最好在获取（elicitation）需求（收集需求）过程中始终有一位软件工程小组的组员与需求分析人员或考虑市场的人员在一起工作，由他负责检查技术可行性。

5) 需求的必要性。验证需求是客户需要的，每一条需求描述都是用户需要的，每一项需求都应把客户真正所需要的和最终系统所需遵从的标准记录下来。“必要性”也可以理解为每项需求都是用来授权你编写文档的“根源”。要使每项需求都能回溯至某项客户的输入。

6) 需求的可检验性。验证是否能写出测试案例来满足需求。检查一下每项需求是否能通过设计测试用例或其他的验证方法，如用演示、检测等来确定产品是否确实按需求实现了。如果需求不可验证，则确定其实施是否正确就成为主观臆断，而非客观分析了。一份前后矛盾、不可行或有二义性的需求也是不可验证的。有的项目范围的定义不够明确，做不到量化、可验证程度不高。很多时候都是一些定性的要求、而不是定量的，例如“界面友好，可操作性强，提高用户满意度”等。类似这些模糊的需求就是导致后续项目扯皮的根源。项目范围的明确定义，有经验的项目经理及系统分析员将起到至关重要的作用。

7) 需求的可跟踪性，验证需求是否是可跟踪的；应能在每项软件需求与它的根源和设计元素、源代码、测试用例之间建立起链接链，这种可跟踪性要求每项需求以一种结构化的、细粒度的方式编写并单独标明，而不是大段的叙述。

软件开发的基本目标是构造能够满足客户基本需求的软件系统，因此需要一种检测手段来检验软件是否满足所有的需求。需求跟踪是一种有效的检测手段，它要求对每一项需求都可以追踪到实现该需求的设计、编码及测试用例。通过需求跟踪，可以检验软件是否实现了所有需求以及软件是否对所有的需求都经过了测试。

有两种类型的跟踪：前向跟踪和后向跟踪。前向跟踪意味着看需求是否在生命周期的后续阶段（如设计和编码阶段）的工作成果中得到体现。后向跟踪则意味着看生命周期后期的各个阶段的工作成果满足何种需求。前向跟踪保证了软件能够满足需求。后向跟踪则在变更、回归测试等情况下更有用。

#### 8) 最后的签字。

实践表明，高质量的需求分析是软件项目成功的关键因素。软件项目的范围控制应该是在需求分析阶段就开始的，就是说软件需求说明书应该是最大可能最大程度地理解了客户实际业务需求的文档，特别对于复杂的业务流程型项目，涉及多方客户干系人需求的项目和涉及引发客户方机构变革的项目，一般应委托客户方关键性干系人内部协调达成一致意见后确定需求，切不可凭经验自作主张想当然。在编码阶段开始后，应做好产品范围的变更控制，尽可能地对客户施加影响，避免需求变更的发生，实在无法避免的变更一定要采用正式书面的形式。

对于我们软件开发人员来说，客户的需求也应该是最大的，不仅仅因为客户是我们的衣食父母，也是我们软件开发人员成长的催化剂。我们不是产品的最终客户，所以有的时候关于某个特定系统上的具体细节可能是我们没有办法考虑到的，是在我们的领域里可能没有办法想象得到的，这个时候我们就必须要通过实际应用系统的最终客户，也就是我们项目的最终客户来了解具体的客户需求。对于一个缺乏需求管理的软件项目而言，必定会导致系统不能实现预期的功能而需要在后期进行昂贵的修正，使得项目拖期、产生严重的质量问题与超出项目预算的现象。

了解了客户的需求，可以让我们在软件的开发过程中少走很多的弯路，缩短软件开发的周期，能够提高软件的友好性，易操作性，易用性，从而来提升软件的质量。

### 2.2.5 需求变更

需求变更是软件项目一个突出的特点，也是软件项目最为普遍的一个特点。做过项目的人可能都会有这样的经历：一个项目做了很久，感觉总是做不完，就像一个“无底洞”。虽然这与人类认识问题的自然规律是一致的，但是频繁而无管理的需求变更非常容易导致复杂、无形的软件在多变的情况下失控，加剧了软件开发过程中的不稳定性，从而造成多方的损失。那么如何对需求变更加以有效的控制和管理，从而保证软件开发的进度、成本和质量，便成为软件开发过程中一个值得思考的问题。

客户的需求为什么一变再变？人类认识世界是一个由无知到已知、由浅入深的过程。我们以及客户对需求的认识也是一个逐步深入逐步明晰的过程。随着认识的深入，客户的需求才逐渐变的明确。软件人员在最初的时候就需要帮助客户深化认识、明确需求。

变化并不是人们最害怕的，最怕的是跟不上变化的步伐。同样，在软件开发过程中需求的变更会

给开发带来不确定性。所以必须接受“需求会变动”这个事实，做好需求变更的管理工作。

需求管理主要的工作如下：

- 1) 建立需求基线。需求基线是需求变更的依据。在开发过程中，需求确定并经过评审后（用户参与评审），可以建立第一个需求基线。此后每次变更并经过评审后，都要重新确定新的需求基线。需求的基线是指是否容许需求变更的分界线。需求分析人员在充分与客户用户进行沟通的基础上形成第一个版本的需求文档，这个需求文档在通过需求评审后即可以建立第一个需求基线。变更控制委员会要有效进行需求变更控制。
- 2) 确定需求变更控制过程。制订简单、有效的变更控制流程，并形成文档。在建立了需求基线后提出的所有变更都必须遵循这个变更控制流程进行控制。同时，这个流程具有一定的普遍性，对以后的项目开发和其他项目都有借鉴作用。
- 3) 建立变更控制委员会（SCCB）。成立项目变更控制委员会（SCCB）或相关职能的类似组织，负责裁定接受哪些变更。SCCB 由项目所涉及的多方人员共同组成，应该包括用户方和开发方的决策人员在内。
- 4) 进行需求变更影响分析。需求变更一定要先申请然后再评估，最后经过与变更大小相当级别的评审确认。对于提交的每项需求变更请求，应确定它对项目整体进度的影响和对其他相关开发任务的影响，并且一定要明确完成这些变更相关任务的工作量。只有经过全面的分析，变更控制委员会才能够做出更好的决策。进行变更影响分析可以对申请的需求变更有更深刻的理解，通过对变更内容的更深刻的理解，才能做出对正在进行的工作的调整部署。
- 5) 跟踪所有受需求变更影响的工作产品。需求变更后，受影响的软件计划、产品、活动都要进行相应的变更，以保持和更新的需求一致。记录每个需求变更文档的版本号、日期、所做的变更、原因等，当然应该明确该文档由谁来负责更新。
- 6) 建立需求基准版本和需求控制版本文档。
- 7) 维护需求变更的历史记录。妥善保存变更产生的相关文档。
- 8) 跟踪每项需求的状态。
- 9) 衡量需求稳定性。变更控制委员会需要对需求变更的整体有良好的把握，通过记录需求基准的数量可以获得宏观需求的变更次数，同时还应该记录一段时间内（如每周、每月）的变更数量，最好按变更的类别来列出详细信息。如果某一需求过于频繁变更，则说明对该问题的认识还不深入或者说还没有达成一致的处理意见。如果需求变更的总体数量过高，则意味着项目范围并未很好地确定下来或是政策变化较大。

并非对需求定义的越细，越能避免需求的渐变，这是两个层面的问题。太细的需求定义对需求渐变没有任何效果。因为需求的变化是永恒的，并非由于需求写细了，它就不会变化了。

## 2.3 编写需求规格的方法

需求分析针对需求做出分析并提出方案模型。需求分析的模型正是产品的原型样本，优秀的需求管理提高了这样的可能性，它使最终产品更接近于解决需求，提高了用户对产品的满意度，从而使产品成为真正优质合格的产品。从这层意义上说，需求管理是产品质量的基础。

需求建模的方法有很多种，例如：原型分析方法，结构化分析法，用例分析法，功能列表法，以及其他方法等。目前，比较流行的是原型分析法和用例分析法。

详见《软件工程案例教程》。

## 2.4 任务分解定义

项目管理的第一法则就是“做正确的事”，其次才是“正确地做事”。项目开发范围出错，就是走向失败的第一步。摆脱开发项目的“无底洞”必须明确项目范围，表现在顾客的可靠需求式样、交付成果及项目结束条件。没有工作范围的定义，项目就可能成为无底洞。可靠的需求式样是指经各方一致同意的、清楚的、完整的、详细的、整体的、可实现的、可测试的需求。交付成果物是项目的最终输出，是项目投入各种资源的依据。项目范围管理的作用就是保证项目计划包括且仅包括为成功地完

成项目所需要进行的所有工作。

范围管理计划是描述项目范围如何进行管理，项目范围怎样变化才能与项目要求相一致等问题的。它也应该包括一个对项目范围稳定的预期而进行的评估（比如：怎样变化、变化频率如何及变化了多少）。范围管理计划也应该包括对变化范围怎样确定，变化应归为哪一类（当产品特征仍在被详细描述的时候，做到这点特别困难，但绝对必要）等问题的清楚描述。

范围计划编制是将产生项目产品所需进行的项目工作（项目范围）渐进明细和归档的过程。做范围计划编制工作是需要参考很多信息的，比如产品描述，首先要清楚最终产品的定义才能规划要做的工作，项目章程也是主要的依据，通常它对项目范围已经有了粗线条的约定，范围计划在此基础上进一步深入和细化。

范围说明在项目参与人之间确认或建立了一个项目范围的共识，作为未来项目决策的文档基准。范围说明中要说明项目论证、项目产品、项目可交付成果和项目目标。项目论证是商家的既定目标，要为估算未来的得失提供基础；项目产品是产品说明的简要概况；项目可交付成果一般要列一个子产品级别概括表，如：为一个软件开发项目设置的主要可交付成果可能包括程序代码、工作手册、人机交互学习程序等。任何没有明确要求的结果，都意味着它在项目可交付成果之外；项目目标是要考虑到项目的成功性，至少要包括成本、进度表和质量检测。项目目标应该有标志（如：成本、单位）和绝对的或相对的价值（如：少于 150 万美元等）。不可量化的目标（如：“客户的满意程度”）要承担很高的风险。

#### 2.4.1 WBS

当解决问题过于复杂时，可以将问题进行分解，直到分解后的子问题容易解决；然后分别解决这些子问题。规划项目时，也应该从任务分解开始，将一个项目分解为更多的工作细目或者子项目，使项目变得更小、更易管理、更易操作。这样可以提高估算成本、时间和资源的准确性。使工作变得更易操作，责任分工更加明确。完成项目本身是一个复杂的过程，必须采取分解的手段把主要的可交付成果分成更容易管理的单元才能一目了然，最终得出项目的分解结构（Work Breakdown Structure，WBS）。

任务分解是对需求的进一步细化，是最后确定项目所有任务范围的过程。任务分解的结果是任务分解结构 WBS。WBS 是面向可交付成果的对项目元素的分组，它组织并定义了整个项目的范围。不包括在 WBS 中的工作就不是该项目的工作。只有在 WBS 中的工作才是该项目的工作范围，它是一个分级的树型结构，是对项目由粗到细的分解过程。

“化繁为简，分而治之”是自古以来解决复杂问题的不二法门，对于软件项目来讲，可以将大的项目划分成几个小项目来做，将周期长的项目化分成几个明确的阶段。

项目越大对项目组的管理人员、开发人员的要求越高，参与的人员越多，需要协调沟通的渠道越多，周期越长，开发人员也容易疲劳，将大项目拆分成几个小项目，可以降低对项目管理人员的要求，减少项目的管理风险，而且能够充分地将项目管理的权力下放，充分调动人员的积极性，目标会比较具体明确，易于取得阶段性的成果，使开发人员有成就感。

“小就是美”，这句话对 IT 项目管理同样适用。项目范围、项目工期和项目规模（成本）可以看成是支持项目成功的三大支柱。尽管 IT 项目的范围越小、项目组规模越小，项目成功的可能性越大，我们却不可能只做规模小的项目，也不可能将规模大的项目人为地缩小。我们能做的只是将项目划分成生命周期的不同阶段以简化项目的复杂度、提高项目的成功率。

任务分解结构每细分一个层次表示对项目元素更细致的描述。其中，工作包（work package）是 WBS 的最低层次的可交付成果，项目完成时，应该完成这些交付成果，这些交付成果也可以通过子项目的方式完成，分配给另外一位项目经理进行计划和执行，这时工作包可进一步分解为子项目的 WBS 或各个活动，这种工作包应当由唯一一个主体负责。任务分解是项目评估的前提和自下而上估算法的基础。图 2-5 是任务分解的例子。

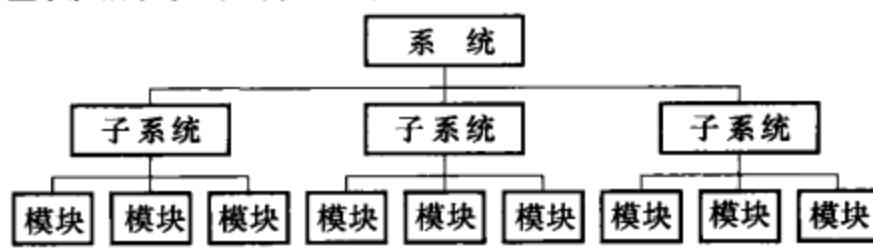


图 2-5 任务分解

WBS 的建立对项目来说意义非常重大，它使得原来看起来非常笼统、非常模糊的项目目标一下子清晰下来，使得项目管理有依据，项目团队的工作目标清楚明了。如果没有一个完善的 WBS 或者范围定义不明确时，变更就不可避免地出现，很可能造成返工、延长工期、降低团队士气等一系列不利的后果。

制定好一个 WBS 的指导思想是逐层深入。先将项目成果框架确定下来，然后每层下面再进行工作分解，这种方式的优点是结合进度划分直观，时间感强，评审中容易发现遗漏或多出的部分，也更容易被大多数人理解。

WBS 中的每一个具体细目通常都指定唯一的编码（Code of Accounts），这对有效的控制整个项目的系统非常重要，不管是对项目经理，还是项目团队成员或者其他项目干系人而言，项目组成元素的编码对所有人来说应当是有共同的认知。因此，WBS 的编码设计与结构设计应该有一一的对应关系，即结构的每一层次代表编码的某一个位数，同时，项目的各组成元素的编码都是唯一的。例如图示 2-6 是确定了 WBS 编码的任务分解结果。

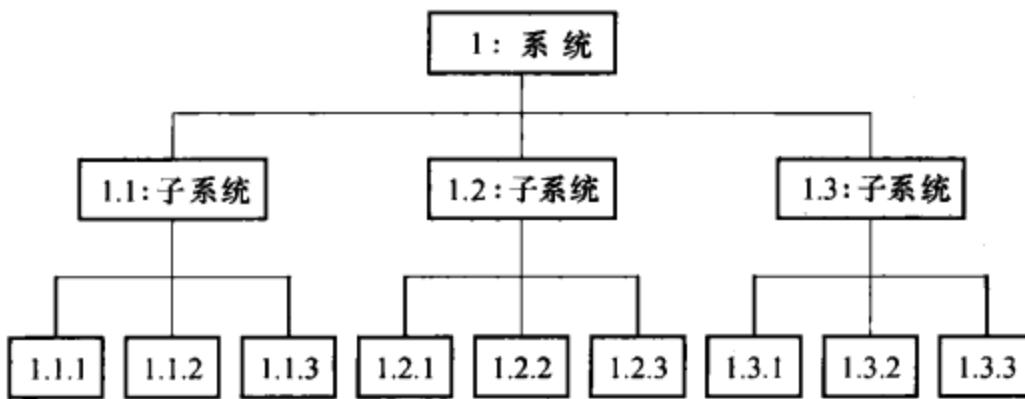


图 2-6 有编码的 WBS

## 2.4.2 任务分解的类型

一般说，进行任务分解时，可以采用清单或者图表的形式表达任务分解的结果。

### 1. 清单类型

采用清单类型的任务分解方式，就是将任务分解的结果以清单的表述形式进行层层分解的方式。现在以一个项目为例子进行说明，这个项目的名字定义为“变化计数器”，它是统计程序大小的软件工具，当修改一个程序的时候，这个工具可以统计各个版本之间有多少代码行增加、删除或修改。这个项目任务分解可以按照不同的标准进行分解，采用清单方式进行任务分解如下：

1. 变化计数器
  - 1.1 比较两个版本的程序
    - 1.1.1 预处理
    - 1.1.2 文件比较
    - 1.1.3 结果处理
  - 1.2 找出修改后的程序中增加和删除的代码行
    - 1.2.1 找出增加的代码行
    - 1.2.2 找出删除的代码行
  - 1.3 统计修改后的程序中增加和删除的代码行数
    - 1.3.1 统计增加代码行数
    - 1.3.2 统计删除代码行数
  - 1.4 统计总的代码行数
  - 1.5 设定标记以指示修改的次数
  - 1.6 在程序的头部增加修改记录

## 2. 图表类型

采用图表类型的任务分解过程就是进行任务分解时采用图表的形式进行层层分解的方式。例如，对于上面的“变化计数器”这个项目采用图表类型的分解结果如下图 2-7。

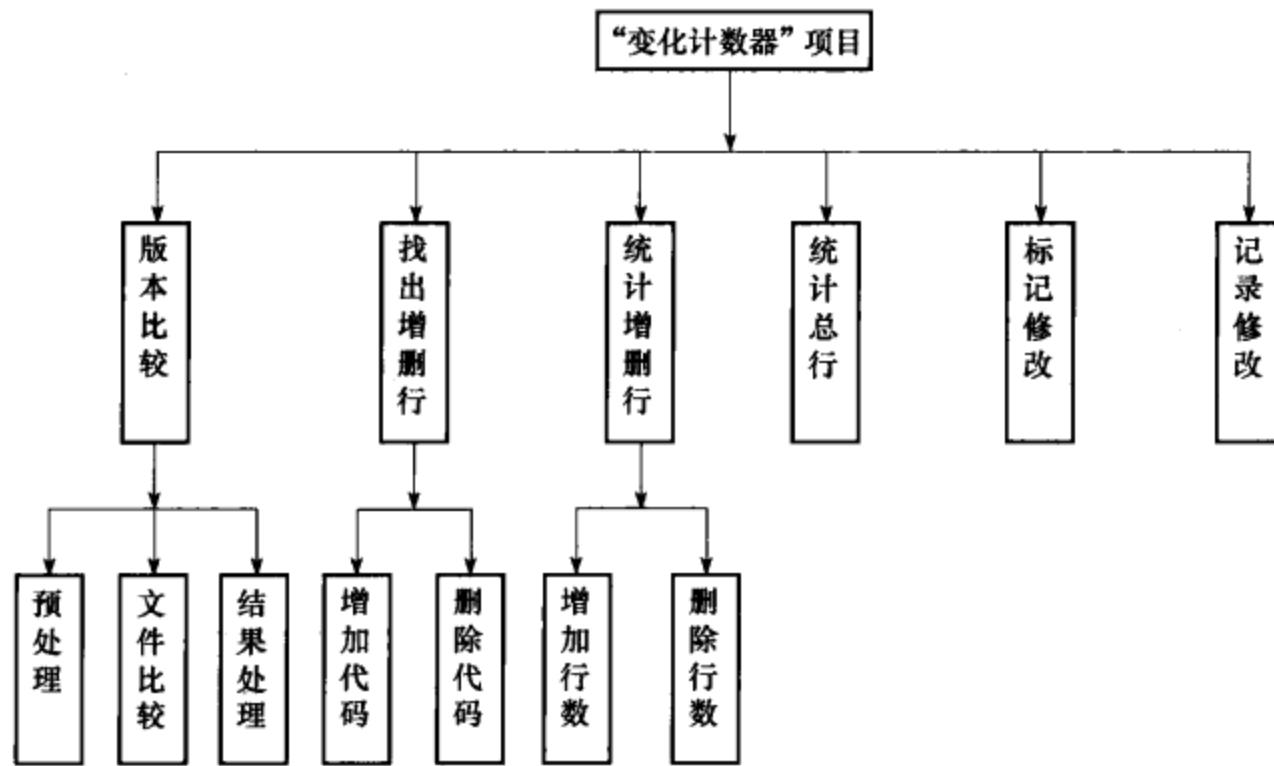


图 2-7 变化计数器图表分解

### 2.4.3 任务分解的过程

进行任务分解应该采取一定的步骤，并且分解过程中保持唯一的分解标准。任务分解的基本过程图示如 2-8。

任务分解应该根据需求分析的结果和项目相关的要求，同时参照以往的项目分解结果进行分解，最终任务分解的结果是 WBS。



图 2-8 任务分解过程

#### 1. 基本步骤

分解意味着分割主要工作细目，使它们变成更小、更易操作的要素，直到工作细目被明确详细的界定，它有助于未来项目的具体活动（规划、评估、控制和选择）的开展。一般说，进行任务分解的基本步骤是：

- 1) 确认并分解项目的主要组成要素。通常，项目的主要要素是这个项目的工作细目。以项目目标为基础，作为第一级的最整体的要素。项目的组成要素应该用有形的、可证实的结果来描述，目的是为了绩效易检测。当我们知道了主要构成要素后，这些要素就应该用项目工作怎样开展、在实际中怎样完成形式来定义。有形的、可证实的结果既包括服务，也包括产品。

- 2) 确定分解标准，按照项目实施管理的方法分解，而且分解的时候标准要统一。分解要素是根据项目的实际管理而定义的。不同的要素有不同的分解层次。例如：项目生存期的阶段可以当作第一层次的划分，把第一层次中的项目细目在第二阶段继续进行划分。

- 3) 确认分解是否详细，是否可以作为费用和时间估计的标准，明确责任。工作细目的分解如果在很久的将来才能完成的话，那么这种分解也就没了确定性。

- 4) 确定项目交付成果。交付成果是有衡量标准的，以此检查交付结果。

- 5) 验证分解正确性。验证分解正确后，建立一套编号系统。

## 2. 分解的标准

进行任务分解的标准应该统一，不能有双重标准，选择一种项目分解标准之后，在分解过程中应该统一使用此标准，避免使用不同标准导致的混乱。可以采用生存期为标准；或者以功能（产品）组成为标准；或者其他的方法等等。例如“变化计数器”这个项目的任务分解采用两种标准进行分解结果如下。

项目“变化计数器”按照功能组成标准进行分解的结果为：

- 1) 比较两个版本的程序。
- 2) 找出修改后的程序中增加和删除的代码行。
- 3) 统计修改后的程序中增加和删除的代码行数。
- 4) 统计总的代码行数。
- 5) 设定标记以指示修改的次数。
- 6) 在程序的头部增加修改记录。

项目“变化计数器”按照生存期标准进行分解的结果为：

- 1) 规划。
- 2) 需求。
- 3) 设计。
- 4) 编码。
- 5) 测试。
- 6) 提交。

如果同时使用这两个标准进行任务分解，可能会有如下的结果：

- 1) 比较两个版本的程序。
- 2) 找出修改后的程序中增加和删除的代码行。
- 3) 统计修改后的程序中增加和删除的代码行数。
- 4) 统计总的代码行数。
- 5) 设定标记以指示修改的次数。
- 6) 在程序的头部增加修改记录。
- 7) 规划。
- 8) 需求。
- 9) 设计。
- 10) 编码。
- 11) 测试。
- 12) 提交。

因此，同时使用多种标准进行任务分解就会导致混乱，会导致任务的重叠。所以，进行任务分解时使用的标准应该统一。

## 2.5 任务分解的方法

任务分解有很多具体方法，例如模板参照、类比、自顶向下和自底向上等方法。

### 2.5.1 模板参照

许多应用领域都有标准或半标准的WBS，它们可以当作模板参考使用。例如图2-9是有些软件企业进行项目分解的WBS模板，当然，本图仅作为参考示例，不代表任何特定项目的具体分解标准，而且也不是唯一的参照模板。

有些企业都有一些WBS分解的指导说明和模板，项目人员应该通过评估相应的信息来开发项目的WBS。

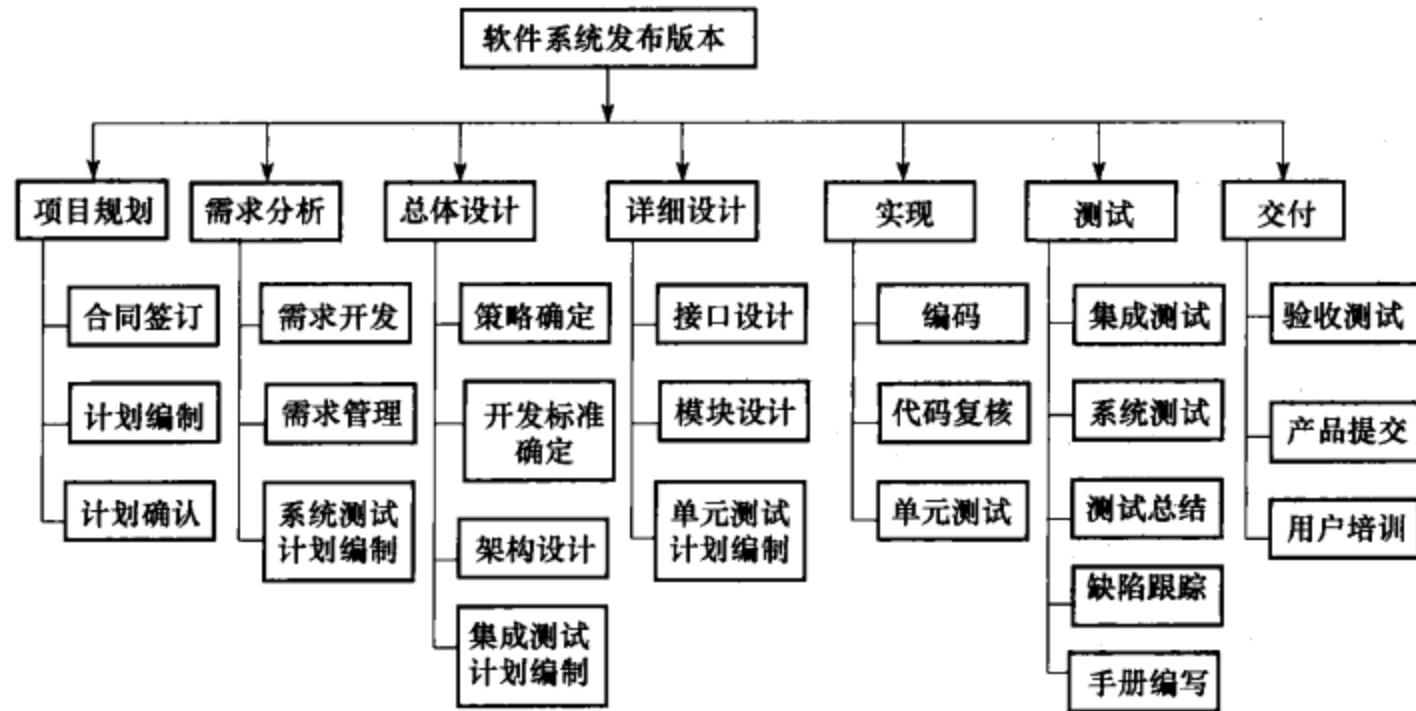


图 2-9 一个 WBS 模板

### 2.5.2 类比方法

虽然每个项目是唯一的，但是，WBS 经常能被“重复使用”，有些项目间在某种程序上是具有相似性的。例如从每个阶段看，许多项目有相同或相似的周期和因此而形成的工作细目要求。可以采用类似的项目的 WBS 作为参考的，一些企业保存一些项目的 WBS 库和一些项目文档为其他项目的开发提供参照。很多项目管理工具都提供了一些 WBS 的例子作为参考，你可以选择一些类似项目作为参考进行开发 WBS。

### 2.5.3 自顶向下方法

自顶向下方法采用的是演绎推理方法，这是因为它沿着从一般到特殊的方向进行。从项目的大局着手，然后逐步分解子细目，将项目变为更细更完善的部分，例如图 2-10。

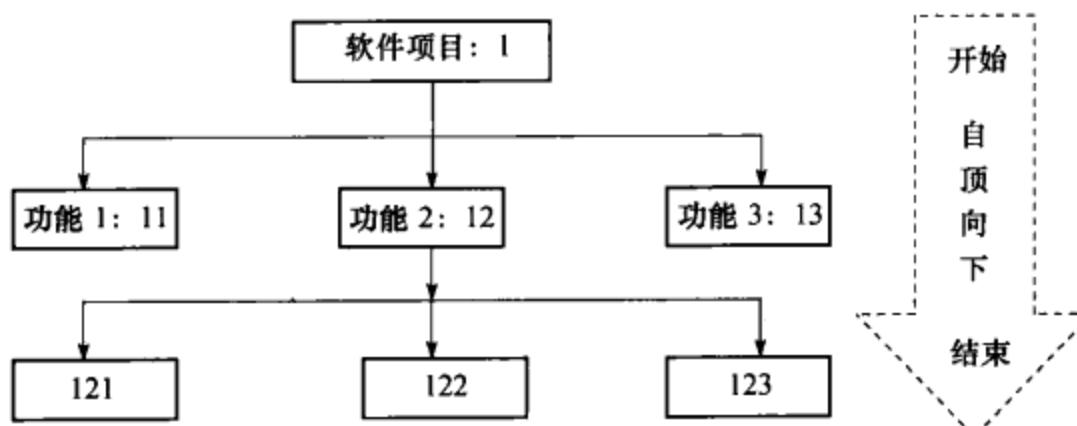


图 2-10 自顶向下的方法

自顶向下方法需要有更多的逻辑和结构，它也是创建 WBS 的最好方法。使用自顶向下方法来生成 WBS 要先确定每一个解决方案，而后将该方案划分成为能够实际执行的若干步骤。在日常生活当中，你可能已经不自觉地使用过自顶向下的工作方法。例如，当你决定要购买一辆小汽车时，你需要先确定买那种类型的汽车：运动型多用途车、赛车、轿车、迷你货车。然后是你能够买得起什么车？什么颜色？喇叭和汽笛如何？可以看出你的思维过程就是一个从主要问题逐渐细化到具体问题的过程。

如果 WBS 开发人员对项目比较熟悉或者对项目大局有把握，可以参用自顶向下方法。应用自顶向下方法开发 WBS 时，可以采用下面的操作：

首先确定主要交付成果或者阶段，将它们分别写在即时贴上，然后按照一定的顺序将它们贴在白板上。现在开始考察第1个交付成果或者第1阶段，将这些部件分解为更小的交付成果。然后继续分解这些交付成果直到分解为比较容易管理的工作包，即WBS的最小的单元。

分解项目交付成果需要一定的技巧。你可能不能将任务划分得太细，但你一定想合适的时间和资源分配给每个阶段中必须完成的活动。你只要能够把握大方向，然后给团队成员分配他们应该完成的工作，而不必详细描述具体的工作机制，这样就可以了。

完成了第1个主要交付成果或者完成第1阶段以后，你就可以进行第2个交付成果或者第2阶段的工作，以此类推，并重复上述过程。直到所有的交付成果都被分解成工作包，现在，呈现在你面前的白板上一定已经粘满了即时贴。实际上，它已经清楚地表达了项目的执行全程。

#### 2.5.4 自底向上方法

自顶向下方法从一般到特殊的方向进行，而自底向上是从特殊向一般方向进行的。自底向上首先定义项目的一些特定任务，然后将这些任务组织起来，形成更高级别的 WBS 层。将详细的任务罗列后，可以形成高的层次，然后再将它们组织成更高的层次。如图 2-11 所示。

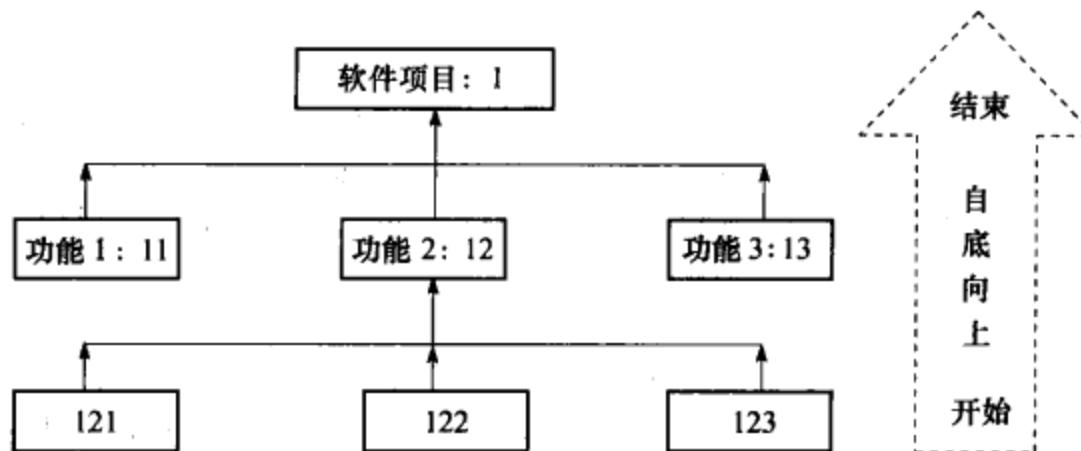


图 2-11 自底向上的方法

采用自底向上开发 WBS 的时候，可以将可能的任务都写在便条上，然后将它们粘在白板上，这样有利于观察和研究任务之间的关系，然后按照逻辑关系层层组合，形成最后的 WBS。

自底向上方法是一种理想的发挥创造力的解决问题方法。试想现在有一个项目团队正设法找到一个廉价的连接北京和上海网络的解决方案。自底向上方法将会设法寻找一个针对该问题的独特方案，而不会对能解决该类问题的每一个方案都进行详细研究。这种方法可能会研究新软件的使用、新服务提供商或者某些实际执行的情况，这些有待执行的工作尚未解决，还有待讨论。

如果对项目人员来说，这个项目是一个崭新的项目，可以采用自底向上方法开发 WBS。

## 2.6 任务分解结果的检验

任务分解后，需要核实分解的正确性：

1) 先明确并识别出项目的各主要组成部分，即明确项目的主要可交付成果。一般来讲，项目的主要组成部分包括项目的可交付成果和项目管理的本身。在进行这一步时需要解答的问题是：要实现项目的目标需要完成哪些主要工作？更低层次的细目是否必要和充分？如果不必要或者不充分，这个组成要素就必须重新修正（增加细目、减少细目或修改细目）。

2) 然后确定每个可交付成果的详细程度是否已经达到了足以编制恰当的成本和历时估算。“恰当”的含义可能会随着项目的进程而发生一定的变化，因为对于将来产生的一项可交付成果进行分解也许是不大可能的。最底层要素是否有重复的？如果存在重复现象就应该重新分解。

3) 确定可交付成果的组成元素。组成元素应当用切实的、可验证的结果来描述，以便于进行绩效测量。组成元素的定义应该根据项目工作实际上是如何组织和完成的。切实、可验证的结果既可包括产品，又可包括服务。这一步要解决的问题是：要完成上述各组成部分，有哪些更具体的工作要做？

每个细目都有明确的、完整的定义吗？如果不是，这种描述需修正或扩充。

4) 核实分解的正确性，需要明确如下问题：

- 最底层项对项目分解来说是否是必需而且充分的呢？如果不是，则必须修改组成元素（添加、删除或重新定义）；
- 每项的定义是否清晰完整？如果不完整，描述则需要修改或扩展；
- 每项是否都能够恰当地编制进度和预算？是否能够分配到接受职责并能够圆满完成这项工作的具体组织单元（例如部门、项目队伍或个人）？如果不能，需要做必要的修改，以便于提供合适的管理控制。

5) 最后与相关人员对 WBS 结果进行评审。

- 是否每个细目可以进行适当的估算？谁能担负起满意地完成这个项目的任务？如果没有，修正正是必要的，为的是提供一个充分的管理控制。

实际操作中对于实际的项目，特别是对于较大的项目而言，在进行任务分解的时候，要注意以下几点：

1) 要清楚地认识到，确定项目的分解结构就是将项目的产品或服务、组织和过程这三种不同的结构综合为项目分解结构的过程。任务分解的规模和数量因项目而异；项目经理和项目的工作人员要善于将项目按照产品或服务的结构进行划分、按照项目的阶段划分以及按照项目组织的责任进行划分等有机地结合起来。

2) 对于项目最底层的工作要非常具体，任务分解结果必须有利于责任分配；而且要完整无缺地分配给项目内外的不同个人或者是组织，以便于明确各个工作块之间的界面，并保证各工作块的负责人能够明确自己的具体任务、努力的目标和所承担的责任。同时，工作如果划分得具体，也便于项目的管理人员对项目的执行情况进行监督和业绩考核。

3) 实际上，进行逐层分解项目或其主要的可交付成果的过程，也就是给项目的组织人员分派各自角色和任务的过程。先分大块任务，然后再细分小的任务，最低层是可控的和可管理的，避免不必要的过细，最好不要超过 7 层。注意收集与项目相关的所有信息；注意参看类似项目的任务分解结果，与相关人员讨论。

4) 对于最底层的工作块，一般要有全面、详细和明确的文字说明，定义任务完成的标准。因为，对于项目，特别是较大的项目来说，或许会有许多的工作块，因此，常常需要把所有的工作块的文字说明汇集到一起，编成一个项目任务分解结构字典（WBS Dictionary）。

5) 并非任务分解结构中所有的分支都必须分解到同一水平，各分支中的组织原则可能会不同。任何分支最底层的细目叫做工作包。按照软件项目的平均规模来说，推荐任务分解时至少拆分到一周的工作量（40 小时）。工作包是完成一项具体工作所要求的一个特定的、可确定的、可交付以及独立的工作单元，需为项目控制提供充分而合适的管理信息。任何项目也并不是只有唯一正确的任务分解结构，例如同一项目按照产品的组成部分或者根据开发过程分解就能作出两种不同的任务分解结构。

根据情况，任务分解中可以包括诸如管理、质量等任务的分解。当然也可以在后续的活动分解时，再分解出相应的管理、质量等活动。

WBS 具体工作要素的阐述通常收集在 WBS 字典（WBS Dictionary）中。一个典型的 WBS 字典，即包括了对工作包的阐述，也包括了对其他规划资料如进度表的日期、成本预算和员工分配等问题的阐述。例如表 2-1 便是一个 WBS 的字典的例子：

表 2-1 WBS 的字典实例

WBS 表示号	BSM-LBL
名称	BSN 事件日志管理系统
主题目标	网管的安全管理系统
描述	1. 存储事件数据：记录相应事件 2. 设置事件滤波：对某些事件可设置滤波 3. 浏览事件日志：对所有事件提供浏览功能 4. 规划 BSN 事件日志 5. 生成历史数据：可生成历史事件报告 6. 管理 BSN 事件日志：可以调整 BSN 事件的配置参数
完成的任务	1, 2, 3 已经完成
责任者	姜岳尊
完成的标识	通过质量保证部的验收报告
备注	

任务分解结构（WBS）提供了项目范围基线，是范围变更的重要输入，范围基线由批准的详细项目范围说明书、WBS 和 WBS 字典。而且有了任务分解，项目经理可以集中注意力到项目的目标上，不必为细枝末节伤脑筋的。同时给开发项目提供了一个实施框架，其中明确了责任，为评估和分配任务提供具体的工作包，是进行估算和编制项目进度的基础，对整个项目成功的集成和控制起到非常重要的作用。

## 2.7 校务通系统案例分析

本项目需求调研阶段，发现用户缺乏相关知识，他们对需求没有很明确的说明，但随着项目的进展，用户的经验也会增加，自然会发现一些不合理或不完整或缺少的需求，必然会引起需求变更。为了避免不必要的需求变化，在开发校务通系统需求的时候，项目组与用户一起来确定需求规格。本项目采用原型分析法确定需求，然后根据用户确认的原型（demo）系统编写软件需求规格。最后，根据需求规格形成项目的最后范围计划，即 WBS 结果。

### 2.7.1 系统原型分析

根据客户的描述，本系统应该提供三个操作平台，即系统管理员平台、教师平台、学生平台，他们希望提供三个平台的统一登录界面，根据不同的角色来进入不同的平台。

- 1) 根据讨论形成主登录界面，如图 2-12 所示。
- 2) 如果用户以教师的角色登录界面，则进入教师平台，界面如图 2-13 所示。

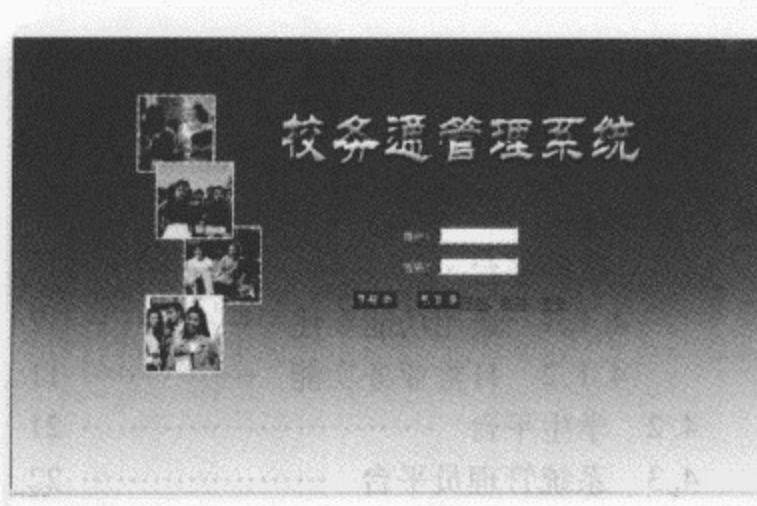


图 2-12 主登录界面



图 2-13 教师登录界面

其界面的内容也是开发人员与客户一同商量的结果，包括界面的风格、颜色等。同时客户提出根据教师的不同身份和权限通过点击“管理功能”可以进入不同的操作界面，如图 2-14 左边的列表显示了这个教师可以进行的操作（权限功能）。这样关于教师的平台基本达成一致了。

- 3) 如果用户以学生的角色登录，则进入学生平台，界面如图 2-15 所示。

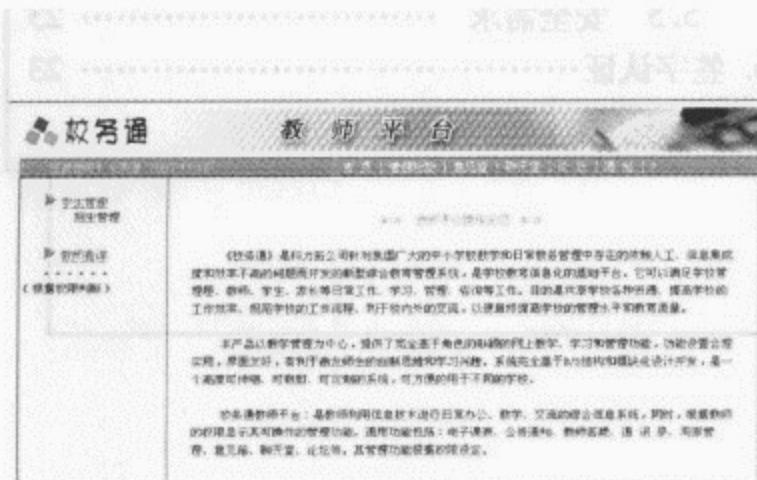


图 2-14 教师操作界面



图 2-15 学生登录界面

由于本系统平台的用户是中小学生，所以客户提出界面最好活泼一些，学生平台的界面经过几翻讨论之后确定如图 2-15 所示的样子。

同时，客户希望学生可以查询到与本人相关的一些信息，所以，通过点击“信息查询”可以进入学生的操作界面，如图 2-16 所示。

左边的列表便是这个学生可以查询的信息。

4) 如果用户以系统管理员的角色登录，则进入系统管理员平台，界面如图 2-17 所示。

系统管理员平台主要是提供系统管理员的操作功能，包括系统的初始化、用户的设置、参数的设置等。这些权限是确定的，如图 2-17 所示。

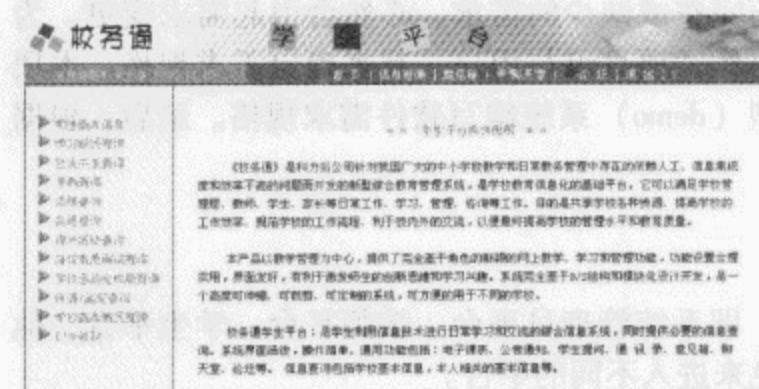


图 2-16 学生操作界面

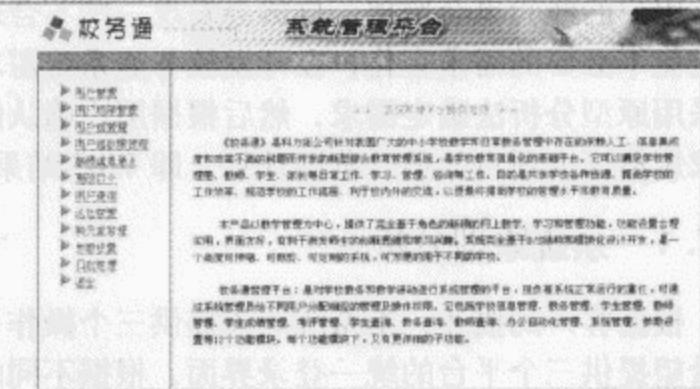


图 2-17 系统管理员界面

## 2.7.2 需求规格说明书

<b>《校务通系统需求规格说明书》</b>	
<b>目 录</b>	
<b>1. 引言</b>	1
1.1 目的	1
1.2 范围	2
1.3 缩写与术语	2
1.4 参考资料	2
1.5 版本更新信息	2
<b>2. 系统定义</b>	2
2.1 项目背景	2
2.2 项目简介	3
<b>3. 应用环境</b>	3
3.1 网络环境	3
3.2 软件环境	4
<b>4. 功能需求</b>	5
<b>4.1 教师平台</b>	5
4.1.1 通用功能描述	7
4.1.2 日常业务功能	11
<b>4.2 学生平台</b>	21
<b>4.3 系统管理员平台</b>	22
<b>5. 性能需求</b>	22
5.1 扩充性	22
5.2 适应性	22
5.3 故障处理	22
5.4 用户界面	23
5.5 安全需求	23
<b>6. 签字认证</b>	23

详细内容略。

### 2.7.3 系统 WBS

根据对本项目的需求规格的分析，采用图表方式进行任务分解，其分解结果如图 2-18 所示，它是按照功能组成标准进行的教师功能部分的任务分解，其中，没有包括管理、质量等相关的任务，WBS 可以随着系统的完善而不断增加和完善的。

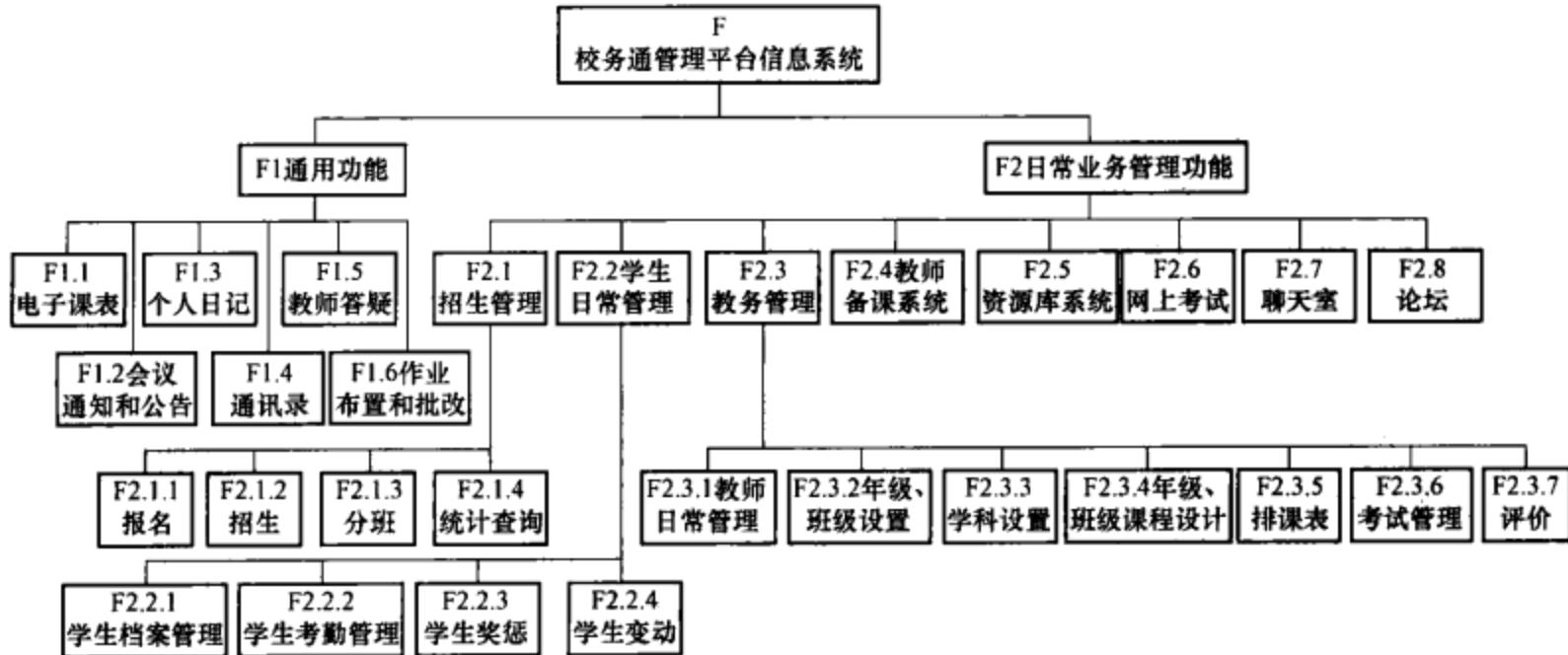


图 2-18 校务通任务分解结果

另外，F2.7（聊天室）和 F2.8（论坛）采用标准的重用技术，F2.5（资源库系统）、F2.4（教师备课系统）、F2.6（网上考试）功能已经有成熟的产品（价格明确），所以这些功能可以不用分解。

Microsoft Project 也是创建 WBS 的好的工具，可以将上面创建好的 WBS 条目加入到 Microsoft Project 中。图 2-19 是采用 Microsoft Project 开发的 WBS。

## 2.8 小结

本章介绍了需求管理的五个过程以及任务分解的过程。需求管理过程包括需求获取、需求分析、需求规格、需求验证、需求变更。软件项目开发人员首先应该明确用户的意图和要求，然后形成一个可以作为开发图纸的软件需求规格。另外，本章重点讲述了软件项目的分解技术，通过任务分解，使项目拆分成很多更小、更易管理、更易操作的细目，目的是提高估算成本、时间和资源的准确性、为绩效测量和控制确定一个基准线、使工作变得更易操作的，责任分工更加明确。任务分解可以采用清单或者图表的形式。分解时采用的标准应该统一。通过任务分解可以界定项目总范围。范围计划主要是对项目的范围进行分析，得出范围说明，提交的文档主要是需求规格说明书（项目范围说明书）以及 WBS。

	WBS	任务名称
1	1	校务通管理系统
2	1.1	软件规划
3	1.1.1	项目规划
4	1.1.2	计划评审
5	1.2	需求开发
6	1.2.1	用户界面设计
7	1.2.2	用户需求评审
8	1.2.3	修改需求、修改用户界面
9	1.2.4	编写需求规格说明书
10	1.2.5	需求验证
11	1.3	设计
12	3.1	概要设计
13	3.2	数据库ER图编制、建库
14	1.3.3	设计评审
15	1.4	实施
16	1.4.1	通用功能-增量1
17	1.4.1.1	电子课表
18	1.4.1.2	会议通知和公告
19	1.4.1.3	个人日记
20	1.4.1.4	通讯录
21	1.4.1.5	教师答疑
22	1.4.1.6	作业布置和批改
23	1.4.1.7	增量1-评审
24	1.4.2	招生管理-增量2
30	1.4.3	学生日常管理-增量3
36	1.4.4	教务管理-增量4
45	1.4.5	教师辅助功能-增量5
50	1.4.6	聊天室/论坛-增量6
54	1.5	系统集成
55	1.5.1	系统集成测试
56	1.5.2	环境测试
57	1.6	提交

图 2-19 校务通清单式 WBS

## 2.9 练习题

### 一、选择题

1. 需求分析是回答系统必须（ ）的问题。
 

A. 做什么	B. 怎么做
C. 何时做	D. 为谁做
2. WBS(任务分解结构) 非常重要, 因为下列原因, 除了（ ）。
 

A. 帮助组织工作	B. 防止遗漏工作
C. 为项目估算提供依据	D. 确定团队成员责任
3. 项目范围（ ）。
 

A. 只在项目开始时重要	B. 在授权项目的合同或者其他文件得以批准后就不再重要了
C. 从项目概念阶段到收尾阶段都应该加以管理和控制	D. 是在项目执行阶段通过变更控制步骤进行处理的问题
4. 为了有效地管理项目, 应该将工作分解为更小的部分, 以下各项中, 哪一项不能说明任务应该分解到什么程度? ( )
 

A. 可以在 80 小时内完成	B. 不能再进一步进行逻辑细分了
C. 可由一个人完成	D. 可以进行实际估算
5. 范围变更是指（ ）。
 

A. 修改技术规格	B. 对范围陈述进行修订
C. 对批准后的 WBS 进行修改	D. 以上都不是
6. 下面哪个不是需求管理的过程（ ）。
 

A. 需求设计	B. 需求获取	C. 需求分析	D. 需求变更
---------	---------	---------	---------
7. 下面哪个不是创建 WBS 的方法? ( )
 

A. 自顶向下	B. 自底向上	C. 控制方法	D. 模板参照
---------	---------	---------	---------
8. 任务分解可以（ ），它是范围变更的一项重要输入。
 

A. 提供项目成本估算结果	B. 提供项目范围基线
C. 规定项目采用的过程	D. 提供项目的关键路径
9. 范围基线由（ ）组成。
 

A. 项目章程、批准的详细的项目范围说明书和 WBS	B. 批准的详细项目范围说明书、WBS 和 WBS 字典
C. 项目章程、项目工作说明书和 WBS	D. WBS

### 二、判断题

1. 需求分析过程是确定项目如何实现的过程, 并确定项目采用的技术方案。( )
2. 对于以前没有做过的项目, 开发 WBS 时, 可以采用自底向上的方法。( )

## ■ 软件项目进度计划

进度管理是软件项目管理中最重要的部分之一。进度管理的主要目标是：最短时间、最少成本、最小风险，即在给定的限制条件下，用最短时间、最少成本，以最小风险完成项目工作。作为一个好的项目管理者首先应该是一个好的时间管理者，编制项目的进度计划是其重要任务。进度管理是核心管理三角形的另外一边，本章进入本书路线图的时间计划，见图 3-1。

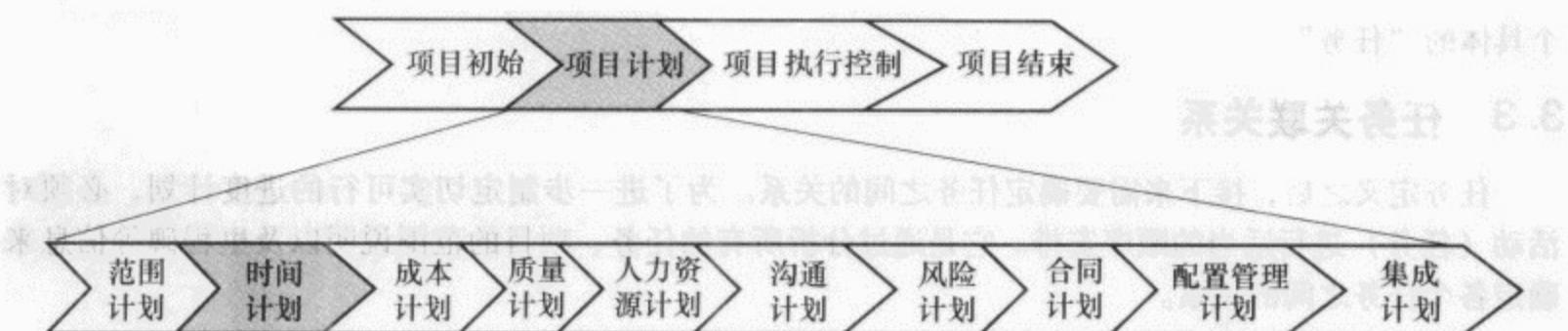


图 3-1 路线图——项目计划 - 时间计划

### 3.1 关于进度估算

一般地说，项目的初期要对项目的规模、成本和进度进行估算，而且基本上是同时进行的，项目的规模和进度估算有一定的关系。进度估算从时间的角度对项目进行规划，而成本估算则是从费用的角度对项目进行规划。这里的费用应理解为一个抽象概念，它可以是工时、材料或人员等。其实，时间计划和成本计划都是估算的过程。在项目的进行过程中，会有更多新的信息，可能需要不断地重复进行估算。在项目的不同阶段可以采用不同的估算方法，开始估算的结果可能误差比较大，随着项目的进展，会逐步的精确。本章讲述进度估算，下一章讲述成本估算。

时间是一种特殊的资源，以其单向性、不可重复性、不可替代性而有别于其他资源。如项目的资金不够，还可以贷款，可以集资，即借用别人的资金；但如果项目的时间不够，就无处可借。时间也不像其他资源有可加合性。

对于一个项目管理者，应该定义所有的项目任务，识别出关键任务，跟踪关键任务的进展情况，同时，能够及时发现拖延进度的情况。为此，项目管理者必须制定一个足够详细的进度表，以便监督项目进度并控制整个项目。

PMI 强调：作为零售摊位成功的关键要素是位置、位置、位置，而项目成功的三大要素是计划、计划、计划。计划是通向项目成功的路线图，而进度计划是项目计划中最重要的部分，是项目计划的核心。本章讲述项目进度计划的主要过程：首先根据任务分解的结果（WBS）再进一步分解出主要的任务（活动），确立任务（活动）之间的关联关系，然后估算出每个任务（活动）需要的资源、历时，最后编制出项目的进度计划。

交付期作为软件开发合同或者软件开发项目中的时间要素，是软件开发能否获得成功的重要判断标准之一，是最核心的关注范围。软件项目管理的主要目标就是提升质量、降低成本、保证交付期，以及追求顾客满意。交付期意味着软件开发在时间上的限制，意味着软件开发的最终速度，也意味着满足交付期带来的预期收益和捍卫交付期需要付出的代价。交付期体现在进度计划中。目前，软件项目的进度是企业普遍最重视的项目要素，原因有很多，例如与客户的协约中或者项目计划中，客户最关心的是进度，最明确的也是进度；进度是项目各要素中最容易度量的，因为容易度量，所以在许多

企业或者企业的领导人看来当然是最理想的管理考核指标。

## 3.2 任务定义

进度是对执行的活动和里程碑制定的工作计划日期表，它决定是否达到预期目的，是跟踪和沟通项目进展状态的依据，也是跟踪变更对项目影响的依据。按时完成项目是对项目经理最大的挑战，因为时间是项目规划中灵活性最小的因素，进度问题又是项目冲突的主要原因，尤其在项目的后期。为了编制进度，首先需要定义任务。

任务定义是一个过程，它涉及确认和描述一些特定的活动，完成了这些活动意味着完成了 WBS 结构中的项目细目和子细目。通过任务（活动）定义这一过程可使项目目标体现出来。WBS 是面向可提交物的，WBS 的每个工作包需要被划分成所需要的任务，任务定义是面向活动的，是对 WBS 做进一步分解的结果，以便清楚应该完成的每个具体任务或者提交物应该执行的活动。我们称“活动”为一个具体的“任务”。

## 3.3 任务关联关系

任务定义之后，接下来需要确定任务之间的关系。为了进一步制定切实可行的进度计划，必须对活动（任务）进行适当的顺序安排。它是通过分析所有的任务、项目的范围说明以及里程碑等信息来确定各个任务之间的关系。

### 3.3.1 任务之间的关系

项目各项任务（活动）之间存在相互联系与相互依赖关系，根据这些关系安排各项活动的先后顺序。活动排序过程包括确认并编制活动间的相关性。活动必须被正确地加以排序以便今后制订现实的、可行的进度计划。排序可由计算机执行（利用计算机软件）或用手工排序。对于小型项目手工排序很方便，对大型项目的早期（此时项目细节了解甚少）用手工排序也是方便的，手工编制和计算机排序可以结合使用。任务之间的关系主要有如下 4 种情况，如图 3-2 所示。

其中：

结束→开始（Finish-to-Start, FS）：表示 A 任务（活动）在 B 任务（活动）开始前结束。

开始→开始（Start-to-Start, SS）：表示 A 任务（活动）开始，B 任务（活动）才可以开始。

结束→结束（Finish-to-Finish, FF）：表示 A 任务（活动）结束，B 任务（活动）才可以结束。

开始→结束（Start-to-Finish, SF）：表示 A 任务（活动）开始，B 任务（活动）才能结束。

结束→开始是最常见的逻辑关系，开始→结束关系极少使用（也许只有职业进度计划工程师使用）。

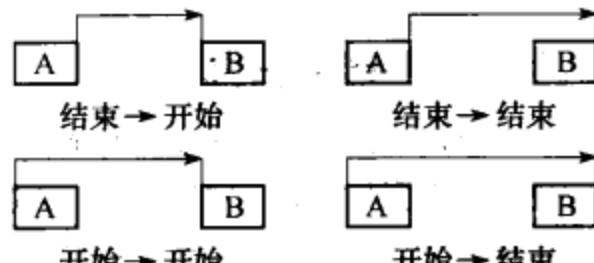


图 3-2 项目各活动（任务）之间的关系

### 3.3.2 任务关联关系的依据

确定任务（活动）之间关联关系的依据主要有以下几种。

1) 强制性依赖关系（mandatory or hard）。强制性依赖关系是工作任务中固有的依赖关系，是一种不可违背的逻辑关系，又称硬逻辑关系，它是因为客观规律和物质条件的限制造成的。有时，也称为内在的相关性。例如需求分析一定要在软件设计之前完成，测试活动一定是在编码任务之后执行。

2) 软逻辑关系（discretionary）。软逻辑是由项目管理人员确定的项目活动之间的关系，是人为的、主观的，是一种根据主观意志去调整和确定的项目活动的关系，也可称指定性相关，或者偏好相关或软相关。例如，安排计划的时候，哪个模块先做，哪个模块后做，哪个任务先做好一些，哪些任务同时做好一些，都可以由项目经理确定。

3) 外部依赖关系（external）。外部依赖是项目活动与非项目活动之间的依赖关系，例如环境测试

依赖于外部提供的环境设备等。

### 3.4 进度管理图示

软件项目进度管理的图示有很多，例如甘特图、网络图、里程碑图、资源图等。下面分别进行介绍。

#### 3.4.1 甘特图

甘特图（Gantt 图）具有历史悠久、直观简明、容易学习、容易绘制等优点。甘特图可以显示任务的基本信息，使用甘特图能方便地查看任务的工期、开始和结束时间以及资源的信息。甘特图有两种表示方法，这两种方法都是将任务（工作）分解结构中的任务排列在垂直轴，而水平轴表示时间。一种是棒状图（bar chart），用棒状图表示任务的起止时间，见图 3-3。空心棒状图表示计划起止时间，实心棒状图表示实际起止时间。用棒状图表示任务进度时，一个任务需要占用两行的空间表示。另外一种表示甘特图的方式，见图 3-4，是用三角形表示特定日期，方向向上三角形表示开始时间，向下三角形表示结束时间，计划时间和实际时间分别用空心三角和实心三角表示。一个任务只需要占用一行的空间。

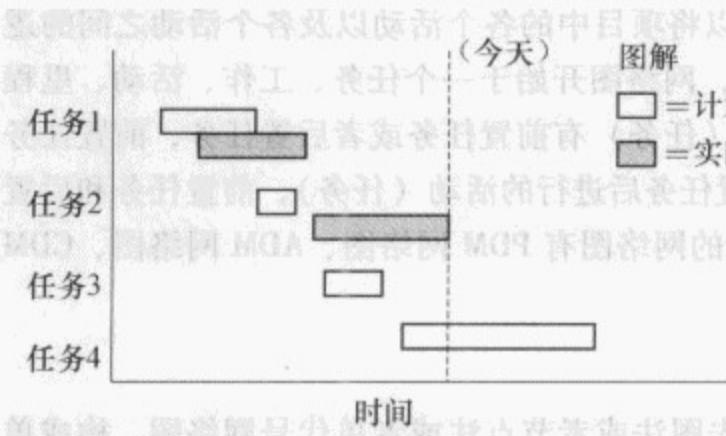


图 3-3 棒状图甘特图

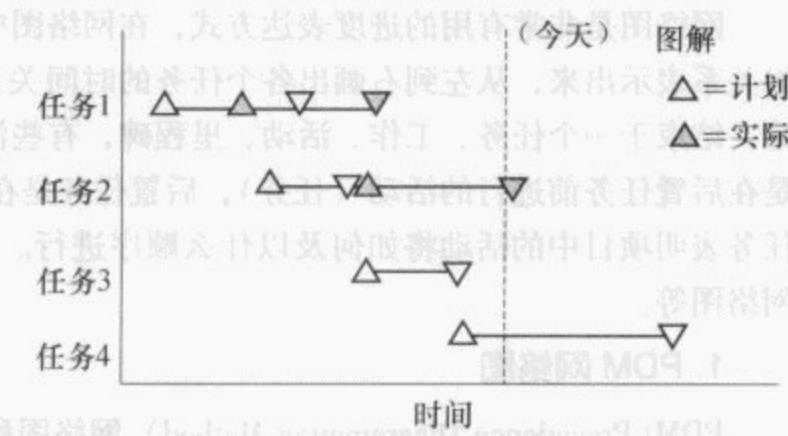


图 3-4 三角形甘特图

这两个图示说明同样的问题，从图可以看出所有的任务的起止时间都比计划推迟了，而且任务 2 的历时长度也比计划的长很多。

利用甘特图可以很方便地进行项目计划和项目计划控制，由于其简单易用而且容易理解，所以，被广泛地应用到项目管理中，尤其被软件项目管理所普遍使用。例如图 3-5 是用工具生成的一个软件项目的甘特图。



图 3-5 项目进度图

甘特图也有缺点，它不能明显地表示各项任务彼此间的依赖关系，也不能明显地表示关键路径和关键任务，进度计划中的关键部分不明确。因此，在管理大型软件项目时，仅用甘特图是不够的。而网络图可以反映任务的起止日期变化对整个项目的影响。

### 3.4.2 网络图

网络图（network diagramming）是活动排序的一个输出，它展示项目中的各个活动以及活动之间的逻辑关系，表明项目任务将如何和以什么顺序进行。进行历时估计时，网络图可以表明项目将需要多长时间完成，当改变某项活动历时时，网络图表明项目历时将如何变化。

网络图不仅能描绘任务分解情况及每项作业的开始时间和结束时间，而且还能清楚地表示各个作业彼此间的依赖关系。网络图中容易识别出关键路径和关键任务。因此，网络图是制定进度计划的强有力的工具。通常联合使用甘特图和网络图这两种工具来制定和管理进度计划，使它们互相补充，取长补短。

网络图是非常有用的进度表达方式，在网络图中可以将项目中的各个活动以及各个活动之间的逻辑关系表示出来，从左到右画出各个任务的时间关系图，网络图开始于一个任务、工作、活动、里程碑，结束于一个任务、工作、活动、里程碑，有些活动（任务）有前置任务或者后置任务，前置任务是在后置任务前进行的活动（任务），后置任务是在前置任务后进行的活动（任务），前置任务和后置任务表明项目中的活动将如何及以什么顺序进行。常用的网络图有 PDM 网络图、ADM 网络图、CDM 网络图等。

#### 1. PDM 网络图

PDM(Precedence Diagramming Method) 网络图称优先图法或者节点法或者单代号网络图。构成单代号网络图的基本特点是节点，节点表示任务（活动），用箭线表示各任务（活动）之间的逻辑关系，见图 3-6。活动 1 是活动 3 的前置任务，活动 3 是活动 1 的后置任务。

PDM 网络图是目前比较流行的网络图，图 3-7 是一个软件项目的 PDM 网络图的实例。

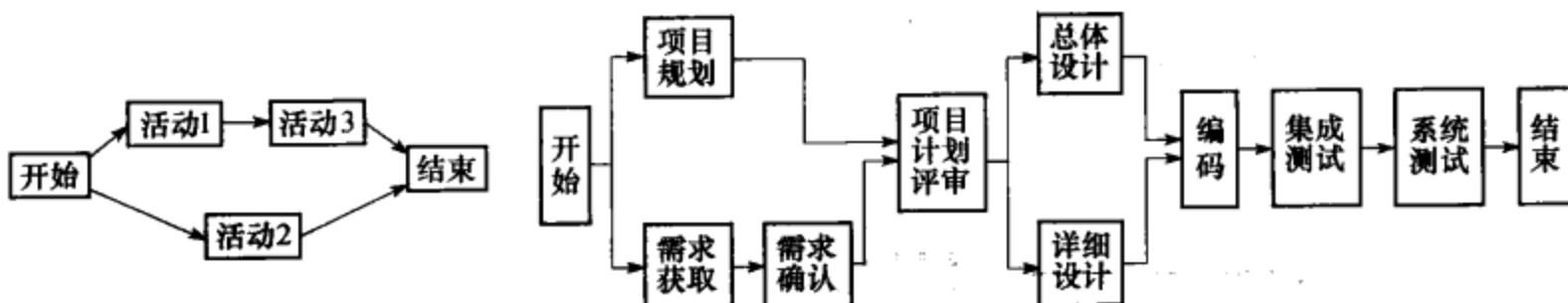


图 3-6 PDM 网络图的关系

图 3-7 软件项目的 PDM 图

#### 2. ADM 网络图

ADM(Arrow Diagramming Method) 网络图也称为箭线法或者双代号网络图。在双代号网络图中，箭线表示活动（任务），节点表示前一个任务的结束，同时也表示后一个任务的开始。将图 3-7 的项目改用 ADM 网络图表示，如图 3-8。这里的双代号表示网络图中两个代号唯一确定一个任务，例如代号 1 和代号 3 确定“项目规划”任务，代号 3 和代号 4 确定“计划评审”任务。

在 ADM 网络图中，有时为了表示逻辑关系，需要设置一个虚活动，虚活动是不需要时间和资源的，一般用虚箭线表示，图 3-9 中的任务 A 和任务 B 表示的 ADM 网络图是不正确的，因为 ADM 图中，代号 1

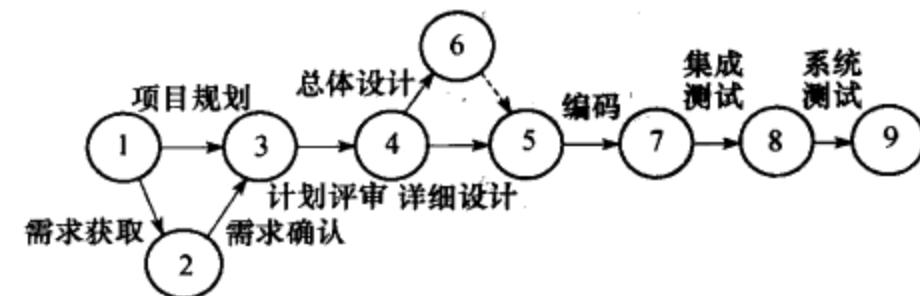


图 3-8 软件项目的 ADM 图

和代号 2 只能确定一个任务，或者任务 A 或者任务 B。为了解决这个问题，需要引入虚活动，见图 3-10，为了表示活动 A、B 的逻辑关系需要引入代号 3，用虚线连接代号 2 到代号 3 的活动就是一个虚活动。它不是一个实际的活动，只是为了表达逻辑关系引入的。图 3-8 中的代号 6 和代号 5 之间的虚线也代表一个虚活动。

### 3. CDM 网络图

CDM (Conditional Diagramming Method) 网络图也称为条件箭线图法。它允许活动序列相互循环与反馈，诸如一个环（例如某试验须重复多次）或条件分支（例如：一旦检查中发现错误，设计就要修改），从而在绘制网络图的过程中会形成许多条件分支，而在 PDM、ADM 中是绝对不允许的。这种网络图在实际项目中使用的很少。

### 3.4.3 里程碑图

里程碑图是由一系列的里程碑事件组成的，所谓“里程碑事件”，往往是一个时间要求为零的任务，就是说它并非是一个要实实在在完成的任务，而是一个标志性的事件，例如在软件开发项目中的“测试”是一个子任务，“撰写测试报告”也是一个子任务，但“完成测试报告”可能就不能成为一个实实在在需要完成的子任务了，但在制定计划以及跟踪计划的时候，往往加上“完成测试报告”这一个子任务，但工期往往设置为“0 工作日”，目的就在于检查这个时间点，这是“测试”整个任务的结束的标志。

里程碑图显示项目进展中的重大工作完成，里程碑不同于活动，活动是需要消耗资源的并且需要花时间来完成，里程碑仅仅表示事件的标记，不消耗资源和时间的。例如：图 3-11 是一个项目的里程碑图，从图可以知道设计在 2003-4-10 完成，测试在 2003-5-30 完成。里程碑图表示了项目管理的环境，对项目干系人是非常重要的，它表示了项目进展过程中的几个重要的点。

项目计划以里程碑为界限，将整个开发周期划分为若干阶段。根据里程碑的完成情况，适当的调整每一个较小的阶段的任务量和完成的任务时间，这种方式非常有利于整个项目计划的动态调整。

对项目里程碑阶段点的设置必须符合实际，它必须有明确的内容并且通过努力能达到，要具有挑战性和可达性，只有这样才能在抵达里程碑时，使开发人员产生喜悦感和成就感，激发大家向下一个里程碑前进。实践表明：未达到项目里程碑的挫败感将严重地影响开发的效率，不能达到里程碑可能是里程碑的设置不切实际造成的。进度管理与控制其实就是确保项目里程碑的达到，因此里程碑的设置要尽量符合实际，并且不轻易改变里程碑的时间。

### 3.4.4 资源图

资源图可以用来显示项目进展过程中资源的分配情况，这个资源包括人力资源、设备资源等。图 3-12 就是一个人力资源随时间分布情况的资源图。

### 3.5 任务资源估计

在估计每个任务的历时之前，首先应该对每个任务需要的资源类型和数量有一定的考虑，这些资源包括人

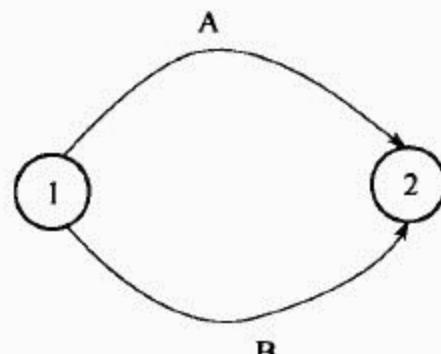


图 3-9 不正确的 ADM 图

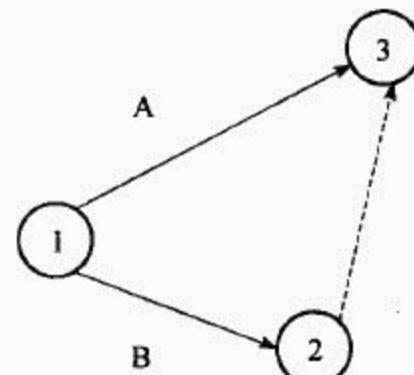


图 3-10 有虚活动的 ADM 图

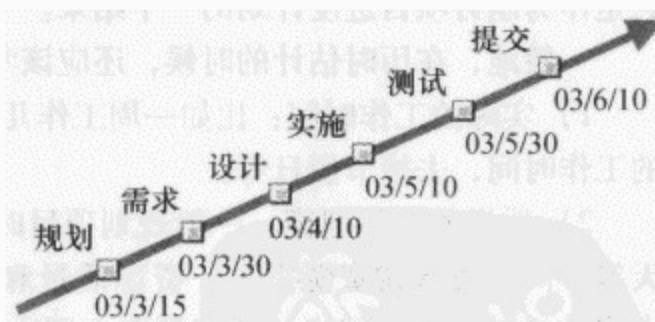


图 3-11 里程碑图

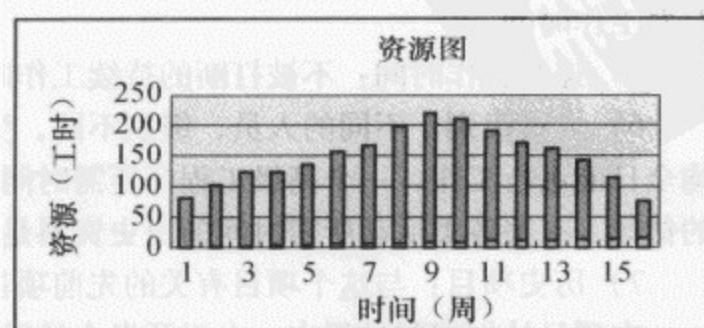


图 3-12 人力资源图

力资源、设备资源，以及其他资源等。对于项目管理者来说应该回答下面的问题：

- 1) 对于特定的任务它的难度如何？
- 2) 是否有唯一的特性影响资源的分配？
- 3) 企业以往类似项目的状况？个人的成本如何？
- 4) 企业现在是否有完成项目合适的资源——人、设备、资料等？企业的政策是否能够影响这些合适的资源。
- 5) 是否需要更多的资源来完成这个项目，是否需要外包等？

为了准确估计任务需要的资源，项目的任务（活动）列表，任务（活动）的属性，历史项目计划，企业的环境因素，企业的过程制度，可用资源状况等信息是需要的。

我们可以采用专家估算方法或者找有类似项目经验的人来辅助估算，也可以采用脑力风暴方法，评估相关选项，由于人力资源是软件项目最主要的成本，所以在项目的早期，应该从不同渠道来获取相关的信息。当然这个结果也会修改和完善。

### 3.6 项目历时估计

定义了项目中的任务（活动），任务之间的关系，估计了需要的资源，接下来就需要估计任务的历时，即花费的时间。

项目历时估计是估计任务的持续时间，它是项目计划的基础工作，直接关系到整个项目所需总时间。估计太长或太短对整个项目都是不利的。项目历时估计首先是对项目中的任务（活动）进行时间估计，然后是确定项目的历时估计。任务（活动）时间估计指预计完成各任务（活动）所需时间长短，在项目团队中熟悉该任务（活动）特性的个人和小组可对活动所需时间作出估计。估计完成某活动所需时间长短要考虑该活动“持续”所需时间。例如，如果软件系统设计需要2~4个工作日，到底是几天取决于：a) 活动的开始日期是星期几？b) 周末是否算工作日？c) 参加设计的人数等等。绝大多数的计算机排序软件会自动处理这类问题。整个项目所需时间也可以运用这些工具和方法加以估算，它是作为制订项目进度计划的一个结果。

一般地，在历时估计的时候，还应该考虑如下信息：

- 1) 实际的工作时间：比如一周工作几天，一天工作几个小时等；正常工作时间，要充分考虑正常的工作时间，去掉节假日等。
- 2) 项目的人员规模：一般规划项目时，应该按照人员完成时间来考虑，比如多少人月，多少人天等。同时要考虑资源需求、资源质量和历史资料等等。资源数量的多少也决定活动的历时估计，大多数活动所需时间由相关资源多少所决定。例如，二人一起工作完成某设计活动只需一半的时间（相对一个人单独工作所需时间）。然而每日只能用半天进行工作的人通常至少需要二倍的时间完成某活动（相对一个人能整天工作的所需时间）。大多数活动所需时间与人和材料的能力（质量）有关。
- 3) 生产率：根据人员的技能考虑完成软件项目的生产率，比如LOC/天等。
- 4) 有效工作时间：在正常的工作时间内，去掉聊天、打电话、去卫生间、抽烟、休息等时间后的有效工作时间。
- 5) 连续工作时间：不被打断的持续工作时间。
- 6) 人员级别：不同的人员，级别不同，生产率不同，成本也不同的，对同一活动，假设有两个人均全日能进行工作，一个高级工程师所需时间少于初级工程师所需时间。这就是资源质量也影响活动的估计，有关各类活动所需时间的历史资料是有用的。
- 7) 历史项目：与这个项目有关的先前项目结果的记录，可以帮助项目进行时间估计。

在项目计划编制过程中，由于开发人员需要休息、吃饭、开会等，可能不会将所有的时间放在项目开发工作上，而且这还不考虑到开发人员的工作效率是否保持在一恒定水平上。其实，一天8小时

工时制并不是花在项目上的时间就是8小时。在实际开发中，开发员工的时间利用率能够达到80%就已经很好了。

历时估计这个数值应该是有效工作时间加上额外的时间（elapsed time），历时估计的输出是各个活动的时间估计，即关于完成一活动需多少时间的数量估计，其估计值可以用一个唯一的值，也可以用某一范围表示，例如：任务的历时是10天，或者8天到12天，或者2周±2天，表示该活动至少需8天和不超过12天。

下面简单介绍几种软件项目历时估计常用的估算方法。

### 3.6.1 定额估算法

定额估算法是比较基本的估算项目历时的方法，公式为：

$$T = Q / (R \times S)$$

其中： $T$  为活动的持续时间，可以用小时、日、周等表示； $Q$  为活动的工作量，可以用人月、人天等单位表示； $R$  为人力或设备的数量，可以用人或设备数等表示； $S$  为开发（生产）效率，以单位时间完成的工作量表示。

此方法比较简单，而且容易计算。但是，此方法比较适合规模较小的项目，它有一定的局限性，没有考虑任务之间的关系。

例如：一个软件项目的规模估算为  $Q=6$  人月，如果有2个开发人员即  $R=2$  人，而每个开发人员的开发效率是  $S=1$ （即正常情况下）， $T=6/(2 \times 1)=3$  月，即这个项目需要3个月完成；如果  $S=1.5$ ，则时间进度估算结果是  $T=6/(2 \times 1.5)=2$  月，即这个项目需要2个月完成。

### 3.6.2 经验导出模型

经验导出模型是根据大量项目数据统计而得出的模型，不同的数据导出的模型略有不同，整体的经验导出模型为  $D = a \times E^b$ ，其中： $D$  表示月进度， $E$  表示人月工作量， $a$  是2~4之间的参数， $b$  为1/3左右的参数，它们是依赖于项目自然属性的参数。

例如 Walston-Felix 的模型为  $D = 2.4 \times E^{0.35}$ 。基本 COCOMO 的模型为  $D = 2.5 \times E^b$ ，其中  $b$  是0.32~0.38之间的参数。

经验导出模型可以根据项目的具体情况选择合适的参数。例如，一个项目的规模估计是  $E=65$  人月，如果模型中的参数  $a=3$ ， $b=1/3$ ，则  $D=3 \times 65^{1/3}=12$  月，即65人月的软件规模，估计需要12个月完成。

### 3.6.3 工程评估评审技术

工程评估评审技术（Program Evaluation and Review Technique，PERT）最初发展于1958年，是用来适应大型工程年代的需要，当时由于美国海军专门项目处关心大型军事项目的发展计划，于1958年将PERT引入到它的海军北极星导弹开发项目中，取得不错的效果。它是利用网络顺序图的逻辑关系和加权历时估算来计算项目历时的，当估计历时存在不确定性，可以采用PERT方法，即估计具有一定的风险时采用。PERT方法采用加权平均的算法是  $(O + 4M + P)/6$ ，其中， $O$  是活动（项目）完成的最小估算值，或者说是最乐观值（optimistic time）； $P$  是活动（项目）完成的最大估算值，或者说是最悲观值（pessimistic time）； $M$  是活动（项目）完成的最大可能估算值（most likely time）。最乐观值是基于最好的情况的估计，最悲观值是基于最差的情况的估计，最大可能估算值是基于最大可能情况的估计或者是基于最期望的情况的估计。

图3-13的网络图中，估计A、B、C的任务历时存在很大不确定性，故采用PERT方法估计任务历时，图3-13中标识了任务A、B、C的最乐观、最悲观和最可能的历时估计，根据PERT公式  $(O + 4M + P)/6$ ，计算各个任务的历时估计结果，见表3-1。

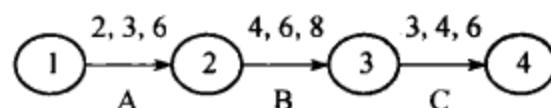


图3-13 ADM网络图

一个路径上的所有活动（任务）的历时估计之和便是这个路径的历时估计，其值称为路径长度。图 3-13 中的路径长度为 13.5，即这个项目总的时间估计是 13.5，见表 3-1。

用 PERT 方法估计历时存在一定的风险，因此有必要进一步给出风险分析结果。为此引入了标准差（standard deviation）和方差（variance）的概念。

- 标准差  $\delta = (P - O)/6$
- 方差  $\delta^2 = [(P - O)/6]^2$

其中， $O$  是最乐观的估计， $P$  是最悲观的估计。

接下来，我们需要估计网络图中一条路径的历时情况，如果一个路径中每个活动的 PERT 历时估计为  $E_1, E_2, \dots, E_n$ ，标准差分别为  $\delta_1, \delta_2, \dots, \delta_n$ ，则这个路径的历时、标准差、方差分别为：

- $E = E_1 + E_2 + \dots + E_n$
- $\delta^2 = (\delta_1)^2 + (\delta_2)^2 + \dots + (\delta_n)^2$
- $\delta = ((\delta_1)^2 + (\delta_2)^2 + \dots + (\delta_n)^2)^{1/2}$

方差是路径上每个活动方差之和，即  $\delta^2 = (\delta_1)^2 + (\delta_2)^2 + \dots + (\delta_n)^2$ ，这个路径的标准差  $\delta = ((\delta_1)^2 + (\delta_2)^2 + \dots + (\delta_n)^2)^{1/2}$ 。图 3-13 中任务 A、B、C 的标准差和方差以及这个路径的标准差和方差，见表 3-2。

根据概率理论，对于遵循正态概率分布的均值  $E$  而言， $E \pm 1\delta$  的概率分布是 68.3%， $E \pm 2\delta$  的概率分布是 95.5%， $E \pm 3\delta$  的概率分布是 99.7%，如图 3-14。

图 3-13 项目的 PERT 总历时估计是 13.5 天，标准差  $\delta = 1.07$ 。所以这个项目总历时估计的概率见表 3-3。即项目在 12.43 天到 14.57 天内完成的概率是 68.3%，项目在 11.36 天到 15.64 天内完成的概率是 95.5%，项目在 10.29 天到 16.71 天内完成的概率是 99.7%。

表 3-1 PERT 方法估计项目历时

任务 \ 估计值	最乐观值	最可能值	最悲观值	PERT 估计值
A	2	3	6	3.33
B	4	6	8	6
C	3	4	6	4.17
项目				13.5

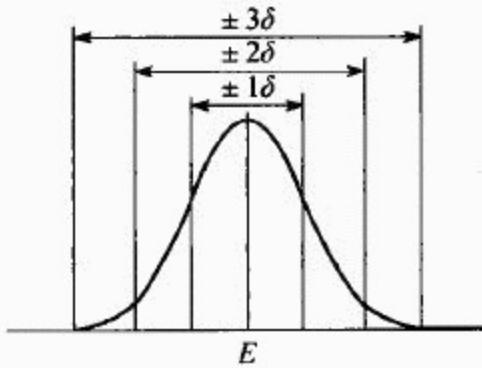


图 3-14 正态概率分布

表 3-2 项目的基本差和方差

项 \ 值	标准差	方差
任务 A	4/6	16/36
任务 B	4/6	16/36
任务 C	3/6	9/36
项目路径	1.07	41/36

表 3-3 项目完成的概率分布

历时估计 $E = 13.5, \delta = 1.07$			
范 围	概 率	从	到
T1	$\pm 1\delta$	68.3%	12.43
T2	$\pm 2\delta$	95.5%	11.36
T3	$\pm 3\delta$	99.7%	10.29
			14.57
			15.64
			16.71

**【例 1】** 图 3-13 所示项目在 14.57 天内完成的概率是多少？

由于  $14.57 = 13.5 + 1.07 = E + \delta$ ，如图 3-15 所示，所以项目在 14.57 天内完成的概率是箭头 1 以左的概率，很显然它等于箭头 2 以左的概率加上  $68.2/2$ ，即 84.2%，所以项目在 14.57 天内完成的概率是 84.2%，即接近于 85%。

### 3.6.4 专家估计方法

估计项目所需时间经常是困难

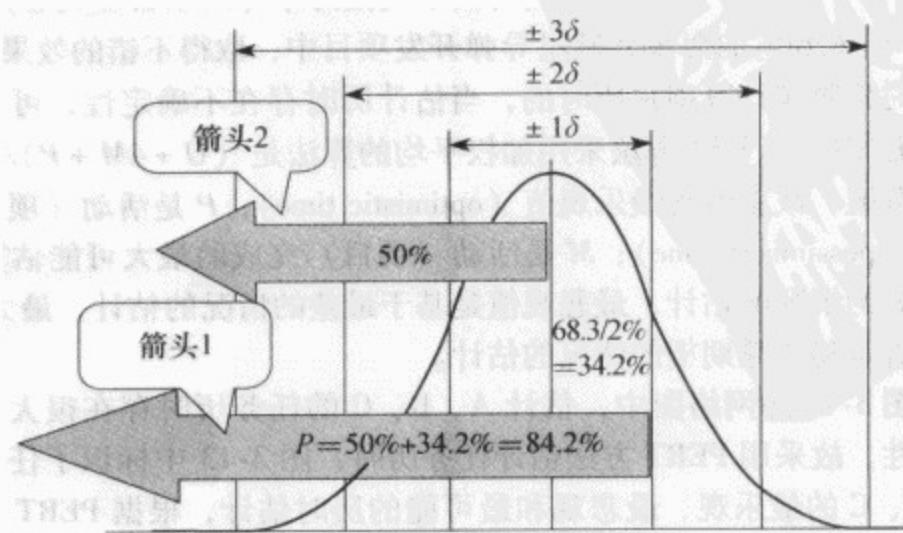


图 3-15 项目在 14.57 天内完成的概率

的，因为许多因素会影响项目所需时间（例如，资源质量的高低，劳动生产率的不同）。专家估计法是通过专家依靠过去资料信息进行判断，以估算进度的方法。如果找不到合适专家，估计结果往往不可靠和具有较大风险。

### 3.6.5 类推估计方法

类推估计意味着利用一个先前类似活动的实际时间作为估计未来活动时间的基础，这种方法常用于项目早期，掌握的项目信息不多。类推估计是专家判断的一种形式，以下情况的类推估计是可靠的：1) 先前活动和当前活动是本质上类似而不仅仅是表面的相似；2) 专家有所需专长。对于软件项目，利用企业的历史数据进行历时估计是常见的方法。

### 3.6.6 模拟估计方法

模拟（simulation）是用不同的假设试验一些情形，以便计算相应的时间，最常见方法是蒙特卡罗分析技术（Monte Carlo analysis）。在这种方法中，假设了各活动所用时间的概率分布以用来计算整个项目完成所需时间的概率分布。让计算机多次进行一个项目的模拟，就可以得出一个可能结果的范围和每一结果的概率。蒙特卡罗分析的结果可能比 PERT 和 CPM 方法结果更加悲观的，原因是采用了最悲观的情形进行分析的。

### 3.6.7 基于承诺的进度估计方法

基于承诺的进度估算方法是从需求出发去安排进度，不进行中间的工作量（规模）估计，通过开发人员做出的进度承诺而进行的进度估计，它本质上不是进度估算。其优点是有利于开发者对进度的关注，有利于开发者在接受承诺之后的士气高昂。但是缺点是开发人员估计存在一定的误差。

### 3.6.8 Jones 的一阶估计准则

Jones 的一阶估计准则是根据项目功能点的总和，从幂次表（见表 3-4）中选择合适的幂次将它升幂。例如，如果一个软件项目的功能点是  $FP = 350$ ，而且承担这个项目是平均水平的商业软件公司，则粗略的进度估算为  $= 350 \exp(0.43) = 12$  月。

表 3-4 一阶幂次表

软件类型	最优级	平均	最差级
系统软件	0.43	0.45	0.48
商业软件	0.41	0.43	0.46
封装商品软件	0.39	0.42	0.45

### 3.6.9 管理预留

管理预留是一项加在项目末端的人为任务，不是加在每一个任务间隔上，即给项目增加一个储备时间。给储备所分配的时间一般是项目中完成所有任务需要时间的 10% ~ 15%。当一项任务超出了分配的时间，超出的部分使用关键路径末端的管理预留。

管理预留使得项目经理用百分比来察看项目的进展情况。例如，如果一个项目完成了 40%，却用去了 65% 的管理预留，那么如果余下的任务从此以后仍然保持这个趋势，这个项目就会陷入困境。

增加管理预留，而不是增加每项任务的历时，因为帕肯森定律（Parkinson's law）指出工作总是拖延到它所能够允许最迟完成的那一天。也就是说如果工作允许它拖延、推迟完成的话，往往这个工作总是推迟到它能够最迟完成的那一刻，很少有提前完成的。也就是说，如果说一项任务需要花费 10 小时完成的话，可能你自己知道只需要 6 小时就可以完成，但却不可思议的花费了 10 小时。这样的话，他们会不珍惜时间，可能会找一些其他事情来做或者简单的等待，直到预留时间花完，才开始正常的项目工作以期待将项目成功完成。

在每一项任务的产生过程中，毫无疑问你会受到夸大任务完成所估算的时间的诱惑。不要受到这种诱惑的影响。应该总是反映任务完成所需要的准确时间。如果愿意，你可以使用 PERT 方法中的最乐观值、最悲观值和最可能值来估算。但不要因为考虑了消极的因素（例如出现错误、返工和活动延

迟等)而增加每一项任务所需要的时间。所以管理预留是将每一项任务的预留时间累加在一起放在关键路径末端,而不要增加每一项任务时间。

### 3.7 进度计划编制

进度计划编制是决定项目活动的开始和结束日期的过程,若开始和结束日期是不现实的,项目不可能按计划完成。编排进度计划时,如果资源分配没有被确定,决定项目活动的开始和结束日期仍是初步的,资源分配可行性的确认应在项目计划编制完成前做好。其实,编制计划的时候,成本估计、时间估计、进度编制等过程常常是交织在一起,这些过程反复多次,最后才能确定项目进度计划。一个进度计划是整个项目计划的一部分。

其中,进度计划编制的输入有项目网络图、活动历时估计、资源需求、资源库描述(对进度编制而言,有关什么资源,在什么时候,以何种方法可供利用是必须知道的)、日历表、超前与滞后、约束和假设(例如:强制性日期、关键事件或里程碑事件,项目支助者、项目顾客或其他项目相关人提出在某一特定日期前完成某些工作细目,一旦定下来,这些日期就很难被更改了)等。

进度估算和进度编制常常是结合在一起进行的,采用的方法也是一致的。一般说,项目进度编制的方法主要有:关键路径法(CPM)、时间压缩法、资源调整尝试法、关键链路法等。

#### 3.7.1 关键路径法

关键路径法 CPM(Critical Path Method)是杜邦公司开发的技术,它是根据指定的网络图逻辑关系进行的单一的历时估算,首先计算每一个活动的单一的、最早和最晚开始和完成日期,然后计算网络图中的最长路径,以便确定项目的完成时间估计。采用此方法可以配合进行进度的编制。借助网络图和各活动所需时间(估计值),计算每一活动的最早或最晚开始和结束时间。CPM 法的关键是计算总时差,这样可决定哪一活动有最短时间弹性,可以为更好的进行项目计划编制提供依据。CPM 算法也在其他类型的数学分析中得到应用。

一个项目往往是由若干个相对独立的任务链条组成的,各链条之间的协作配合就直接关系到整个项目的进度。

关键路径法属于一种数学分析方法,包括理论上计算所有活动各自的最早和最晚开始与结束日期。讲述关键路径进度编排方法前,先来了解一下有关进度编制的基本术语:

- 1) 最早开始时间(Early Start, ES):表示一项任务(活动)的最早可以开始执行的时间。
- 2) 最晚开始时间(Late Start, LS):表示一项任务(活动)的最晚可以开始执行的时间。
- 3) 最早完成时间(Early Finish, EF):表示一项任务(活动)的最早可以完成的时间。
- 4) 最晚完成时间(Late Finish, LF):表示一项任务(活动)的最晚可以完成的时间。
- 5) 超前(Lead):表示两个任务(活动)的逻辑关系所允许的提前提后置任务(活动)的时间,它是网络图中活动间的固定可提前时间。
- 6) 滞后(Lag):表示两个任务(活动)的逻辑关系所允许的推迟后置任务(活动)的时间,是网络图中活动间的固定等待时间。举一个简单的例子:装修房子的时候,需要刷房子,刷油漆的后续活动是刷涂料,他们之间需要至少一段时间(一般是一天)的等待时间,等油漆变干后,再刷涂料,这个等待时间就是滞后。
- 7) 浮动时间(Float):浮动时间是一个任务(活动)的机动性,它是一个活动在不影响项目完成的情况下可以延迟的时间量。其中:
  - 总浮动(Total Float):是在不影响项目最早完成时间本任务(活动)可以延迟的时间。 $Total\ Float = LS - ES$  或者  $Total\ Float = LF - EF$ 。
  - 自由浮动(Free Float):是在不影响后置任务最早开始时间本任务(活动)可以延迟的时间。 $Free\ Float = ES(\text{successor}) - EF(\text{predecessor}) - Lag$ ( $\text{successor}$  表示后置任务,  $\text{predecessor}$  表示前置任务,  $Lag$  是  $\text{successor}$  与  $\text{predecessor}$  之间的滞后时间)。

8) 关键路径：项目是由各个任务构成的，每个任务都有一个最早、最迟的开始时间和结束时间，如果一个任务的最早和最迟时间相同，则表示其为关键任务，一系列不同任务链条上的关键任务链接成为项目的关键路径，关键路径是整个项目的主要矛盾，是确保项目能否按时完成的关键。关键路径是网络图中浮动为0，而且是网络图中的最长路径。关键路径上的任何活动延迟，都会导致整个项目完成时间的延迟。它是完成项目的最短时间量。

下面以图3-16来进一步说明这些基本术语的含义（假设图中所有任务的历时以天为单位）。

如果A、B、C、E四个任务是并行的关系，则项目的完成时间是100。任务A的最早开始时间和最早结束时间都为0，最早结束时间和最晚结束时间都为100，而任务B的历时为10，所以可以有一定的浮动时间，只要在任务A完成之前任务B完成就可以了，所以，任务B的最早开始时间是0，最早结束时间是10；而它的最晚开始时间是90，最晚结束时间是100；所以它有90天的浮动时间，这个浮动是总浮动（Total Float）。总浮动是在不影响项目最早完成时间本活动可以延迟的时间。任务B的总浮动 $=90 - 0 = 100 - 10 = 90$ 。

任务C是任务D的前置任务，任务D是任务C的后置任务，它们之间的Lag=3表示任务C完成后的3天任务D开始执行。任务C的历时是10，任务D的历时是5，所以C任务和D任务的最早开始时间分别是0和13，最早结束时间分别是10和18，如果保证任务D的最早开始时间不受影响的话，任务C是不能自由浮动的，所以，任务C的自由浮动为0，即ES(D)-EF(C)-lag=0(ES(D))是任务D的最早开始时间，EF(C)是任务C的最早完成时间）。

任务F是任务E的后置任务，它们之间的Lead=3表示任务F在任务E结束之前的3天开始，也就是说任务F可以与任务E并行（搭接）工作一段时间。

从图3-16看，路径A是浮动为0而且是网络图中的最长路径，所以它是关键路径，是完成项目的最短时间。

再看图3-17。如何确定其中的关键路径（假设图中所有任务的历时以天为单位）。

1) 从网络图可以知道有两条路径：

$A \rightarrow B \rightarrow C \rightarrow E$  和  $A \rightarrow B \rightarrow D \rightarrow F$ 。

2)  $A \rightarrow B \rightarrow C \rightarrow E$  的长度是10(天)，有

浮动时间； $A \rightarrow B \rightarrow D \rightarrow F$ 的长度是16(天)，没有浮动时间。

3) 最长而且没有浮动的路径 $A \rightarrow B \rightarrow D \rightarrow F$ 便是关键路径。

4) 项目完成的最短时间是16天，即关键路径的长度是16天。

图3-18代表网络图中的一个任务（活动），其中，图中标识出任务的名称、任务的工期，同时也可标识出任务的最早开始时间ES、最早完成时间EF、最晚开始时间LS以及最晚完成时间LF。为了能够确定项目路径中各个任务的最早开始时间、最早完成时间、最晚开始时间、最晚完成时间，可以采用正推法和逆推法来确定。

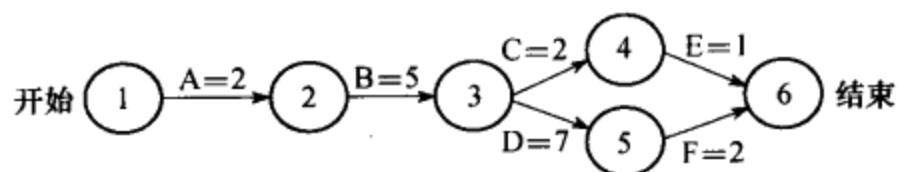


图3-16 项目网络图

图3-17 项目的网络图2

ES: 最早开始时间	任务名称:	EF: 最早完成时间
LS: 最晚开始时间	工期:	LF: 最晚完成时间

图3-18 任务图示

### 1. 正推法

在网络图中按照时间顺序计算各个任务（活动）的最早开始时间和最早完成时间的方法称为正推法。此方法的执行过程如下：

- 首先确定项目的开始时间。
- 项目的开始时间是网络图中第一个任务（活动）的最早开始时间。

3) 从左到右, 从上到下进行任务编排。

4) 当一个任务有多个前置任务时, 选择其中最大的最早完成日期 (加上 Lag 或者减 Lead) 作为其后置任务的最早开始日期。

5) 公式:

- $ES + Duration = EF$  ( $Duration$  是一个任务 (活动) 的历时时间)
- $EF + Lag = ES_s$  ( $ES_s$  是后置任务 (活动) 的最早开始时间)

项目的开始时间是网络图中第一个活动的最早开始时间, 图 3-19 这个网络图 (假设图中所有任务的历时以天为单位) 中项目的开始时间是 1, 比如任务 A; 它的最早开始时间  $ES(A) = 1$ , 任务历时  $Duration = 7$ , 则任务 A 的最早完成时间是  $EF(A) = 1 + 7 = 8$ ; 同理可以计算任务 B 的  $EF(B) = 1 + 3 = 4$ ; 任务 C 的最早开始时间  $ES(C) = EF(A) + 0 = 8$ , 最早完成时间是  $EF(C) = 8 + 6 = 14$ ; 任务 G 的最早开始时间和最早完成时间  $ES(G) = 14$ ,  $EF(G) = 14 + 3 = 17$ ; 同理  $ES(D) = 4$ ,  $EF(D) = 7$ ,  $ES(F) = 4$ ,  $EF(F) = 6$ 。由于任务 E 有两个前置任务, 选择其中最大的最早完成日期 (因为  $Lag = 0$ ,  $Lead = 0$ ) 作为其后置任务的最早开始日期, 所以  $ES(E) = 7$ ,  $EF(E) = 10$ ; 任务 H 也有两个前置: 任务 E 和任务 G, 选择其中最大的最早完成时间 17 (因为  $Lag = 0$ ,  $Lead = 0$ ), 作为任务 H 的最早开始时间  $EF(H) = 17$ ,  $EF(H) = 19$ 。

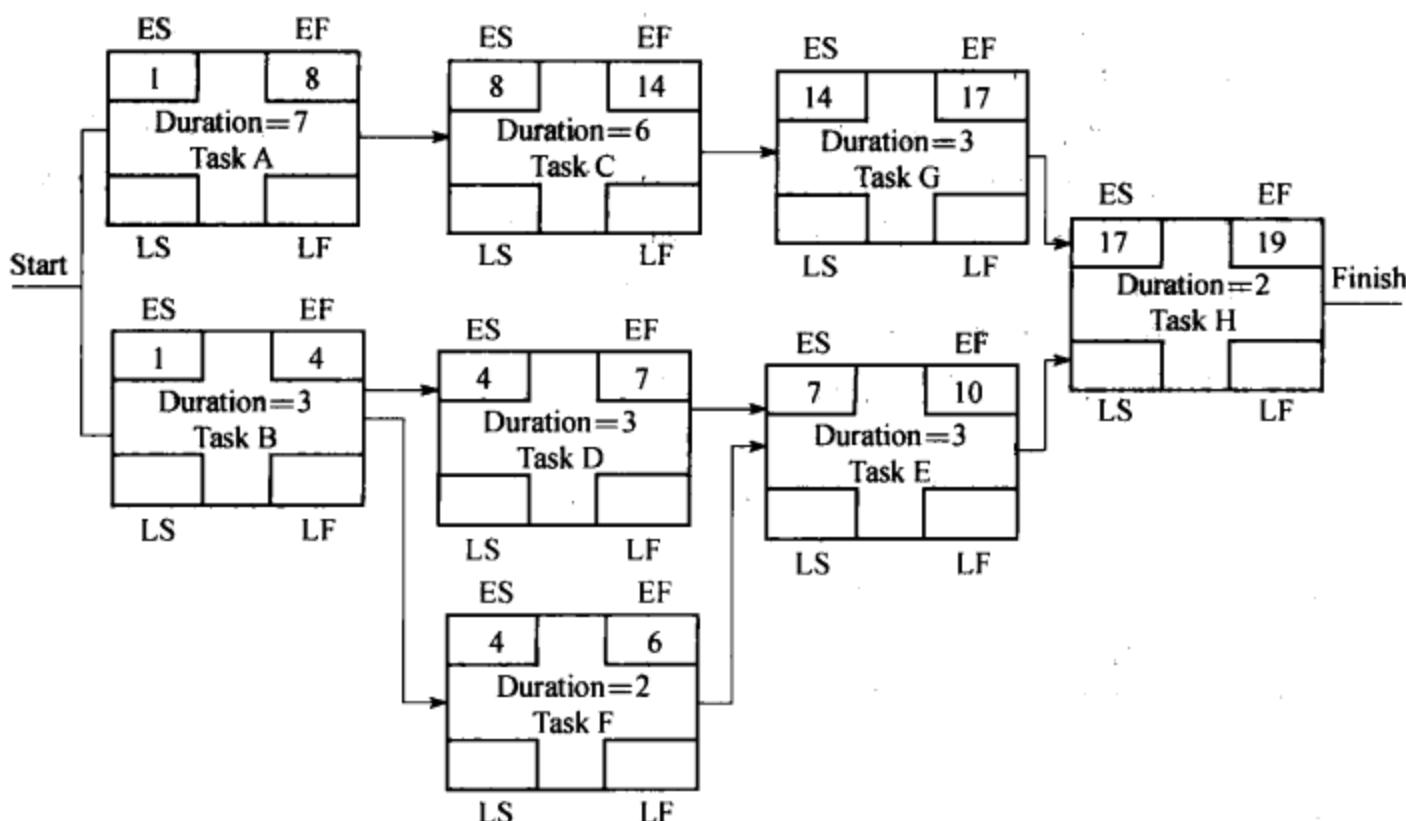


图 3-19 正推法

这样, 通过正推法确定了网络图中各个任务 (活动) 的最早开始时间和最早完成时间。

## 2. 逆推法

在网络图中按照逆时间顺序计算各个任务 (活动) 的最晚开始时间和最晚完成时间的方法, 称为逆推法。此方法的执行过程如下:

- 1) 首先确定项目的结束时间。
- 2) 项目的结束时间是网络图中最后一个任务 (活动) 的最晚结束时间。
- 3) 从右到左, 从上到下进行计算。
- 4) 当一个前置任务有多个后置任务时, 选择其中最小最晚开始日期 (减 Lag 或者加上 Lead) 作为其前置任务的最晚完成日期。

5) 公式:

- $LF - Duration = LS$
- $LS - Lag = LF_p$ :  $LF_p$  是前置任务的最晚完成时间

接下来确定图3-19中各个任务（活动）的最晚开始时间和最晚完成时间。由于项目的结束时间是网络图中最后一个任务（活动）的最晚开始时间，对图3-19网络图，这个项目的结束时间是19，即 $LF(H) = 19$ ，则 $LS(H) = 19 - 2 = 17$ ， $LF(E) = 17$ ， $LS(E) = 17 - 3 = 14$ ，同理，任务G、C、A、D、F最晚完成和最晚开始时间分别为： $LF(G) = 17$ ， $LS(G) = 17 - 3 = 14$ ； $LF(C) = 14$ ， $LS(C) = 14 - 6 = 8$ ； $LF(A) = 8$ ， $LS(A) = 8 - 7 = 1$ ； $LF(D) = 14$ ， $LS(D) = 14 - 3 = 11$ ； $LF(F) = 14$ ， $LS(F) = 14 - 2 = 12$ ；任务B有两个后置任务，选择其中最小最晚开始日期（因为 $Lag = 0$ ,  $Lead = 0$ ）作为其前置任务的最晚完成日期，所以将11作为任务B的最晚完成日期即 $LF(B) = 11$ ， $LS(B) = 11 - 3 = 8$ 。另外，对于任务F，它的自由浮动时间是1（即7-6）；而它的总浮动时间是8（即 $12 - 4 = 8$ ，或者 $14 - 6 = 8$ ）。结果见图3-20，图中的 $A \rightarrow C \rightarrow G \rightarrow H$ 是浮动为0而且是最长的路径，所以它是关键路径，关键路径长度是19，所以项目的完成时间是19（天）而A、C、G、H都是关键任务。

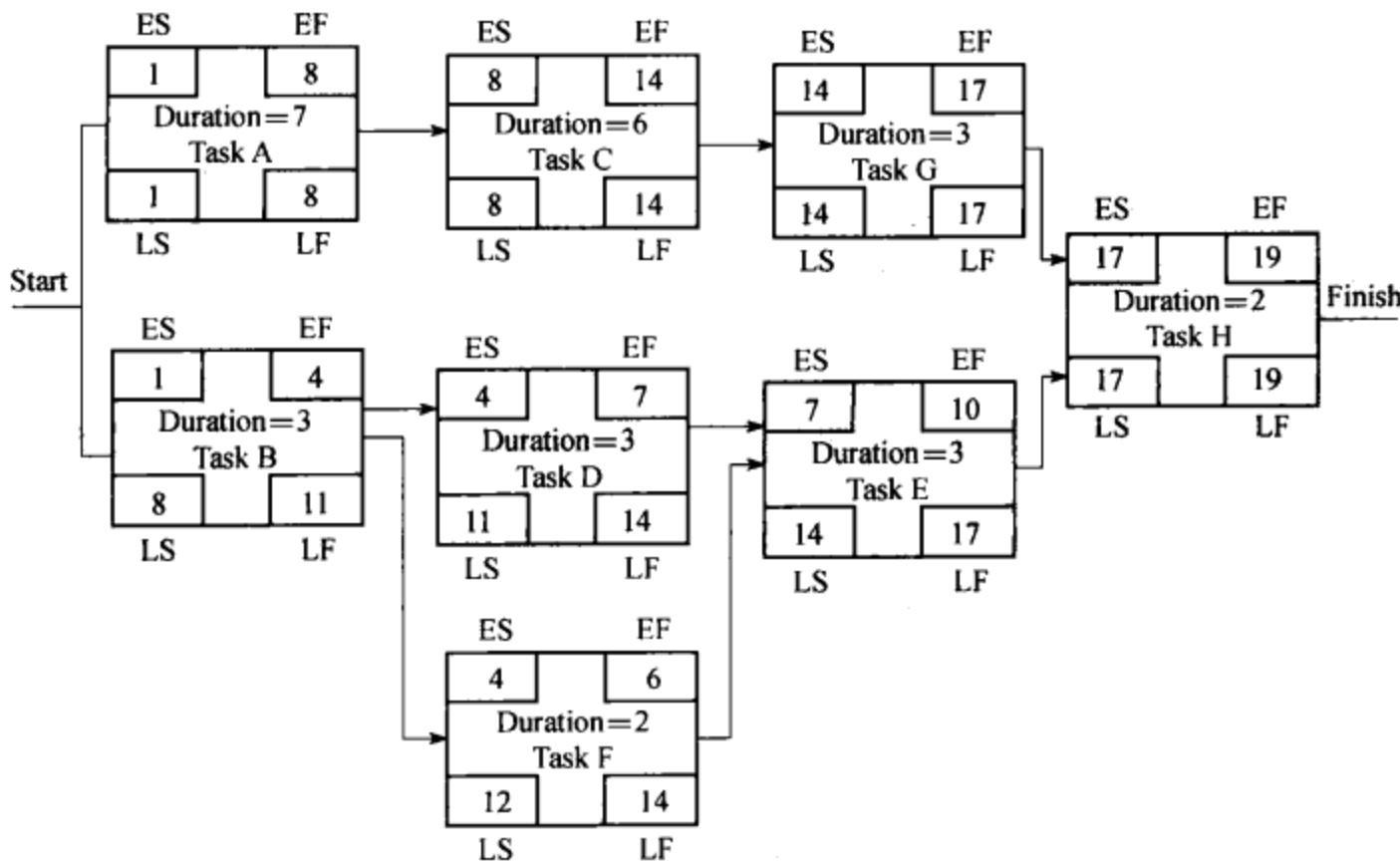


图3-20 逆推法

图3-20网络图可以称为CPM网络图，如果采用PERT进行历时估计，则可以称为PERT网络图。PERT网络与CPM网络是五十年代末发展起来的两项重要的技术，他们的主要区别是PERT计算历时时存在一定的不确定性，采用的算法是加权平均 $(O + 4M + P)/6$ ，CPM计算历时时意见比较统一，采用的算法是最大可能值M。1956年美国杜邦公司首先在化学工业上使用了CPM（关键路径法）进行计划编排；美国海军在建立北极星导弹时，采用了BuzzAllen提出的PERT（计划评审法）技术。这两种方法才逐渐渗透到许多领域，为越来越多的人所采用，成为网络计划技术的主流。网络计划技术作为现代管理的方法与传统的计划管理方法相比较，具有明显优点，主要表现为：

- 1) 利用网络图模型，明确表达各项工作的逻辑关系。按照网络计划方法，在制订工程计划时，首先必须理清楚该项目内的全部工作和它们之间的相互关系，然后才能绘制网络图模型。
- 2) 通过网络图时间参数计算，确定关键工作和关键线路。
- 3) 掌握机动时间，进行资源合理分配。
- 4) 运用计算机辅助手段，方便网络计划的调整与控制。

我国从20世纪60年代中期开始，在著名数学家华罗庚教授的倡导和亲自指导下，开始试点应用网络计划，并根据“统筹兼顾，全面安排”的指导思想，将这种方法命名为“统筹方法”。网络计划技术从此在国内生产建设中卓有成效地推广开来。

为确保网络图的完整和安排的合理，可以进行如下的检查：

- 1) 是否正确标识了关键路径?
- 2) 是否有哪个任务存在很大的浮动?如果有,则需要重新规划。
- 3) 是否有不合理的空闲时间?
- 4) 关键路径上有什么风险?
- 5) 浮动有多大?
- 6) 哪些任务有哪种类型的浮动?
- 7) 工作可以在期望的时间内完成吗?
- 8) 提交物可以在规定的时间内完成吗?

关键路径法是理论上计算所有活动各自的最早和最晚开始与结束日期,但计算时并没有考虑资源限制。这样算出的日期并不是实际进度,而是表示所需的时间长短,在编排实际的进度时,应该考虑资源限制和其他约束条件,把活动安排在上述时间区间内。所以还需要诸如时间压缩、资源调整等方法。

### 3.7.2 时间压缩法

时间压缩法是一种数学分析的方法,是在不改变项目范围前提下(例如,满足规定的日期或满足其他计划目标),寻找缩短项目时间途径的方法。应急法和平行作业法都是时间压缩法。

#### 1. 应急法

应急法也称赶工(crash),赶工也称为时间-成本平衡方法,是权衡成本和进度间的得失关系,以决定如何用最小增量成本以达到最大量的时间压缩。应急法并不总是产生一个可行的方案且常常导致成本的增加。

一旦项目的工作方法和工具得当,可以简单的通过增加人员和加班时间来缩短进度,进行进度压缩。在进行进度压缩时存在一定的进度压缩和费用增长的关系,很多人提出不同的方法来估算进度压缩与费用增长的关系,这里介绍其中两种方法。

##### (1) 时间成本平衡(Time-Cost Trade-Off)方法

时间成本平衡方法是基于下面的假设:

- 1) 每个任务存在一个“正常”进度(normal time)和“可压缩”进度(crash time),一个“正常”成本(normal cost)和“可压缩”成本(crash cost)。
- 2) 通过增加资源,每个任务的历时可以从“正常”的进度压缩到“可压缩”进度。
- 3) 每个任务无法在低于“可压缩”进度内完成。
- 4) 有足够的资源可以利用。
- 5) 在“正常”与“可压缩”之间,进度压缩与成本的增长成正比的,单位进度压缩的成本(cost per time period)=(可压缩成本-正常成本)/(正常进度-可压缩进度)。

上述的线性关系方法是假设如果任务在可压缩进度内,进度压缩与成本的增长成正比的。所以可以通过计算任务的单位进度压缩的成本,来计算在压缩范围之内的进度压缩产生的压缩费用。

**【例2】**图3-21是一个项目的PDM网络图,如果A、B、C、D任务在可压缩的范围内,进度压缩与成本增长成线性正比关系,表3-5分别给出了A、B、C、D任务的正常进度和可压缩进度,正常成本和可压缩成本。从PDM网络图可知:目前项目的总工期为18周,如果将工期分别压缩到17周、16周、15周并且保证每个任务在可压缩的范围内,应该压缩哪些任务,并计算压缩之后的总成本?

表3-5 正常进度、可压缩进度,正常成本、可压缩成本

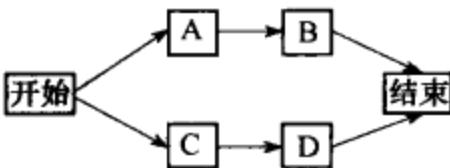


图3-21 项目PDM网络图

	A	B	C	D
正常进度	7周	9周	10周	8周
正常成本	5万	8万	4万	3万
可压缩进度	5周	6周	9周	6周
可压缩成本	6.2万	11万	4.5万	4.2万

1) 从 PDM 网络图可以看到, 有“开始→A→B→结束”和“开始→C→D→结束”两个路径, 前者的长度是 16 周, 后者的长度是 18 周。所以“开始→C→D→结束”是关键路径。即项目完成的最短时间是 18 周。

2) 如果将工期分别压缩到 17 周、16 周、15 周并且保证每个任务在可压缩的范围内, 必须满足两个前提:

- A、B、C、D 任务必须在可压缩的范围内。
- 保证压缩之后的成本最小。

根据表 3-5 计算 A、B、C、D 任务单位进度压缩的成本, 如表 3-6 所示。

表 3-6 每个任务的单位进度压缩成本

单位压缩成本 任务	A	B	C	D
压缩成本 (万/周)	0.6	1	0.5	0.6

根据上述两个条件, 首先看可以压缩的任务, 然后根据压缩后的情况, 计算总成本最小的情况, 此情况为我们选择的压缩结果, 如表 3-7 所示。

表 3-7 压缩后的项目成本

压缩任务及成本 完成周期 (单位: 周)	可以压缩的任务	压缩的任务	成本计算 (单位: 万)	项目成本 (单位: 万)
18			$5 + 8 + 4 + 3$	20
17	C, D	C	$20 + 0.5$	20.5
16	C, D	D	$20.5 + 0.6$	21.1
15	A, B, C, D	A, D	$21.1 + 0.6 + 0.6$	22.3

3) 如果希望总工期压缩到 17 周, 需要压缩关键路径“开始→C→D→结束”, 可以压缩的任务有 C 或者 D, 但是根据表 3-6 知道压缩任务 C 的成本最小 (压缩任务 C 增加 0.5 万, 压缩任务 D 增加 0.6 万), 故选择压缩任务 C 一周。所以, 压缩到 17 周后的总成本是 20.5 万。

4) 如果希望总工期压缩到 16 周, 需要压缩关键路径“开始→C→D→结束”, 可以压缩的任务还是 C 或者 D, 但是这时任务 C 在可压缩范围内是不能再压缩的, 否则压缩成本会非常高, 应该选择压缩任务 D 一周, 所以, 项目压缩到 16 周后的总成本是 21.1 万。这时, 项目网络图的两条路径的长度都是 16 周, 即有两条关键路径。

5) 如果希望总工期压缩到 15 周, 应该压缩两条关键路径, 即“开始→A→B→结束”和“开始→C→D→结束”两条路径都需要压缩, 在 A、B 任务中应该选择压缩任务 A 一周 (压缩任务 A 一周增加 0.6 万成本, 压缩任务 B 一周增加 1 万成本), 在 C、D 中选择压缩 D 一周 (这样的压缩成本是最低的), 所以, 项目压缩到 15 周后的总成本是 22.3 万。

## (2) 进度压缩因子方法

进度压缩与费用的上涨不是总能呈现正比的关系, 当进度被压缩到“正常”范围之外, 工作量就会急剧增加, 费用会迅速上涨。而且, 软件项目存在一个可能的最短进度, 这个最短进度是不能突破的, 见图 3-22。在某些时候, 增加更多的软件开发人员会减慢开发速度而不是加快速度。例如, 一个人 5 天写 1000 行程序, 5 个人 1 天不一定写 1000 行程序, 40 个人 1 个小时不一定写 1000 行程序。增加人员会存在更多的交流和管理的时间。软件项目中存在的这个最短的进度点, 不管怎样努力工作, 无论怎样聪明工作, 无论怎么寻求创造性的解决办法, 也无论你组织多大的团队, 都不能突破这个最短的进度点。

进度压缩因子方法是由著名的 Charles Symons 提出来的, 而且被认为是精确度比较高的一种方法。它的公式为:

- 进度压缩因子 = 期望进度 / 估算进度

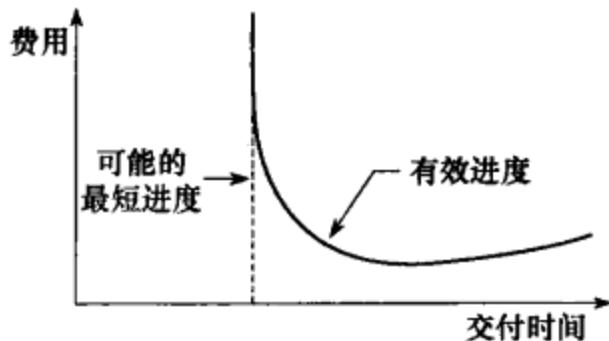


图 3-22 进度与费用的关系图

- 压缩进度的工作量 = 估算工作量 / 进度压缩因子

这个方法是首先估算初始的工作量和初始的进度，然后将估算与期望的进度相结合，计算进度压缩因子，以及压缩进度的工作量。例如：项目的初始估算进度是 12 月，初始估算工作量 78 人月。如果期望压缩到 10 月，则进度压缩因子 =  $10/12 = 0.83$ ，压缩进度后的工作量 =  $78/0.83 = 94$ （人月），即压缩进度增加的工作量是 16 人月。也就是说进度缩短 17%，增加 21% 的工作量。

很多的研究表明：进度压缩因子不应该小于 0.75。

## 2. 平行作业法

平行作业法也称为快速跟进（fast tracking），是平行地做活动，这些活动通常要按前后顺序进行（例如，在设计完成前，就开始软件程序的编写）。例如图 3-23，正常情况下，15 天内需求、设计完成。但是，如果需要设计在第 12 天内完成的话，需要对项目的历时进行压缩。有两种压缩方法，一个 是应急法，不改变任务之间的逻辑关系，将需求压缩到 8 天，设计压缩到 4 天，这样需求、设计可以在 12 天内完成。也可以采用另外的方法，即可以调整任务需求和设计之间的逻辑关系，即在需求还没有完成之前 3 天就开始设计，相当于需求任务与设计任务并行工作一段时间，或者说需求与设计任务之间的 lead = 3，它解决任务的搭接。这样就压缩了项目的时间。但是，平行作业常导致返工和增加风险。

### 3.7.3 资源平衡

为了成功地编制了一个项目进度计划，项目中的任务必须分配资源，项目中的任务都必须在一定的条件下人为去操纵完成。要使用资源来完成项目中的任务，就必须将资源与任务联系起来。每项任务需要的资源包括人力资源，设备资源等。

资源平衡（resource leveling）方法是通过调整任务的时间来协调资源的冲突，这个方法的主要目的是形成平稳连续的资源需求，最有效的利用资源，使资源闲置的时间最小化，同时，尽量避免超出资源能力。

关键路径法通常可以产生一个初始进度计划，而实施这个计划需要的资源可能比实际拥有的多。资源平衡法可在资源有约束条件下制定一个进度计划。例如网络图 3-24 中的 A、B、C 三个任务，A 需要 2 天 2 个开发人员完成，B 需要 5 天 4 个开发人员完成，C 需要 3 天 2 个开发人员完成。如果三个活动同时开始执行，则如图 3-25，一共需要 8 个开发人员，而资源高峰在项目的开始前 2 天，之后，就会陆续有人出现空闲状态，资源利用率不合理。如果 A 任务利用浮动，使用它的最晚开始时间，即 A 任务完成之后 C 任务再开始，如图 3-26 所示，从项目开始到结束，一共需要 6 个开发人员，而且项目同样是 5 天内全部完成，但是资源利用率提高了，这就是一个资源平衡的例子。

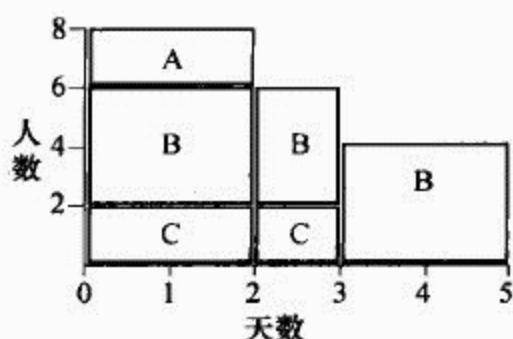


图 3-25 三个任务同时开始的人员情况

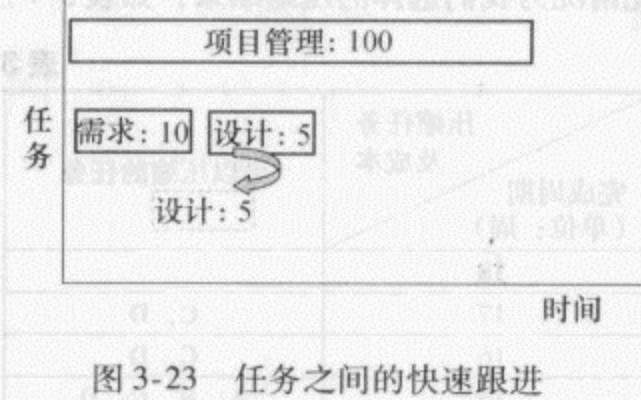


图 3-23 任务之间的快速跟进

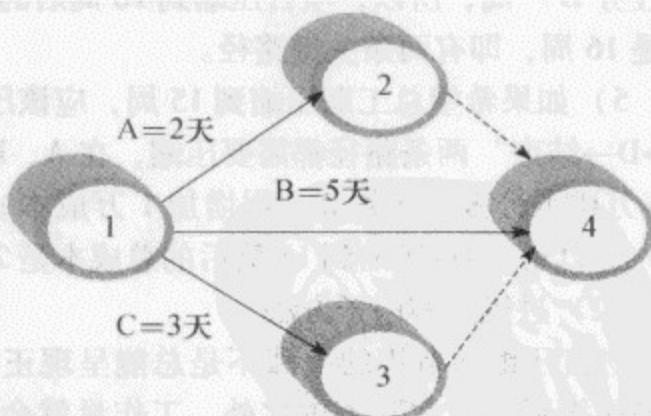


图 3-24 项目网络图

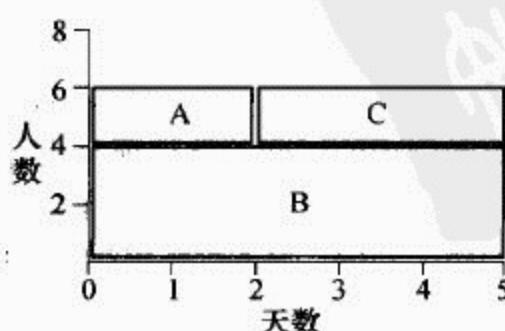


图 3-26 三个任务不同时开始的人员情况

### 3.7.4 关键链法

关键链项目管理（Critical Chain Project Management, CCPM）自提出以来，就引起了广泛的反响，被认为是项目管理领域自发明关键路径法（CPM）和工程评估评审技术（PERT）以来最重要的进展之一，已经成为近年来项目管理领域理论研究的一个热点。关键链方法在实际应用中已取得了巨大成功，许多国际知名公司纷纷采用关键链方法提高项目管理绩效。例如，美国通信设备制造商 Harris 公司通过应用关键链方法成功将建设半导体工厂的时间从行业平均水平 46 个月缩短到了 14 个月；以色列航空公司则将飞机的平均维护时间从 3 个月降到了 2 周；朗讯公司则将 R&D 项目的平均工期缩短了 25%。

关键链法的思路是怎样把人们的工作习惯考虑到管理工作中去，在项目估算和项目管理中因地制宜地来提高项目的绩效。关键链法和关键路径法的区别是：关键路径法是工作安排尽早开始，尽可能提前。而关键链法是尽可能推迟。关键链法的提出主要基于两个方面的考虑：

1) 如果一项工作尽早开始往往存在着一定的松弛量、浮动时间和安全富裕量，那么这个工作往往推遲到它最后所允许的那一天为止。这一期间整个工作就没有充分发挥它的效率，造成了人力、物力的浪费。如果按最迟的时间开始做安排，没有浮动和安全富裕量，无形当中对从事这个项目的人员施加了压力，他没有任何选择余地，只有尽可能努力地按时完成既定任务。这是关键链法所采用的一种思路。

2) 在进行项目估算时，需要设法把个人估算中的一些隐藏的富裕量剔除。经验表明，人们在进行估算时，往往是按照能够 100% 所需要的时间来进行时间估算。在这种情况下，如果按照 50% 的可能性，只有一半的可能性能够完成任务，有 50% 的可能性又要延期，这样就大大缩短原来对工作的时间估算。

按照平均规律，把项目中所有的任务都按照 50% 的规律进行项目的时间估算，结果使项目整个估算时间总体压缩了 50%，如果把它富余的时间压缩出来，作为一个统一的安全备用，作为项目管理的一个公共资源统一调度、统一使用，使备用的资源有效运用到真正需要它的地方，这样就可以大大缩短原来的项目工期。

关键链项目管理方法是约束理论在项目管理中的应用。

#### 1. 约束理论

约束理论（Theory Of Constraints, TOC）是由 Goldratt 博士于 20 世纪 80 年代中期在最优化生产技术（OPT）基础上发展起来的。约束理论的核心思想可以归纳为两点：(1) 所有现实系统都存在约束。如果一个系统不存在约束，就可以无限提高产出或降低成本，而这显然是不实际的。因此，任何妨碍系统进一步提升生产率的因素，就构成了一个约束。约束理论将一个企业看作一个系统，在企业内部的所有流程中，必然存在阻碍企业进一步降低成本和提高利润的因素，这些因素也就是企业的约束。(2) 约束的存在表明系统存在改进的机会。虽然约束妨碍了系统的效率，但约束也恰恰指出了系统最需要改进的地方。一个形象的类比就是“木桶效应”，一只木桶的容量取决于最短的那块木板，而不是最长的木板。因此，对约束因素的投资，才是最有效率的改进系统效率的方法。

与其他管理理念不同，约束理论将制约企业发展的约束看作企业突破现状不断取得改进的关键，因此约束具有正面的意义。在这样的理念基础上，为了有效提升系统的效率，约束理论提出了五大关键步骤：(1) 找出系统中的约束因素；(2) 决定如何挖掘约束因素的潜力；(3) 使系统中所有其他工作服从于第二步的决策；(4) 提升约束因素的能力；(5) 若该约束已经转化为非约束性因素，则回到第一步，否则回到第二步，要注意不要让思维惯性成为新的主要约束因素。这五个步骤构成一个不间断的循环，帮助系统实现持续改进。

约束理论可以直接应用于企业实践，并已取得良好效果。例如，通过对 77 家应用约束理论的企业统计发现，有 14 家企业将生产周期缩短了 66%，有 28 家企业将库存压缩了 50%，有 32 家企业的投产时间缩短了 69%。

约束理论在项目管理，尤其是项目进度管理上的应用，导致了关键链项目管理方法的产生。

## 2. 关键链法

以项目管理为中心是企业提升核心竞争力的有效手段，但目前我国项目管理水平与发达国家存在较大差距，项目失败的比例很高。鉴于关键链方法在国外企业实践中取得的巨大成功，可预期该方法在我国的推广应用将有助于提高我国的项目管理水平、提升企业竞争力。

如果将一个项目看作一个系统，那么应用约束理论的第一步，就是要确定项目的约束。从 CPM 和 PERT 开始，项目中的关键路径就被看作项目管理的基础。但是，关键路径法主要是分析前置后置关系，并不考虑项目实际能调动的资源是有限的，因此关键路径法被广泛批评的一点就是其进度通常不具有可行性，而是需要进行后续调整。与关键路径不同，关键链不仅考虑项目中各任务的前置后置关系，也充分考虑项目中现实存在的资源约束。如图 3-27 所示，在关键路径方法中，任务 A、D、E、F 组成了项目的关键路径，但如果考虑资源限制，假设任务 C 和任务 E 需要同一种资源，例如需要同一台机器进行开发，而该机器一次只能执行一项作业，那么事实上任务 C 和任务 E 是不能同时进行的。因此，在考虑资源约束的情况下，项目的关键任务为 A、D、C、F，这五个任务就构成了项目的关键链。可见，是关键链而不是关键路径，决定了项目在给定的紧前关系和资源条件下完成项目所需的最短时间。

如果将关键链看作项目的“约束”因素，那么应用约束理论的第二步就是要考虑如何来挖掘该约束因素的潜力，即如何缩短关键链所需的时间，因为关键链所需的时间正是完成项目所需的时间。在关键路径法中，为了保证任务能够有较高的概率在计划时间内完成，同时也由于项目组成员普遍存在的风险规避心理，一般的计划时间都大于完成任务所需的平均时间，也可以看作是在任务所需的平均时间上增加了一块“安全时间”(Safety Time, ST)。这样的处理方式具有两方面的效果，正面效果是提高了管理不确定因素的能力，负面效果则是延长了完成项目所需的时间。关键链方法采取了另一种方式，用任务所需的平均时间作为最终的计划时间，但考虑到任务内在的不确定性，在关键链的末端附加整块的安全时间，也就是项目的缓冲时间(Project Buffer, PB)(即管理预留)。可以认为关键链方法是重新配置了关键路径法中分散存在的安全时间，但这样的重新配置能够缩短项目所需的时间，因为根据概率理论，在整合安全时间后，在相同概率下，只需要较少的时间就可以完成所有任务。图 3-28 直观地说明了关键链方法在这方面的优越性。

在完成关键链的进度安排后，还需要保证所有关键链上的任务(关键任务)不受其他非关键任务的影响，以保证项目能够按计划及时完成。在现实的项目实践中，虽然采用了增加安全时间的方法，但仍然有大量的项目未能按期完成。造成项目延期的原因很多。第一，前置任务的延迟会导致后续任务的延迟。第二，当存在多个前置任务时，延迟最久的任务起了决定性作用，导致项目的延迟，而提前完成的任务并不能使后续任务提前开始。第三，由于任务时间包含了安全时间，导致项目组成员在心理上觉得还有充裕的时间，结果使得任务启动过晚，同时调查还发现，项目成员总是倾向于完全消耗掉任务分配到的时间。其中第三个原因也是关键链方法采用平均时间的理由，以期推动项目组成员能够全力以赴地开展工作。而前两个原因则使关键链方法引入了非关键链缓冲时间(Feeding Buffer, FB)这一概念。

如图 3-29 所示，任务 C、D、E 组成了项目的关键链，而任务 A、B 为非关键任务。由于任务 B 是任务 E 的前置任务，为了防止任务 A 和 B 可能发生的延迟导致任务 E 不能按时开始，因此需要在任务 B 之后安排一定的缓冲时间，或者说让任务 A 和 B 有一定的提前量。这样，就可以有效地防止非关键任务对关键链产生负面影响。

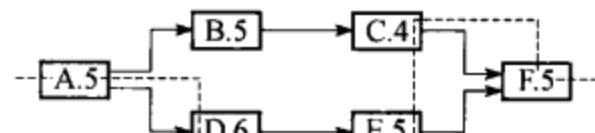


图 3-27 关键链



图 3-28 关键路径方法和关键链方法

在风险管理上的差异

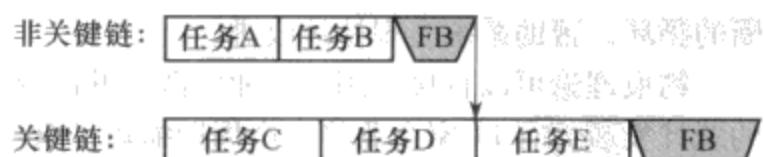


图 3-29 非关键链的缓冲时间

与项目的缓冲时间类似，非关键链缓冲时间整合与压缩了所有非关键链任务的安全时间。

非关键链缓冲时间能够保护项目按计划进行，不受任务的不确定因素影响，同时还可以作为一种预警机制。如果前置任务因为不确定因素而没有在计划时间内完成，那么其后置任务就无法按计划时间启动，其结果就是缓冲时间被占用。缓冲时间被占用得越多，就说明越有可能延误后续的关键任务。因此，当占用比例到达一定程度，比如三分之一或三分之二，就需要发出一个警告信号来提醒项目经理对延迟的任务加以关注，并考虑是否采取措施防止任务进一步延期。

非关键链缓冲时间可以保护关键链任务不受非关键链任务的影响，但还需要考虑资源约束对关键链的影响，尤其是同一资源在不同任务间切换常常需要一定的准备时间。因此，关键链方法引入了资源缓冲（Resource Buffer, RB）的概念，以防止关键链任务因资源没有及时到位而发生延误。与缓冲时间不同，资源缓冲本质上是一种警示信号，用来提醒项目经理或者部门经理保证资源及时到位。关键链方法要求在关键任务所需的资源被前置的非关键任务占用时，应当提前一定时间在项目进度计划上标识资源缓冲，以便及时提醒项目经理协调资源，防止因资源不能及时到位而延误关键任务。

关键链项目管理方法提供了一种有效管理项目进度的新思路。考虑到现实的资源约束和项目内在的不确定性，关键链方法通过合理设置缓冲时间，从而有效保证任务的不确定性不会影响项目的完工时间。因此，关键链方法尤其适用于不确定性较大的项目，例如新产品开发等项目。

关键链方法较好地解决了缩短项目工期和降低项目成本的问题。关键链方法以最小化项目工期为首要目标，要求对项目进度实现动态管理，对于关键链上的任务，要求一个任务完成后能立刻启动下一个任务，尤其是在一个任务提前完成的情况下，要能够通过项目组的协调和沟通提前启动下一个任务。因此，关键链方法对项目组的沟通管理提出了较高的要求。因为后置任务一般也需要一定的准备时间，这就要求前置任务在可能提前完成时，能够及时和负责后续任务的项目组成员进行沟通。对于非关键链上的任务，则推荐以最晚时间启动任务，这样可以减少同步进行的工作数量，在发现前置任务出现的问题时还可以降低返工次数。

目前的关键链方法主要是针对单一资源下的项目管理，重点强调人力资源在项目进度中的安排。因此，关键链方法提出要避免在同一时间段内给某一个人（或某一种资源）安排多个任务，认为正是由于项目中存在着这样不合理的安排，从而降低了资源的工作效率，延长了完成项目所需要的时间。但在实际大型项目中，往往存在多种资源，尤其某些重要资源，例如柔性制造系统等，能够同时处理多个任务。因此，对于复杂资源约束下的关键链项目管理，还需要进一步研究。

尽管关键链方法在理论与实践上都取得了巨大成功，但其理论仍然不够成熟。对关键链方法的批评主要集中在其过于简单化的时间估算方法上。首先，Goldratt 主张以项目成员所提供的估算时间的一半作为任务的平均时间，该做法过于偏激和简化。在多数工程项目中，这样“切一半”的方法是不可行的，很可能直接导致项目质量出现问题。其次，Goldratt 建议以关键链或非关键链时间长度的一半作为其缓冲时间的长度，无疑这样的估算方法也过于简化，在很多实际项目中会导致缓冲时间过长，从而产生不必要的资源浪费或丧失商机。因此，一些研究者提出了其他方法来改进缓冲时间的估算。例如，朗讯公司假设任务时间的最乐观估计和最悲观估计之差为任务所需时间的变动量，然后采用所有关键链上任务的变动量的平方和的平方根作为所需的缓冲时间。

关键链方法另一有待改进之处在于对缓冲时间的管理。虽然缓冲时间有助于控制项目的不确定性，但也增加了项目进度管理的复杂性。首先，非关键链的总时差可能不足以容纳所需的缓冲时间，这时问题就出现了。如果保持非关键链上的任务进度不变，那么后续的关键任务很可能因为缓冲时间被完全占用而不得不延期，这样就违背了设置缓冲时间的意义。或者考虑提前开始非关键链任务，但这样又非常可能和其他任务发生资源冲突，甚至可能会发现原先的关键链已经不是项目中最长的任务路线了。

Goldratt 并未详细讨论如何用关键链方法对多个项目进行管理。其他研究者则考虑了关键链方法或者约束理论在多项目管理中的应用。例如，Leach 认为应该找出多个项目共享的关键资源作为约束因素，从而对不同优先权的项目进行不同的管理。杨雪松等也采用类似思路进行了研究。

与传统项目管理方法相比，关键链方法较好地解决了资源约束和不确定性的问题，为项目管理提

提供了一种新思路，在国外众多企业的成功应用也证实了该方法的实际价值。当然，关键链方法还存在明显的不足。首先，该方法主要局限于单一资源下的单项目管理，对于涉及多种资源的项目，如何确定关键链尚缺乏研究。同时，目前的关键链方法在时间估计上过于简化，对缓冲时间的管理不够成熟。因此，关键链方法还需要通过进一步的深入研究来扩展应用领域和完善相关方法。

总结一下关键链法的优点。

#### (1) 可以提高项目的绩效

为了保证项目能够完成，我们还需要在工作当中安排一个裕量，也就是说在估算当中挖掘出潜力。如每一项工作都缩减 50%，把富余出来的时间，按照项目工期的 50% 来安排工作裕量，仍然按照项目的最晚开始，根据项目完工所需要的时间，首先安排项目的最后一个工作，然后再确定其次工作、长期工作，最后安排项目的起始工作，整个工作安排采用逆推法，由项目的结束向前进行安排。把安全裕量安排在项目工期的最后阶段。

#### (2) 在项目管理中便于抓住重点

第二个好处是在项目的管理中抓住重点，对重点工作进行管理，而不是像传统项目管理那样需要面面俱到，要对每一项工作定期地记录工作进展、分析偏差和跟踪项目的进展情况。用关键链法只需要关注那些已经延期的项目、工作，如果工作是在正常范围内进行，就可以在管理工作中摆在稍微次要的位置上。同时对项目的备用管理，对项目的总体进展情况做一个总体的管理和控制。这是关键链法的一个基本原理。

#### (3) 提前完成项目

关键链法管理的一个好处是可以提前完成项目。它通常比关键路径法至少提前 1/3 的时间。

关键链法主要的思路就是怎样把人们的工作习惯考虑到管理工作当中，在项目的估算和项目管理当中因地制宜地采用一种新的方法来提高项目绩效。

### 3.7.5 项目进度计划的优化

编制一个好的项目计划需要不断完善的过程，需要不断的优化、评审、修改、再评审、细化等，最后才可以确定成为基准的项目计划。刚刚编制出来的计划进度，如果与要求有差距，就要进行项目计划优化、调整资源、解决资源冲突，例如有的资源在两个任务中同时被使用，肯定是不合理的；同时，调整优化项目的进度，缩短工期；最后调整项目成本预算，以便减少项目费用。

对于进度的安排，应该有适度的压力，让开发人员有适度的紧迫感，同时，进度不可以太过分强调，以至于让大家焦点总集中在进度上。一方面，有了适度的紧迫感，可以让开发人员集中精力在最重要的事情上，如果没有紧迫感，开发人员就不会全力冲刺，紧张、有压力的脑力风暴可以激发好的点子。另一方面，不能过分强调进度，这对项目人员的士气伤害太大。如果过分强调进度，项目人员就随时有一种威胁的信息——我们快落后了，焦点总是集中在进度，而不是更重要的软件质量。过分强调进度，以至于做多少事情，都没有成就感，这样，开发人员总是被落后的威胁淹没，再怎么努力，也看不到工作的进展。所以，应该倡导一种是比较人性化的进度控制方式，即适度压力同时又不过分依赖进度。

项目具有渐近明了的特性，因此项目计划的制定需要在一定条件的限制和假设之下采用渐近明细的方式进行不断完善。一个好的项目计划的开发应该是渐近式的。例如可以采用局部细化的方法，对于较为大型的软件开发项目的任务（工作）分解结构 WBS 可采用多次 WBS 方法。其中部分的原因是需求的颗粒度在一开始往往是比较粗的，因此对于整体项目规模的估计误差范围也是比较大的。

一般来说，项目进度计划可以采用工具来协助编制，项目管理软件被广泛地使用以帮助项目进度的编制。这些软件可自动进行数学计算和资源调整，可迅速地对许多方案加以考虑和选择。用这些软件，还可打印显示出计划编制的结果。例如可以采用 Microsoft Project 编制计划等。

### 3.8 校务通系统案例分析

校务通系统的进度计划采用渐进完善的方式进行的。

## 1. 初期项目进度计划

由于项目初期信息不足，所以初期项目计划只是一个计划表格，相当于一个大计划，简单说明计划的执行步骤，如表 3-8。

## 2. 项目计划的细化

表 3-8 的大计划不能作为指导详细工作的计划，是比较粗的，还需要进一步细化。随着对项目了解的深入，项目计划也会逐步完善和细化的。

首先，根据 WBS 的分解情况，继续分解相应的活动（任务），使用 MS project 工具，将分解后的所有活动（任务）和 WBS 的代码录入项目计划文件中，然后确定各个活动之间的关系，由于采用增量式的生存期模型，在需求设计之后，系统的功能采用增量方式实现，实施阶段分 6 个增量，对各个任务（活动）分配相应的资源，然后经过不断的资源调整优化以及工期、活动关系的调整等，再经过多次的评审讨论，最后计划通过评审，将此计划存为基准计划。项目的基准计划如下。

### (1) 进度计划

表 3-9 的进度计划是细化后的项目进度计划。

表 3-8 项目初期计划

任 务	完成时间	负责人	资 源	备 注
需求讨论	2003-4-9	韩万江	2 开发人员参与	
项目规划	2003-4-12	韩万江	全体人员参与	
需求确定	2003-4-16	韩万江	全体人员参与	
设计	2003-4-20	姜岳尊	杨炎泰参与	
项目实施	2003-5-30	姜岳尊	杨炎泰参与	有待细化
测试	2003-6-4	郭天奇	开发人员参与	
提交	2003-6-5	韩万江		

表 3-9 项目计划

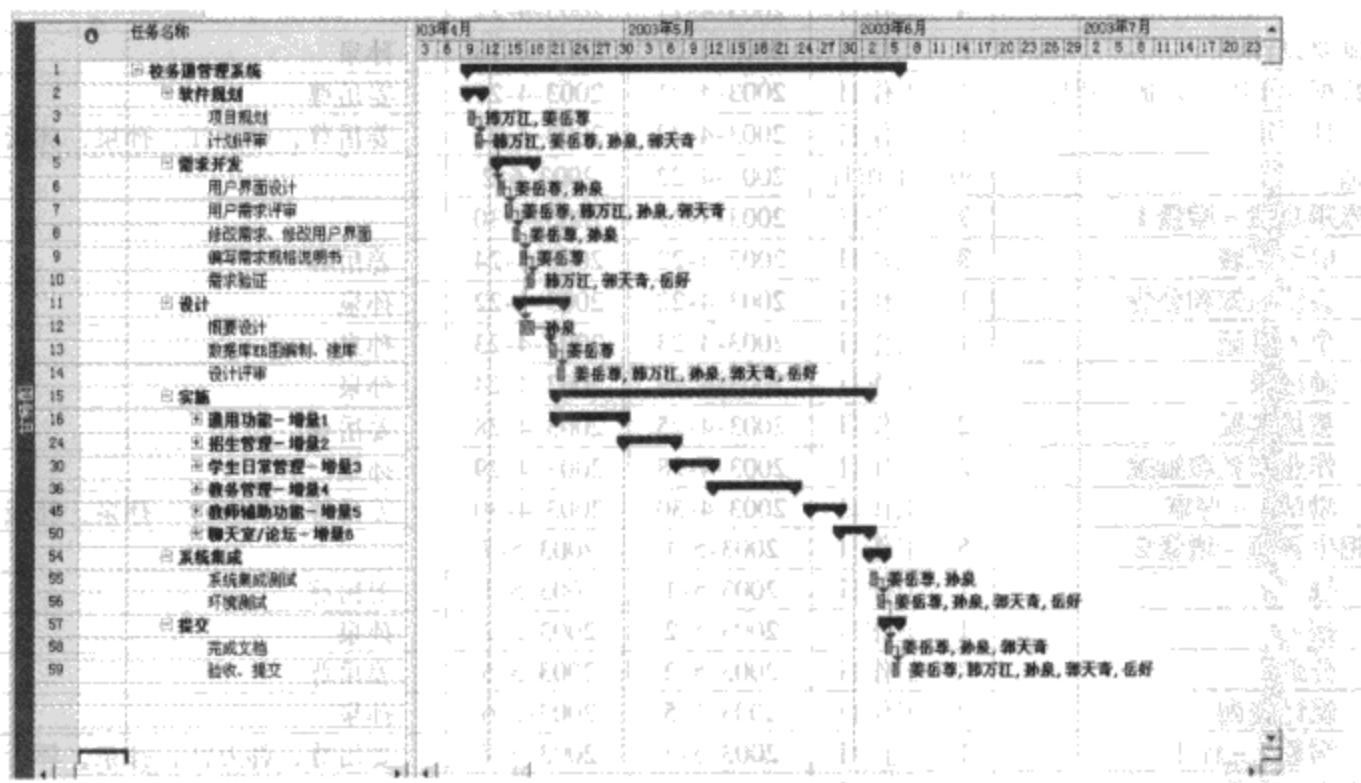
任务名称	工期	开始时间	结束时间	资源
校务通管理系统	42 工作日	2003-4-10	2003-6-6	
软件规划	2 工作日	2003-4-10	2003-4-11	
项目规划	1 工作日	2003-4-10	2003-4-10	韩万江, 姜岳尊
计划评审	1 工作日	2003-4-11	2003-4-11	韩万江, 姜岳尊, 孙泉, 郭天奇
需求开发	5 工作日	2003-4-14	2003-4-18	
用户界面设计	1 工作日	2003-4-14	2003-4-14	姜岳尊, 孙泉
用户需求评审	1 工作日	2003-4-15	2003-4-15	姜岳尊, 韩万江, 孙泉, 郭天奇
修改需求、修改用户界面	1 工作日	2003-4-16	2003-4-16	姜岳尊, 孙泉
编写需求规格说明书	1 工作日	2003-4-17	2003-4-17	姜岳尊
需求验证	1 工作日	2003-4-18	2003-4-18	韩万江, 郭天奇, 岳好
设计	4 工作日	2003-4-17	2003-4-22	
概要设计	2 工作日	2003-4-17	2003-4-18	孙泉
数据库 ER 图编制、建库	1 工作日	2003-4-21	2003-4-21	姜岳尊
设计评审	1 工作日	2003-4-22	2003-4-22	姜岳尊, 韩万江, 孙泉, 郭天奇, 岳好
实施	30 工作日	2003-4-22	2003-6-2	
通用功能 - 增量 1	7 工作日	2003-4-22	2003-4-30	
电子课表	3 工作日	2003-4-22	2003-4-24	姜岳尊
会议通知和公告	1 工作日	2003-4-22	2003-4-22	孙泉
个人日记	1 工作日	2003-4-23	2003-4-23	孙泉
通讯录	1 工作日	2003-4-24	2003-4-24	孙泉
教师答疑	2 工作日	2003-4-25	2003-4-28	姜岳尊
作业布置和批改	2 工作日	2003-4-28	2003-4-29	孙泉
增量 1 - 评审	1 工作日	2003-4-30	2003-4-30	姜岳尊, 韩万江, 孙泉, 郭天奇, 岳好
招生管理 - 增量 2	5 工作日	2003-5-1	2003-5-7	
报名	1 工作日	2003-5-1	2003-5-1	姜岳尊
招生	1 工作日	2003-5-2	2003-5-2	孙泉
分班	2 工作日	2003-5-2	2003-5-5	姜岳尊
统计查询	2 工作日	2003-5-5	2003-5-6	孙泉
增量 2 - 评审	1 工作日	2003-5-7	2003-5-7	姜岳尊, 韩万江, 孙泉, 郭天奇, 岳好
学生日常管理 - 增量 3	3 工作日	2003-5-8	2003-5-12	
学生档案管理	1 工作日	2003-5-8	2003-5-8	孙泉

(续)

任务名称	工期	开始时间	结束时间	资源
学生考勤管理	1 工作日	2003-5-8	2003-5-8	姜岳尊
学生奖惩	1 工作日	2003-5-9	2003-5-9	孙泉
学生变动	1 工作日	2003-5-9	2003-5-9	姜岳尊
增量3 - 评审	1 工作日	2003-5-12	2003-5-12	姜岳尊, 韩万江, 孙泉, 郭天奇, 岳好
教务管理 - 增量4	9 工作日	2003-5-13	2003-5-23	
教师日常管理	1 工作日	2003-5-13	2003-5-13	姜岳尊
年级、班级设置	3 工作日	2003-5-13	2003-5-15	孙泉
学科设置	1 工作日	2003-5-14	2003-5-14	姜岳尊
年班级课程设计	2 工作日	2003-5-15	2003-5-16	孙泉
排课表	2 工作日	2003-5-16	2003-5-19	孙泉
考试管理	2 工作日	2003-5-19	2003-5-20	姜岳尊
评价	3 工作日	2003-5-20	2003-5-22	孙泉
增量4 - 评审	1 工作日	2003-5-23	2003-5-23	姜岳尊, 韩万江, 孙泉, 郭天奇, 岳好
教师辅助功能 - 增量5	4 工作日	2003-5-26	2003-5-29	
教师备课系统 (外包)	1 工作日	2003-5-26	2003-5-26	姜岳尊
资源库系统	1 工作日	2003-5-26	2003-5-26	孙泉
网上考试	2 工作日	2003-5-27	2003-5-28	姜岳尊, 孙泉
增量5 - 评审	1 工作日	2003-5-29	2003-5-29	姜岳尊, 韩万江, 孙泉, 郭天奇, 岳好
聊天室/论坛 - 增量6	2 工作日	2003-5-30	2003-6-2	
论坛	1 工作日	2003-5-30	2003-5-30	姜岳尊
聊天室	1 工作日	2003-5-30	2003-5-30	孙泉
实施过程评审	1 工作日	2003-6-2	2003-6-2	姜岳尊, 韩万江, 孙泉, 郭天奇, 岳好
系统集成	2 工作日	2003-6-3	2003-6-4	
系统集成测试	1 工作日	2003-6-3	2003-6-3	姜岳尊, 孙泉
环境测试	1 工作日	2003-6-4	2003-6-4	姜岳尊, 孙泉, 郭天奇, 岳好
提交	2 工作日	2003-6-5	2003-6-6	
完成文档	1 工作日	2003-6-5	2003-6-5	姜岳尊, 孙泉, 郭天奇
验收、提交	1 工作日	2003-6-6	2003-6-6	姜岳尊, 韩万江, 孙泉, 郭天奇, 岳好

## (2) 项目甘特图

图 3-30 和图 3-31 (实施任务的展开) 是项目进度计划的甘特图。



实施部分的计划展开如下：

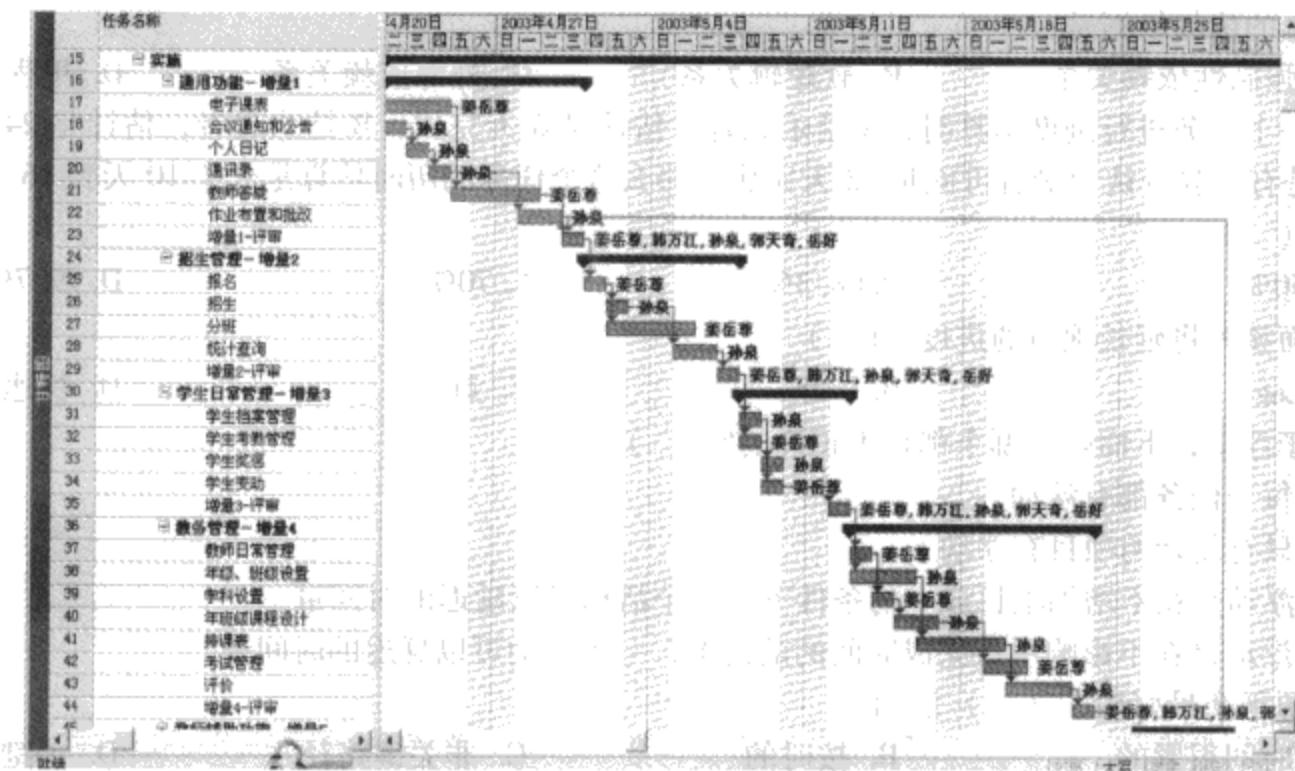


图 3-31 实施部分进度甘特图

### (3) 里程碑

本项目也需要里程碑计划，因为一些人员（例如高层领导）可能更加关注里程碑的进展，里程碑的图示如表 3-10。

## 3.9 小结

项目进度计划的开发是贯穿项目始终的，可以渐进式进行的。网络图是非常有用的进度表达方式，网络图主要有 PDM 图、ADM 图和 CDM 图，网络图中可以通过正推法确定各个活动的最早开始时间和最早完成时间，通过逆推法确定各个活动的最晚开始时间和最晚结束时间。进度历时估算的基本方法有定额计算法和经验导出模型、工程评估评审技术（PERT）、关键路径法（CPM）、专家估算方法、类推估算方法、模拟估算方法、基于承诺的进度估算方法、Jones 的一阶估算准则等。进度编制的主要方法有关键路径法、时间压缩法、资源调整尝试法、关键链法。编制项目计划的基本步骤是：首先创建项目，然后分解项目的任务，确定任务之间的关系，分配任务需要的资源，确定任务的工期，以及相应的成本，最后将计划进行优化后形成基准计划。

## 3.10 练习题

### 一、选择题

1. 快速跟进是指（ ）。
  - A. 采用并行执行任务，加速项目进展
  - B. 用一个任务取代另外的任务
  - C. 如有可能，减少任务数量
  - D. 减轻项目风险
2. 赶工一个任务时，你应该关注（ ）。
  - A. 尽可能多的任务
  - B. 非关键任务
  - C. 加速执行关键路径上的任务
  - D. 通过成本最低化加速执行任务
3. “软件编码完成之后，我才可以对它进行软件测试”，这句话说明了哪种依赖关系？（ ）
  - A. 强制性依赖关系
  - B. 软逻辑关系
  - C. 外部依赖关系
  - D. 里程碑

表 3-10 里程碑图示

事件	时间
增量 1 - 通用功能	2003-4-30
增量 2 - 招生管理	2003-5-7
增量 3 - 学生日常管理	2003-5-13
增量 4 - 教务管理	2003-5-23
增量 5 - 教师辅助功能	2003-5-29
增量 6 - 聊天室/论坛	2003-6-2
提交	2003-6-6

4. 如果用户提供的环境设备需要 5 月 10 日到位，所以环境测试安排在 5 月 10 日以后，这种活动安排的依赖依据是：（ ）  
 A. 强制性依赖关系      B. 软逻辑关系      C. 外部依赖关系      D. 里程碑
5. 对一个任务进行进度估算时，A 是乐观者，估计是 6 天完成，B 是悲观者，估计是 24 天完成，C 是有经验者认为最有可能是 12 天完成，那么这个任务的历时估算介于 10 天到 16 天的概率是（ ）。  
 A. 50%      B. 68.3%      C. 70%      D. 99.7%
6. 下面哪项将延长项目的进度？（ ）  
 A. Lag      B. Lead      C. 赶工      D. 快速跟进
7. 关于浮动，下面除了哪个都是不正确的？（ ）  
 A. 每个任务都有浮动  
 B. 只有复杂的项目有浮动  
 C. 浮动是在不增加项目成本的条件下，一个活动可以延迟的时间量  
 D. 浮动是在不影响项目完成时间的前提下，一个活动可以延迟的时间量
8. 资源平衡最好用于（ ）活动。  
 A. 时间很紧的      B. 按时的      C. 非关键路径      D. 关键路径
9. 下面哪项可以决定进度的灵活性？（ ）  
 A. PERT      B. Total Float      C. ADM      D. 赶工

二、根据表 3-11 和图 3-32，采用 PERT 方法估算，说明项目在 20 天内完成的概率的近似值？

表 3-11 PERT 方法估计项目历时

项	估计值	最乐观值	最可能值	最悲观值
任务 1		8	10	24
任务 2		1	5	9



图 3-32 项目网络图

## ■ 软件项目成本计划

成本管理也是项目管理核心三角形的一边，本章开始项目成本估算，进入本书路线图的项目成本计划，如图 4-1 所示。

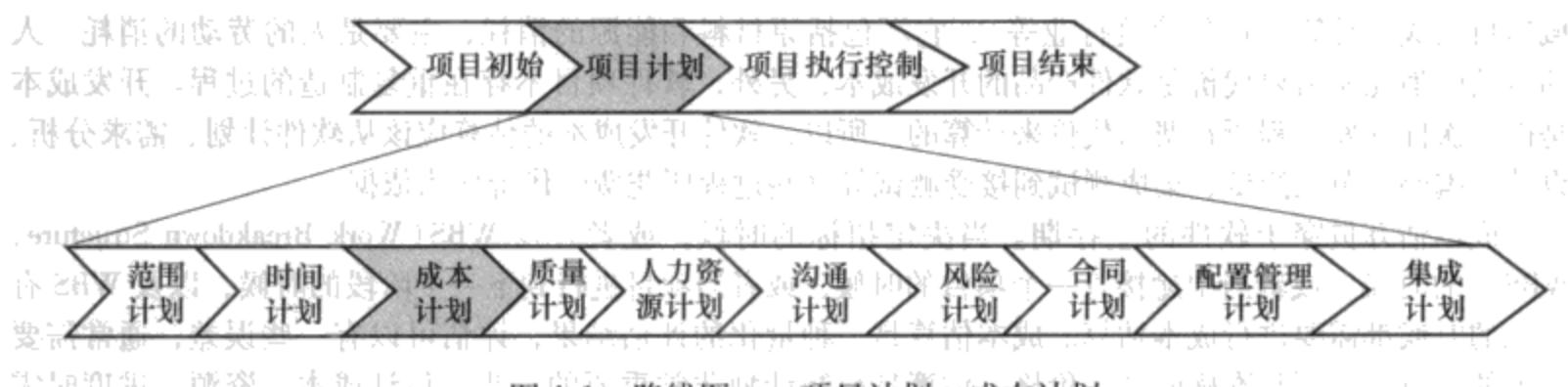


图 4-1 路线图——项目计划 - 成本计划

### 4.1 关于成本估算

估算一般可能不是非常的精确，尤其软件更是如此，软件项目中存在太多的不确定性，而且，在项目初期，人们对需求和技术的了解还不是很透彻。可能没有人可以精确的回答“这个估算是否正确吗？”这样一个问题。有效的软件估算，特别是软件成本估算，一直是软件工程和软件项目管理中最具有挑战、最为重要的问题之一。软件项目估算永远不会是一门精确的科学，但将良好的历史数据与系统化的技术结合起来能够提高估算的精确度。

对于估算既没有特效的办法也没有很通用的模型，项目经理可以根据以前的项目经验和验证过的指南来提高精度。无论是成本，进度，还是预算都可以采用渐进的方式逐步完善。

为了得到一个相对准确的估算结果，项目管理者应该系统的学习相关的成本管理过程。成本估算时对完成项目所需费用的估计和计划，是项目计划中的一个重要组成部分。要实行成本控制，首先要进行成本估算。理想的是，完成某项任务所需费用可根据历史标准估算。但由于项目和计划变化多端，将以前的活动与现实对比存在一定的差距。费用的信息，不管是否根据历史标准，都只能将其作为一种估算。而且，在费时较长的大型项目中，还应考虑到今后几年的职工工资结构是否会发生变化，今后几年其他费用的变化如何。所以，成本估算显然是在一个无法以高度可靠性预计的环境下进行。

IT 项目尤其是软件项目的成本管理一直没有一个好的口碑，成本超支是很常见的现象。成本估算不准有很多原因，有主观的原因，也有客观的原因，例如有的人认为估算本身就是估计，可以不准确的，有的人对需求缺乏了解的，当然当今信息技术的飞速发展，使得项目会不断采用新的技术和新的商业流程，这都加大了估算的难度。

随着软件系统规模的不断扩大和复杂程度的日益加大，从 20 世纪 60 年代末期开始，出现了以大量软件项目进度延期、预算超支和质量缺陷为典型特征的软件危机，至今仍频繁发生。软件成本估算不足和需求不稳定是造成软件项目失控最普遍的两个原因。所以，成本估算对于项目管理者是一项必需和重要的技能。

#### 4.1.1 项目规模与成本的关系

软件项目规模即工作量，是从软件项目范围中抽出的软件功能，然后确定每个软件功能所必须执行的一系列软件工作任务。软件项目成本是指完成软件项目规模相应付出的代价，是待开发的软件项

目需要的资金。代码行 (Lines Of Code, LOC)、功能点、人天、人月、人年等都是规模的单位。成本一般采用货币单位，例如人民币和美元等。

项目规模 (effort) 是成本的主要因素，一般来说，项目的规模估算和成本估算同时进行的，而且有时对这两个概念不做区分，规模确定了就可以确定项目的成本。例如，如果一个软件项目的规模是 20 人月，而企业的人力成本参数是 3 万/人月，则项目的成本是 60 万。

#### 4.1.2 成本估算的定义

软件的成本估算也是成本管理的核心，是预测开发一个软件系统所需要的总工作量的过程。软件开发成本是指软件开发过程中所花费的工作量及相应的代价。软件不同于其他物理产品的成本和其他领域项目的成本计算（比如建筑行业等），它不包括原材料和能源的消耗，主要是人的劳动的消耗。人的劳动的消耗所需要代价是软件产品的开发成本，另外，软件项目不存在重复制造的过程。开发成本是以一次性开发过程所花费的代价来计算的。所以，软件开发成本的估算应该从软件计划、需求分析、设计、编码、单元测试、集成测试到接受测试等这些过程所花费的代价作为依据。

成本估算贯穿于软件的生存期。当决定招标的时候，或者开发 WBS (Work Breakdown Structure, WBS) 的时候，或者当中途接管一个项目的时候，或者当项目进行到下一个阶段的时候，以及 WBS 有变化的时候都需要进行成本估算。成本估算是一种量化的评估结果，评估可以有一些误差，通常需要一定的调节，它不是准确的产品价格。估算也是编制计划非常重要的一步，估计成本、资源、进度时需要经验，历史信息等。如果我们建好 99 座相同的房子以后，可以很准确地估计第 100 座的成本和进度。这是因为我们已经掌握了很丰富的经验，但是更多的情况是我们所做的推测是缺乏经验的，尤其，对于当今信息时代的项目，有好多的未知领域，所以进行成本估算可能有一定的风险。

成本估算也是项目成本管理的核心，通过成本估算，分析并确定项目的估算成本，并以此为基础进行项目成本预算，开展项目成本控制等管理活动。成本估算也是进行项目规划相关活动的基础。

### 4.2 成本估算的过程

企业经营的最直接目标就是利润，而成本与利润的关系最为密切。软件开发项目中的成本指完成项目需要的所有费用，包括人力成本、材料成本、设备租金、咨询费用、日常费用等。项目结束时的最终成本应控制在预算内。软件企业的经济性基础就是利润，而利润的最直接决定因素就是成本。项目作为软件企业的最基本利润单位，其成本就成为软件企业成本的最基本构成单位。成本管理就是确保项目在预算范围之内的管理过程，包括成本估算、成本预算、成本控制等过程。

成本估算涉及计算完成项目所需各资源成本的近似值。当一个项目按合同进行时，应区分成本估算和定价这两个不同意义的词。成本估算涉及的是对可能数量结果的估计，即乙方为提供产品或服务的花费是多少。而定价是一个商业决策，即乙方为它提供的产品或服务索取多少费用，而成本估算只是定价要考虑的因素之一。

成本估算应当考虑各种不同的成本替代关系。例如，在设计阶段增加额外工作量可减少开发阶段的成本。成本估算过程必须考虑增加的设计工作所多花的成本能否被以后的节省所抵消。

成本估算也是针对资源进行的，因项目性质的不同可以进行多次，对于独特的项目产品所进行的逐步细化需要进行几次成本估算。

由于影响软件成本的因素太多（例如人、技术、环境、政治等），软件估算仍然是很不成熟的技术，一些方法只能作为借鉴，更多的时候需要经验。目前没有一个估算方法或者成本估算模型可以适用于所有的软件类型和开发环境。

#### 4.2.1 估算输入

估算的输入一般包括如下几项：

- 1) 需求或者 WBS：根据估算的不同阶段，有不同的输入可用于成本估算，以确保所有工作均一被估计进成本了。

- 2) 资源要求(资源编制计划): 它可以让项目组掌握资源需要和分配的情况。
- 3) 资源消耗率, 即资源单价, 成本估算时必须知道每种资源单价(例如: 每小时人员费用等)以计算项目成本。如果实际单价不知道, 那么必须要估计单价本身。它是非常重要的一项输入, 如人员成本: 100 元/小时, 某资源消耗/小时等等, 是估算的基础。
- 4) 进度规划: 它是主要的项目活动时间的估计, 活动时间估计会影响项目成本估计。
- 5) 历史项目数据: 它是以往项目的数据, 包括规模、进度、成本等等, 是项目估算的主要参考。一个成熟的软件企业应该建立完善的项目档案, 记录先前项目的信息。
- 6) 学习曲线: 它是项目组学习某项技术或者工作的时间, 当一件事情被重复的时候, 完成这件事情的时间将缩短, 业绩会以一定的百分比提高。

#### 4.2.2 估算处理

规模成本估算也是项目各活动所需资源消耗的定量估算, 规模成本估算主要是对各种资源的估算, 包括人力资源、设备、资料等等。在估算过程中采用一定的估算方法进行, 而成本包括直接成本和间接成本。直接成本是与开发的具体项目直接相关的成本, 例如人员的工资、材料费、外包外购成本等, 包括开发成本、管理成本、质量成本等。间接成本不能归属于一个具体的项目, 是企业的运营成本, 可以分摊到各个项目中。例如房租、水电、保安、员工福利、保险、税收、行政管理、市场费用。常用的估算方法在下一节将做详细讲述。

#### 4.2.3 估算输出

规模成本估算的结果可以以简略或详细的形式表示。对项目所需的所有资源的成本均需加以估计, 这包括(但不局限于)劳力、材料和其他内容(如考虑通货膨胀或成本余地)。估算通常以货币单位表达, 例如元、法郎、美元等, 这个估算便是成本估算的结果; 也可用人月、人天或人小时这样的单位, 这个估算结果便是项目规模估算的结果。为便于成本的管理控制, 有时成本估算要用复合单位。成本估算是一个不断优化的过程。随着项目的进展和相关详细资料的不断出现, 应该对原有成本估算做相应的修正, 在有些应用项目中提出了何时应修正成本估算, 估算应达到什么样的精确度。

估算文件包括项目需要的资源、资源的数量、质量标准、估算成本等信息, 单位一般是货币单位, 也可以是规模单位, 然后转换为货币单位。估算结果可以用一个范围表示, 例如: \$10 000 ± \$1000 表示估计成本在 \$9000 和 \$11 000 之间。

估算说明应该包括工作范围的描述, 这通常可由 WBS 获得; 说明估算的基础和依据, 即确认估算是否合理, 说明估算怎样产生的。确认为成本估算所作的任何假设的合理性以及估算的误差变动等。它能提供如何估算成本的一个较好的说明。

### 4.3 成本估算的方法

在项目管理过程中, 为了使时间、费用和工作范围内的资源得到最佳利用, 人们开发出了不少成本估算方法, 以尽量得到较好的估算。这里介绍以下几种常用的成本估算方法: 代码行、功能点、类比(自顶向下)估算法、自下而上估算法、参数估算法、专家估算法、猜测估算法等等。

#### 4.3.1 代码行、功能点及对象点

尽管代码行、功能点是表达软件项目规模的单位, 但是有时也称它们是一种软件规模估算方法。

##### 1. 代码行

代码行(Loc)技术是比较简单的定量估算软件规模的方法。这种方法依据以往开发类似产品的经验和历史数据, 估计实现一个功能所需要的源程序行数。代码行是从软件程序量的角度定义项目规模的。使用代码行作为规模单位的时候, 要求功能分解足够详细, 而且有一定的经验数据, 采用不同的开发语言, 代码行可能不一样。当然, 也应该掌握相关的比例数据等, 例如生产率 Loc/PM(人月)和 Loc/hour 等。

代码行是在软件规模度量中最早使用也是最简单的方法, 在用代码行度量规模时, 常会被描述为

源代码行 (Source Lines Of Code, SLOC) 或者交付源指令 (Delivered Source Instruction, DSI), 目前成本估算模型通常采用非注释的源代码行。

代码行技术的主要优点体现在代码是所有软件开发项目都有的“产品”，而且很容易计算代码行数。但是代码行也存在许多问题：

1) 对代码行没有公认的标准定义，例如，最常见的计算代码行时的分歧有空代码行、注释代码行、数据声明、复用的代码，以及包含多条指令的代码行等。在 Jones 的研究中发现，对同一个产品进行代码行计算，不同的计算方式可以带来 5 倍之大的差异。

2) 代码行数量依赖于所用的编程语言和个人的编程风格，因此，计算的差异也会影响用多种语言编写的程序规模，进而也很难对不同语言开发的项目的生产率进行直接比较。

3) 在项目早期，需求不稳定、设计不成熟、实现不确定的情况下很难准确地估算代码量。

4) 代码行强调编码的工作量，只是项目实现阶段的一部分。

## 2. 功能点

### (1) 基本概念

功能点 (Function Point, FP) 是用系统的功能数量来测量其规模，它以一个标准的单位来度量软件产品的功能，与实现产品所使用的语言和技术没有关系的。因为功能点是从功能的角度来度量系统，因此它同所使用的技术无关。不考虑开发语言、开发方法、使用的硬件平台，因此系统的功能点数量会保持恒定，唯一的变化就是交付一套功能点所需要的工作量不同。1979 年 IBM 的 Alan Albrech 首先开发了计算功能点的方法，功能点提供一种解决问题的结构化技术，它是一种将系统分解为较小组件的方法，使系统能够更容易被理解和分析。在功能点分析中，系统被分为 5 类组件和一些常规系统特性。前三类组件是：外部输入 (External Input, EI)、外部输出 (External Output, EO) 和外部查询 (External Inquiry, EQ)。这些组件中的每一个组件都处理文件，因此他们被称为“事务” (transaction)。另外两类组件是内部逻辑文件 (Internal Logical File, ILF) 和外部接口文件 (External Interface File, EIF)，它们是构成逻辑信息的数据存储之地。使用功能点方法需要评估产品所需要的内部基本功能和外部基本功能，然后根据技术复杂度因子 (权) 对它们进行量化，产生规模的最终结果。

功能点计算公式是  $FP = UFC * TCF$ ，其中：

- UFC(Unadjusted Function Component)：表示未调整功能点计数。
- TCF(Technical Complexity Factor,)：表示技术复杂度因子 (即调整系数)。

### (2) 未调整功能点计数

在计算未调整功能点计数时，应该先计算五类功能组件的计数项。对于计算机系统来说，同其他计算机系统交互是一个非常普遍的事情，因此，在分类组件之前必须划出每个被度量的系统的边界。必须要从用户的角度来划边界。简而言之，边界表明了被度量的系统或应用同外部系统或应用之间的界限。一旦边界被建立，则组件就能够被分类、分级和评分。

下面介绍一下这五类组件。

1) 内部逻辑文件。如图 4-2 所示，内部逻辑文件是用户可以识别的一组逻辑相关的数据，而且完全存在于应用的边界之内，并且通过外部输入维护，是逻辑主文件 (即数据的一个逻辑组合，它可能是大型数据库的一部分或是一个独立的文件) 的数目。

2) 外部输入。如图 4-3 所示，外部输入给软件提供面向应用的数据项 (如屏幕、表单、对话框、控件、文件等)；在这个过程中，数据穿越外部边界进入到系统内部。这里的数据可能来自于输入界面，也可以来自于另外的应用。数据将被用来维护一个或者多个内部逻辑文件。数据既可能是控制信息，也可能是业务逻辑信息。如果数据是控制信息，则它不会更新内部逻辑文件。图 4-3 展现了一个更新两个内部逻辑文件的简单的外部输入。输入不同于查询，后者单独计数，不计入输入项数中。

3) 外部输出。如图 4-4 所示，外部输出向用户提供面向应用的信息，例如，报表和出错信息等。报表内的数据项不单独计数。在这个过程中，派生数据由内部穿越边界传送到外部。另外，一个外部输出可以更新内部逻辑文件。数据生成报表或者传送给其他应用的数据文件。这些报表或者文件从一个或者多个内部逻辑文件以及外部接口文件生成。

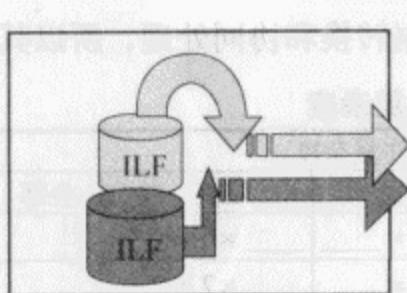


图 4-2 内部逻辑文件

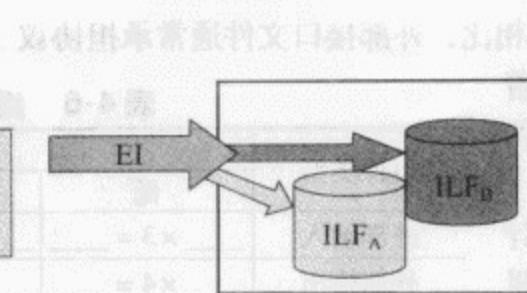


图 4-3 外部输入

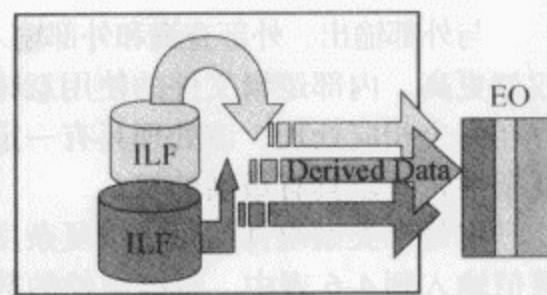


图 4-4 外部输出

4) 外部查询。外部查询即是一次联机输入，它导致软件以联机输出方式产生某种即时响应。这个过程中的输入和输出部分都导致数据从一个或者多个内部逻辑文件或外部接口文件中提取出来。输入过程不能更新任何内部逻辑文件，并且输出端不能包括任何派生数据。

5) 外部接口文件。外部接口文件是用户可以识别的一组逻辑相关数据，这组数据只能被引用。是机器可读的全部接口（例如，磁盘或磁带上的数据文件）数量，用这些接口把信息传送给另一个系统。数据完全存在于应用的外部，并且由另一个应用维护。外部接口文件是另外一个应用的内部逻辑文件。

当组件被归为以上 5 类主要组件中的一类之后，就要为之指定级别，所有组件都被定级为高、中、低三个级别。对于事务组件来说，它们的级别取决于被更新或引用的文件个数以及数据元素类型（Data Element Types, DET）的个数。对于内部逻辑文件和外部接口文件来说，它们的级别取决于记录元素类型（Record Element Types, RET）和数据元素类型的个数。记录元素类型是 ILF 或者 EIF 中用户能够识别的数据元素小组。一个数据元素类型是一个用户可识别的、唯一性的、非递归的域。

表 4-1、表 4-2、表 4-3 用来帮助对事务组件进行定级。例如：引用或者更新 2 个引用的文件类型（File Types Referenced, FTR），并且有 7 个数据元素的 EI 将被定级为中级，相关的级数是 4。这里，FTR 是被引用或更新的内部逻辑文件和被引用的外部接口文件的综合。

表 4-1 外部输入定级表

引用的文件 类型个数	数据元素		
	1~4	5~15	>15
0~1	低	低	低
2	低	中	高
≥3	中	高	高

表 4-2 外部输出和外部查询共用的定级表

引用的文件 类型个数	数据元素		
	1~5	6~19	>19
0~1	低	低	中
2~3	低	中	高
>3	中	高	高

表 4-3 外部输入、外部输出和外部查询定级取值表

级数	值		
	EO	EQ	EI
低	4	3	3
中	5	4	4
高	7	6	6

基本上，外部查询的定级同外部输出一样（低、中、高），但取值则同外部输入一样。级别取决于（综合了唯一输入和外端的）数据元素类型个数及被引用的文件类型（综合了唯一的输入和输出端）的个数。如果同一个 FTR 被输入和输出同时使用，那么它只计算一次。如果同一个数据元素类型被输入和输出同时使用，那么它也只能被计算一次。

同样复杂的外部输出产生的功能点数要比外部查询/外部输入多出 20% 到 33%，由于一个外部输出意味着产生一个有意义的需要显示的结果，因此，相应的权应该比外部查询/外部输入高一些。

对于内部逻辑文件和外部接口文件来说，记录元素类型和数据元素类型个数来决定它们的低、中、高级别。见表 4-4 和表 4-5，记录元素类型是一个用户可以识别的内部逻辑文件或者外部接口文件中的数据元素小组。数据元素类型是内部逻辑文件或者外部接口文件中单一的、用户可识别的、非递归的域。

表 4-4 内部逻辑文件或者外部接口文件的定级表

记录元素类型	数据元素		
	1~19	20~50	>50
1	低	低	中
2~5	低	中	高
>5	中	高	高

表 4-5 内部逻辑文件或者外部接口文件级数取值表

级数	值	
	ILF	EIF
低	7	5
中	10	7
高	15	10

与外部输出、外部查询和外部输入相比，外部接口文件通常承担协议、数据转换和协同处理，所以其权就更高。内部逻辑文件的使用意味着存在一个相应处理，该处理具有一定的复杂性，所以具有最高的权。

将每个类型组件的每一级复杂度计算值输入到表 4-6 中。每级组件的数量乘以所示的级数（numeric rating）得出定级的值（rated value）。表中每一行的定级值相加得出每类组件的定级值之和。这些定级值之和再相加，得出全部组件的功能点值。这个和被填写到未调整的功能点数一栏中。

### (3) 技术复杂度因子

功能点计算的下一步是评估影响系统功能规模的 14 个技术复杂度因子（调整系数值），即计算技术因素对软件规模的综合影响程度，技术复杂度因子取决于 14 个通用系统特性（General System Characteristics, GSC）。这些系统特性用来评定功能点应用的通用功能的级别。每个特性有相关的描述以帮助确定这个系统特性的影响程度。影响程度的取值范围从 0 到 5，从没有影响到有强烈影响。表 4-7 是 14 个技术复杂度因子，每个因子的取值范围是 0~5，表 4-8 显示了每个因子取值范围的情况。技术复杂度因子的计算公式为：

$$TCF = 0.65 + \left[ \sum_{i=1}^{14} Fi / 100 \right]$$

其中，TCF 表示技术复杂度因子； $Fi$  为每个通用系统特性的影响程度； $i$  代表每个通用系统特性，取值 1~14； $\Sigma$  表示所有 14 项通用系统特性的和。

表 4-7 技术复杂度因子

通用特性		描述
F1	数据通信	多少个通信设施在应用或系统之间辅助传输和交换信息
F2	分布数据处理	分布的数据和过程函数如何处理
F3	性能	用户要求响应时间或者吞吐量吗
F4	硬件负荷	应用运行在的硬件平台工作强度如何
F5	事务频度	事务执行的频率（天、周、月）如何
F6	在线数据输入	在线数据输入率是多少
F7	终端用户效率	应用程序设计考虑到终端用户的效率吗
F8	在线更新	多少内部逻辑文件被在线事务所更新
F9	处理复杂度	应用有很多的逻辑或者数据处理吗
F10	重用性	被开发的应用要满足一个或者多个用户需要吗
F11	易安装性	升级或者安装的难度如何
F12	易操作性	启动、备份、恢复过程的效率和自动化程度如何
F13	跨平台性	应用被设计、开发和支持被安装在多个组织的多个安装点（不同的安装点的软硬件平台环境不同）吗
F14	可扩展性	应用被设计、开发以适应变化吗

一旦所有 14 个 GSC 项的评分值都被确定下来，则通过上面的调整公式可以计算出技术复杂度因子。

诸如事务频度、终端用户效率、在线更新、可复用性等项通常在 GUI 应用中的评分比传统应用要高。另一方面，性能、硬件负荷、跨平台性等项在 GUI 应用中的评分比传统应用要低。

$Fi$  的取值范围是 0~5，见表 4-8，所以根据公式可知技术复杂度因子 TCF 取值范围是 0.65~1.35。

表 4-6 组件的复杂度

组件类型	组件复杂度			
	低	中	高	全部
外部输入	$\_\_ \times 3 = \_\_$	$\_\_ \times 4 = \_\_$	$\_\_ \times 6 = \_\_$	
外部输出	$\_\_ \times 4 = \_\_$	$\_\_ \times 5 = \_\_$	$\_\_ \times 7 = \_\_$	
外部查询	$\_\_ \times 3 = \_\_$	$\_\_ \times 4 = \_\_$	$\_\_ \times 6 = \_\_$	
内部逻辑文件	$\_\_ \times 7 = \_\_$	$\_\_ \times 10 = \_\_$	$\_\_ \times 15 = \_\_$	
外部接口文件	$\_\_ \times 5 = \_\_$	$\_\_ \times 7 = \_\_$	$\_\_ \times 10 = \_\_$	
全部未调整的功能点数				
调整系数值				
全部调整后的功能点数				

表 4-8 技术复杂度因子的取值情况

调整系数	描述
0	不存在或者没有影响
1	不显著的影响
2	相当的影响
3	平均的影响
4	显著的影响
5	强大的影响

**【例1】** 一个软件的五类功能计数项如表4-9所示，计算这个软件的功能点。

1) 计算 UFC:

按照 UFC 的计算过程计算出  $UFC = 301$ ，如表 4-10 所示。

表 4-9 软件的功能计数项

复杂度 各类计数项	简单	一般	复杂
外部输入	6	2	3
外部输出	7	7	0
外部查询	0	2	4
外部接口文件	5	2	3
内部逻辑文件	9	0	2

表 4-10 计算 UFC 的结果

组件	组件复杂度		
	低	中	高
外部输入	$6 \times 3$	$2 \times 4$	$3 \times 6$
外部输出	$7 \times 4$	$7 \times 5$	$0 \times 7$
外部查询	$0 \times 3$	$2 \times 4$	$4 \times 6$
外部接口文件	$5 \times 5$	$2 \times 7$	$3 \times 10$
内部逻辑文件	$9 \times 7$	$0 \times 10$	$2 \times 15$
总计	134	65	102
UFC			301

2) 计算 TCF:

假设这个软件项目所有的技术复杂度因子的值都为 3，即技术复杂影响程度都是平均程度，则  $TCF = 0.65 + 0.01(14 \times 3) = 1.07$ 。

3) 计算功能点 FP:

由于  $UFC = 301$ ，由公式  $FP = UFC \times TCF$  得出功能点  $FP = 301 \times 1.07 = 322$ ，即项目的功能点为 322。

注：尽管功能点计算方法是结构化的，但是权重的确定是主观的，另外要求估算人员要仔细地将需求映射为外部和内部的行为，必须避免双重计算。所以，这个方法也存在一定的主观性。

#### (4) 功能点与代码行的转换

功能点可以按照一定的条件转换为代码行，表 4-11 就是一个转换表，是针对各种语言的转换率，它是根据经验的研究得出来的。

#### (5) 功能点分析总结

功能点作为度量软件规模的方法已经被越来越广泛地接受。了解软件规模是了解软件产生率的关键。没有可靠的规模度量方法，则相关的生产率（功能点数/每月）变化或者相关的质量（缺陷/功能点）的变化就不能被统计出来。如果相关的生产率和质量的变化能随时统计和策划，则组织可以将注意力集中到组织的强项和弱项上。更为重要的是：任何试图改进弱项的努力都可以被度量以确定其效果。

### 3. 其他方法

随着越来越多的项目开始应用组合方法来进行开发，又出现了诸如对象点、用例点等新的一类估算方法。

#### (1) 对象点

对象点（object point）技术可以应用于所有类型的软件开发，而不局限于面向对象的开发。利用功能点的基本原理，对象点方法需要投入大工作量来获取规模，例如，服务器数据表的数量、客户数据表的数量、报表（report）和屏幕（screen）中可重用的百分比等等。

要计算对象点，分析人员首先要计算应用中屏幕、报表和第三代语言组件（3GL component）数量的最可能值。在这个步骤中，对象点的计算与功能点计算类似，每个对象可以被分级为高、中、低不同的级别。

就像功能点一样，将这些加权后的数字相加即得到最后的数值。虽然没有一系列调整因子，但是对象点仍然考虑了重用。如果分析人员可以确定有百分之  $r$  的构件可以由以前的开发重用，则新的对象点会相应减少：

表 4-11 功能点到代码行的转换表  
(Garmus & David 1996)

语言	代码行/FP
Assembly	320
C	150
Cobol	105
Fortran	105
Pascal	91
ADA	71
PL/I	65
Prolog/LISP	64
SMALLTALK	21
SPREADSHEET	6

$$\text{根据重用调整后的对象点} = \frac{(\text{估算的对象点}) \times (100 - r)}{100}$$

仅当需求和设计足够详尽并可以得到可用的估算输入时，对象点才有利用价值。有分析认为，完成一个对象点估算的平均时间仅为功能点估算相应时间的 47%。然而，这个技术相对较新，因此，关于对象点的长期而成功应用的记录还较为缺乏。

### (2) 用例点

随着软件系统更多地采用统一建模语言 (Unified Modeling Language, UML) 进行开发，继代码行、功能点、对象点之后，出现了基于 UML 的规模度量方法，而基于用例 (use case) 的估算，即用例点 (Use Case Point, UCP) 估算方法则是其中具有代表性的一种。

总体来说，用例点方法通过分析用例角色、场景和不同的技术与环境因子，把它们抽象到一个等式中。该等式由多个变量组成：未调整用例点 (Unadjusted Use Case Point, UUCP)、技术复杂度因子 (Technical Complexity Factor, TCF) 和环境复杂度因子 (Environment Complexity Factor, ECF)。

UCP 的计算等式表示为：

$$UCP = UUCP \times TCF \times ECF$$

最终工作量计算等式为：

$$Effort = UCP \times ProductivityFactor。 (\text{其中 Productivity Factor 指生产率因子})$$

UCP 方法从 1993 年提出至今，有许多研究者在此基础上作了进一步的应用研究。2001 年，Nageswaran 把测试和项目管理的系数加入 UCP 等式，在其项目的实际应用中得到了更准确的估算。2005 年，Anda 对 UCP 方法步骤进行的修改结合了 COCOMO II 中针对改编软件的计算公式，使得它更适于增量开发的大型软件项目。2005 年，Carroll 在其研究中对 UCP 等式加入了一个风险系数，最后估算结果是，95% 的项目估算误差在 9% 以内。

### 4.3.2 类比估算法

类比 (analogy) 估算法是从项目的整体出发，进行类推，即估算人员根据以往的完成类似项目所消耗的总成本（或工作量），来推算将要开发的软件的总成本（或工作量），然后按比例将它分配到各个开发任务单元中，是一种自上而下的估算形式，也称为自顶向下。通常在项目的初期或信息不足时采用此方法，例如在合同期和市场招标时等。它的特点是简单易行，花费少，但是具有一定的局限性，准确性差，可能导致项目出现困难。

使用类比的方法进行估算基于实例推理 (Case-Based Reasoning, CBR) 的一种形式，即通过对一个或多个已完成的项目与新的类似项目的对比来预测当前项目的成本与进度。在软件成本估算中，当把当前问题抽象为待估算的项目时，每个实例即指已完成的软件项目。

类比估算要解决的主要问题是：如何描述实例特征，即如何从相关项目特征中抽取出最具代表性的特征；通过选取合适的相似度/相异度的表达式，评价相似程度；如何用相似的项目数据得到最终估算值。特征量的选取是一个决定哪些信息可用的实际问题，通常会征求专家意见以找出哪些可以帮助我们确认出最相似实例的特征。当选取的特征不够全面时，所用的解决方法也是使用专家意见。

对于度量相似度，目前的研究中常有两种求值方式来度量差距，即不加权的欧式距离 (unweighted Euclidean distance) 和加权的欧式距离 (weighted Euclidean distance)。对于相似度函数的定义，有一些不同的形式，但本质上是一致的。

$$distance(P_i, P_j) = \sqrt{\frac{\sum_{k=1}^n \delta(P_{ik}, P_{jk})}{n}}$$

$$\delta(P_{ik}, P_{jk}) = \begin{cases} \left( \frac{|P_{ik}, P_{jk}|}{\max_k - \min_k} \right)^2, & \text{如果 } k \text{ 是连续的} \\ 0, & \text{如果 } k \text{ 是分散的并且 } P_{ik} = P_{jk} \\ 1, & \text{如果 } k \text{ 是分散的并且 } P_{ik} \neq P_{jk} \end{cases}$$

**【例2】**一个待估算的项目工作量  $P_0$ ，与已经完成的项目  $P_1$ ,  $P_2$  有一定的相似度，比较它们的相似点如表 4-12 所示。

项目间的相似度计算如下：

表 4-12 项目  $P_0$  与项目  $P_1$  和  $P_2$  的相似点比较

项目	项目类型	编程语言	团队规模	项目规模	工作量
$P_0$	MIS	C	9	180	
$P_1$	MIS	C	11	200	1 000
$P_2$	实时系统	C	10	175	900

$P_0$ 对比 $P_1$	$P_0$ 对比 $P_2$
$\delta(P_{01}, P_{11}) = \delta(\text{MIS}, \text{MIS}) = 0$	$\delta(P_{01}, P_{21}) = \delta(\text{MIS}, \text{Real Time}) = 1$
$\delta(P_{02}, P_{12}) = \delta(\text{C}, \text{C}) = 0$	$\delta(P_{02}, P_{22}) = \delta(\text{C}, \text{C}) = 0$
$\delta(P_{03}, P_{13}) = \delta(9, 11)$ $= [(9 - 11) / (9 - 11)]^2 = 1$	$\delta(P_{03}, P_{23}) = \delta(9, 10)$ $= [(9 - 10) / (9 - 11)]^2 = 0.25$
$\delta(P_{04}, P_{14}) = \delta(180, 200)$ $= [(180 - 200) / (200 - 175)]^2 = 0.64$	$\delta(P_{04}, P_{24}) = \delta(180, 175)$ $= [(180 - 175) / (200 - 175)]^2 = 0.04$
$distance(P_0, P_1) = (1.64/4)^{0.5} = 0.64$	$distance(P_0, P_2) = 0.57$

对于工作量最后估算值的确定，有不同的方法：可以直接取最相似的项目的工作量（对应  $P_0$  工作量取 1 000）；也可以取比较相似的几个项目的工作量平均值（对应  $P_0$  工作量取  $1900/2 = 950$ ）；还可以采用某种调整策略，例如用项目的规模作调整参考，对应到例子中，可采用如下调整： $Size(P_0)/Size(P_1) = Effort(P_0)/Effort(P_1)$ ，得到  $P_0$  工作量为  $1000 \times 180/200 = 900$ 。目前，类比估算的方法，基本上采用主观推测，很少采用相似度计算的方法。

类比估算最主要的优点是比较直观，而且能够基于过去实际的项目经验来确定与新的类似项目的具体差异以及可能对成本产生的影响。其主要缺点，一是不能适用于早期规模等数据都不确定的情况；二是应用一般集中于已有经验的狭窄领域，不能跨领域应用；三是难以适应新的项目中约束条件、技术、人员等发生重大变化的情况。

#### 4.3.3 自下而上估算法

自下而上估算法是利用任务分解结构图，对各个具体工作包进行详细的成本估算，然后将结果累加起来得出项目总成本。用这种方法估算的准确度较好，通常是在项目开始以后，或者 WBS 已经确定的项目，需要进行准确估算的时候采用。这种方法的特点是估算准确。它的准确度来源于每个任务的估算情况，但是这个方法需要花费一定的时间，因为估算本身也需要成本支持，而且可能发生虚报现象。如果对每个元素的成本设定一个相应的费率，就可以对整个开发的费用得到一个自下而上的全面期望值。例如表 4-13，是采用自下而上估算法计算的过程结果，首先根据任务分解的结果，评估每个子任务的成本，然后逐步的累加，最后得出项目的总成本。

表 4-13 自下而上估算法估算项目的成本

子任务	人力	时间 (月)	成本 (万元)	总计 (万元)
项目准备阶段	M: 2/D: 8/Q: 1	0.5	16.5	
设计阶段	M: 2/D: 8/Q: 2/S: 1	1	39	
基础模块开发： 公共控制子系统	M: 2/D: 15/Q: 2/S: 1	1.5	90	145.5
中央会计子系统				
客户信息子系统				
基本功能模块开发： 账户管理子系统	M: 2/D: 12/Q: 2/S: 1	0.25	12.75	
出纳管理子系统	M: 2/D: 18/Q: 2/S: 1	0.5	34.5	126
凭证管理子系统	M: 2/D: 8/Q: 2/S: 1	0.25	9.75	
会计核算子系统	M: 2/D: 18/Q: 2/S: 1	0.5	34.5	
储蓄子系统	M: 2/D: 18/Q: 2/S: 1	0.5	34.5	
扩展功能模块开发： 同城业务子系统	M: 2/D: 15/Q: 2/S: 1	0.25	15	
联行业务子系统	M: 2/D: 18/Q: 2/S: 1	0.5	34.5	
内部清算子系统	M: 2/D: 12/Q: 2/S: 1	0.25	12.75	189.75
固定资产管理系统	M: 2/D: 10/Q: 2/S: 1	0.25	11.25	
信贷管理子系统	M: 2/D: 18/Q: 2/S: 1	0.5	34.5	
一卡通业务子系统	M: 2/D: 18/Q: 2/S: 1	0.5	34.5	
中间业务子系统	M: 2/D: 12/Q: 2/S: 1	0.25	12.75	
金卡接口模块	M: 2/D: 18/Q: 2/S: 1	0.5	34.5	
现场联调	M: 2/D: 10/Q: 2	1	42	42
总成本			503.25	

#### 4.3.4 参数模型估算法

参数模型估算法，也称为算法模型或者经验导出模型，是一种使用项目特性参数建立数学模型来估算成本的方法，是通过大量的项目数据进行数学分析导出的模型，是一种统计技术。数学模型可以简单也可以复杂。一个模型不能适合所有的情况，只能适应某些特定的项目情况。其实，目前没有一种模型或者方法能适应所有项目。

参数模型估算法的基本思想是：找到软件工作量的各种成本影响因子，并判定它对工作量所产生影响的程度是可加的、乘数的还是指数的，以期得到最佳的模型算法表达形式。当某个因子只影响系统的局部时，我们一般说它是可加的，例如，如果我们给系统增加源指令、功能点实体、模块、接口等，大多只会对系统产生局部的可加的影响。当某个因子对整个系统具有全局性的影响时，我们则说它是乘数的或指数的，例如，增加服务需求的等级或者不兼容的客户等。

相当一部分软件估计模型是近 20 年开发的，这些模型中很多是专用模型，因此不能通过模型结构来进行对比，理论和实践决定了这些模型的功能形式。

一般来说，参数模型提供工作量（规模）的直接估计，典型的参数模型是通过过去项目数据进行回归分析得出的回归模型。基于回归分析的模型可以分为两类，一类是静态单变量模型，一类是动态多变量模型。

##### (1) 静态单变量模型

这类模型的总体结构形式如下：

$$E = a + b \times S^c$$

其中， $E$  是以人月表示的工作量； $a$ 、 $b$ 、 $c$  是经验导出的系数； $S$  为估算变量，是主要的输入参数（通常是 LOC，FP 等）。下面给出几个典型的静态单变量模型。

面向 LOC 的估算模型：

- Walston-Felix(IBM) 模型： $E = 5.2 \times (\text{KLOC})^{0.91}$
- Balley-Basili 模型： $E = 5.5 + 0.73 \times (\text{KLOC})^{1.16}$
- Boehm（基本 COCOMO）模型： $E = 3.2 \times (\text{KLOC})^{1.05}$
- Doty 模型（在  $\text{KLOC} > 9$  的情况下）： $E = 5.288 \times (\text{KLOC})^{1.047}$

面向 FP 的估算模型：

- Albrecht and Gaffney 模型： $E = -12.39 + 0.0545\text{FP}$
- Matson, Barnett 模型： $E = 585.7 + 15.12\text{FP}$

##### (2) 动态多变量模型

动态多变量模型也称为软件方程式，它是根据从 4000 多个当代软件项目中收集的生产率数据推导出来的。该模型把工作量看作是软件规模和开发时间这两个变量的函数。

$$E = (\text{LOC} \times B^{0.333}/P)^3 \times (1/t)^4$$

其中： $E$  是以人月或人年为单位的工作量。 $t$  是以月或年为单位的项目持续时间。 $B$  是特殊技术因子，它随着对测试、质量保证、文档及管理技术的需求的增加而缓慢增加。对于较小的程序（ $\text{KLOC} = 5 \sim 15$ ）， $B = 0.16$ ；对于超过 70KLOC 的程序， $B = 0.39$ 。 $P$  是生产率参数，它反映了下述因素对工作量的影响：

- 总体过程成熟度及管理水平。
- 使用良好的软件工程实践的程度。
- 使用的程序设计语言的级别。
- 软件环境的状态。
- 软件项目组的技术及经验。
- 应用系统的复杂程度。

开发实时嵌入式软件时， $P$  的典型值为 2000；开发电信系统和系统软件时， $P = 10\,000$ ；对于商业应用系统来说， $P = 28\,000$ 。可以从历史数据导出适用于当前项目的生产率参数值。

一般来说，参数模型估算方法适合比较成熟的软件企业，这些企业积累了丰富的历史项目数据，

并可以归纳出成熟的项目估算模型。它的特点是比较简单，而且也比较准确，一般参考历史信息，重要参数必须量化处理，根据实际情况，对参数模型按适当比例调整。但是如果模型选择不当或者数据不准，也会导致偏差。

下面我们主要讲 COCOMO 模型（Boehm 模型）。COCOMO 模型是世界上应用最广泛的参数型软件成本估计模型，由 B. W. Boehm 在 1981 年出版的《软件工程经济学》（Software Engineering Economics）中首先提出，其本意是“结构化成本模型”（Constructive Cost Model）。

由于 COCOMO 模型的可用性，并且没有版权问题，所以得到了非常广泛的应用。在 COCOMO 模型的发展中，无论是最初的 COCOMO 81 模型，还是 20 世纪 90 年代中期提出的逐步成熟完善的 COCOMO II，所解决的问题都具有当时软件工程实践的代表性。作为目前应用最广泛、得到学术界与工业界普遍认可的软件估算模型之一，已经发展到了一组模型套件（包含了软件成本模型、软件扩展与其他独立估算模型 3 大类），形成了 COCOMO 模型系列，也给基于算法模型的方法提供了一个通用的公式：

$$PM = A \times (\sum Size)^B \times \prod (EM)$$

其中，PM 为工作量，通常表示为人月（person months）；A 为校准因子（calibration factor）；Size 为对工作量呈可加性影响的软件模块的功能尺寸的度量；B 为对工作量呈指数或非线性影响的比例因子（scale factor）；EM 为影响软件开发工作量的工作量乘数（effort multiplicative）。

### 1. COCOMO 81

COCOMO 81 有三个等级的模型：（1）基本（basic）模型，在项目相关信息极少的情况下使用；（2）中等（intermediate）模型，在需求确定以后使用；（3）高级或者详细（detailed）模型，在设计完成后使用。级别越高，模型中的参数约束越多。

COCOMO 81 这 3 个等级的模型也都满足类似上面通用公式的通式，即  $Effort = a \times (KLOC)^b \times F$ 。其中，Effort 为工作量，表示为人月。 $a$  和  $b$  为系数，具体的值取决于建模等级（即基本、中等或详细）以及项目的模式（即有机型、半嵌入型或嵌入型）。这个系数的取值先由专家意见来决定，然后用 COCOMO 81 数据库的 63 个项目数据来对专家给出的取值再进一步求精。KLOC 为软件项目开发中交付的有用代码行（line of codes，LOC），代表着软件规模。 $F$  是调整因子，基本 COCOMO 81 模型中， $F=1$ ，后两个模型中， $F$  为 15 个成本因子对应的工作量乘数的乘积。

COCOMO 81 模型将项目的模式分为有机的、嵌入式和半嵌入式的三种类型：

1) Organic（有机的、组织性的、独立的）：主要指各类应用程序，例如数据处理、科学计算。相对较小、较简单的软件项目，开发人员对开发目标理解比较充分，与软件系统相关的工作经验丰富，对软件的使用环境很熟悉，受硬件的约束比较小，程序的规模不是很大。

2) Embedded（嵌入式）：主要指各类系统程序，例如实时处理、控制程序等，要求在紧密联系的硬件、软件和操作的限制条件下运行，通常与某种复杂的硬件设备紧密结合在一起，对接口、数据结构、算法的要求高，软件规模任意，例如大而且复杂的事务处理系统、大型、超大型操作系统、航天用控制系统、大型指挥系统等。

3) Semidetached（半嵌入式）：主要指各类实用程序，例如编译器（程序）、联接器（程序）、分析器（程序）等，介于上述两种模式之间，规模和复杂度都属于中等或者更高。

下面重点研究 COCOMO 81 的三个等级的模型。

#### （1）基本 COCOMO 81 模型

基本 COCOMO 81 模型是静态、单变量模型，不考虑任何成本驱动，它用一个已估算出来的源代码行数（LOC）为自变量的函数来计算软件开发工作量，只适于粗略迅速估算。在开发的起始阶段，项目的相关信息很少，只要确定软件项目的模式与可能的规模，就可以用基本 COCOMO 81 模型作一个工作量初始估算。基本 COCOMO 模型是一个静态单变量模型，公式为  $Effort = a \times (KLOC)^b$ ，即通用公式  $Effort = a \times (KLOC)^b \times F$  中  $F=1$ ，Effort 是所需的人力（人月）， $a$ ， $b$  参数值见表 4-14。

方式	$a$	$b$
有机	2.4	1.05
半嵌入式	3.0	1.12
嵌入式	3.6	1.2

**【例3】** 要开发一个估计规模为 30KLOC 的银行系统应用程序项目，其功能以数据处理为主，属于有机型软件模式，根据专家意见和项目数据校准，系数  $a = 2.4$ ,  $b = 1.05$ ；调整因子  $F = 1$ ，则工作量 Effort 估算为  $\text{Effort} = 2.4 \times (30)^{1.05} = 85.3$  (人月)。

### (2) 中等 COCOMO 81 模型

随着项目的进展和需求的确定，可以使用中等 COCOMO 81 模型进行估算。中等 COCOMO 模型则在用 LOC 为自变量的函数计算软件开发工作量的基础上，再用涉及产品、硬件、人员、项目等方面属性的影响因素来调整工作量的估算，即用 15 个成本驱动改进基本模型，这是对产品、硬件、工作人员、项目的特性等因素的主观评估。

这里，通用公式  $\text{Effort} = a \times (\text{KLOC})^b \times F$ ，其中  $F$  为乘法因子，是根据成本驱动属性打分的结果，是对公式的校正系数。 $a$ 、 $b$  是系数，系数取值如表 4-15。

中等 COCOMO 81 模型定义了 15 个成本驱动因子，见表

表 4-15 中等 COCOMO 模型

方式	a	b
有机	3.2	1.05
半嵌入式	3.0	1.12
嵌入式	2.8	1.2

4-16，按照对应的项目描述，可将各个成本因子归为不同等级：很低 (Very low)、低 (Low)、正常 (Normal)、高 (High)、很高 (Very high)、极高 (Extra high)。例如，当软件失效造成的影响只是稍有不便时，要求的软件可靠性因子 (required software reliability, RELY) 等级为“很低”；当软件失效会造成很高的财务损失时，RELY 等级为“高”；当造成的影响危及人的生命时，RELY 等级为“很高”。不同等级的成本因子会对工作量 (也即开发成本) 产生不同的影响。例如，当一个项目的可靠性要求“很高”时，其可靠性因子 RELY 取值为 1.40，就是说，该项目相对于一个其他属性相同但可靠性要求为标称 (即取值为 1.00) 的项目来说，要多出 40% 的工作量。

表 4-16 中等 COCOMO 81 模型的成本驱动因子及等级列表

成本驱动因子	级别					
	很低	低	正常	高	很高	极高
<b>产品属性</b>						
可靠性 (RELY)	0.75	0.88	1	1.15	1.40	
数据规模 (DATA)		0.94	1	1.08	1.16	
复杂性 (CPLX)	0.70	0.85	1	1.15	1.30	1.65
<b>平台属性</b>						
产品运行时间 (TIME)			1	1.11	1.30	1.66
应用的存储使用 (STOR)			1	1.06	1.21	1.56
虚拟机空间 (VIRT)		0.87	1	1.15	1.30	
平台的稳定性 (TURN)		0.87	1	1.07	1.15	
<b>人员属性</b>						
分析能力 (ACAP)	1.46	1.19	1	0.86	0.71	
应用经验 (AEXP)	1.29	1.13	1	0.91	0.82	
程序员水平 (PCAP)	1.42	1.17	1	0.86	0.70	
平台经验 (VEXP)	1.21	1.10	1	0.90		
语言经验 (LEXP)	1.14	1.07	1	0.95		
<b>过程属性</b>						
实效编程经验的使用情况 (MODP)	1.24	1.10	1	0.91	0.82	
使用软件工具的水平 (TOOL)	1.24	1.10	1	0.91	0.83	
预计的进度压缩程度 (SCED)	1.23	1.08	1	1.04	1.10	

每个成本驱动因子按照不同等级对项目成本的影响程度，得到不同的工作量乘数，即调整因子  $F$ 。按照中等 COCOMO 81 模型的工作量公式，确定软件项目的模式和因子取值，即可估算工作量。

**【例4】** 对于例3的系统，若随着项目进展，可以确定其15个成本因子的情况，除了RELY、TURN、SCED因子的取值见表4-17外，其余因子取值均为1.00，系数 $a=3.2$ ， $b=1.05$ ，则其工作量估算为：

$$\begin{aligned}\text{Effort} &= 3.2 \times (30)^{1.05} \times (1.15 \times 0.87 \times 1.08) \\ &= 113.8 \times (1.15 \times 0.87 \times 1.08) = 123.0 \text{人月。}\end{aligned}$$

### (3) 高级(详细)的COCOMO 81模型

一旦软件的各个模块都已确定，估算者就可以使用高级(详细)COCOMO 81模型。高级(详细)COCOMO 81模型包括中间COCOMO模型的所有特性，但用上述各种影响因素调整工作量估算时，还要考虑对软件工程过程中分析、设计等各步骤的影响，将项目分解为一系列的子系统或者子模型，这样可以在一组子模型的基础上更加精确地调整一个模型的属性，当成本和进度的估算过程转到开发的详细阶段时，就可以使用这一机制。

详细COCOMO 81模型的工作量计算公式与中等COCOMO 81模型相同，其主要区别在于：1) 将待估算的软件项目分解为模块、子系统、系统3个等级。2) 增加了与开发阶段相关的工作量乘数，它可以准确反映成本驱动因子对工作量阶段分布的影响。例如，“低”级别的应用经验因子(application experience, AEXP)会增加早期开发的额外工作量，但到后期待团队熟悉应用后，就不会再耗费那么多工作量去做了解背景知识之类的工作了。对每个成本驱动属性，详细COCOMO 81模型都有一组表格，每个因子级别都有一个独立的工作量乘数来说明它对各主要开发阶段的影响(以应用经验因子AEXP为例，见表4-18。表格中对应的4个阶段分别为：需求与产品设计RPD(Requirements & Product Design)、详细设计DD(Detailed Design)、代码与单元测试CUT(Code & Unit Test)和集成与测试IT(Integration & Test)。

总而言之，详细COCOMO 81模型通过更细粒度的因子影响分析、考虑阶段的区别，使我们能更加细致地理解和掌控项目，有助于更好地控制预算和项目管理。

## 2. COCOMO II

COCOMO 81以及后来的专用Ada COCOMO，虽然较好地适应了它们所建模的一类软件项目，但是随着软件工程技术的发展，新模型和新方法不断涌现，不但没有好的软件成本和进度估算模型相匹配，甚至因为产品模型、过程模型、属性模型和商业模型等之间发生的模型冲突(model clash)等问题，不断导致项目的超支与失败，COCOMO 81也显得越来越不够灵活和准确。针对这些问题，Boehm教授与他的同事们在改进和发展COCOMO 81的基础上，于1995年提出了COCOMO II。

COCOMO II给出了三个层次的软件开发工作量估算模型，这三个层次的模型在估算工作量时，对软件细节考虑的详尽程度逐级增加。

1) 应用系统组成(application composition)模型。这个模型主要用于估算构建原型的工作量，模型名字暗示在构建原型时大量使用已有的构件。它是基于对象点(object point)对采用集成计算机辅助软件工程工具快速应用开发的软件项目工作量和进度进行估算，用于项目规划阶段。

2) 早期设计(early design)模型。这个模型适用于体系结构设计阶段。基于功能点(function point, FP)或可用代码行以及5个规模指数因子、7个工作量乘数因子，选择软件体系结构和操作，用于信息还不足以支持详细的细粒度估算阶段。

3) 后体系结构(post-architecture)模型。这个模型适用于完成体系结构设计之后的软件开发阶段。后体系结构模型，顾名思义，发生在软件体系结构完好定义和建立之后，基于源代码行和/或功能点以及17个工作量乘数因子，用于完成顶层设计和获取详细项目信息阶段。

下面以后体系结构模型为例，介绍COCOMO II模型。该模型把软件开发工作量表示成代码行数

表4-17 成本驱动因子的取值

成本驱动因子	级别	取值
RELY	高	1.15
TURN	低	0.87
SCED	低	1.08

表4-18 高级(详细)COCOMO 81模型工作量乘数的阶段差异性示例

成本驱动因子	开发阶段	级别					
		很低	低	正常	高	很高	极高
AEXP (应用经验因子)	需求设计阶段	1.40	1.20	1	0.87	0.75	
	详细设计阶段	1.30	1.15	1	0.9	0.80	
	编码和单元测试阶段	1.25	1.10	1	0.92	0.85	
	集成测试阶段	1.25	1.10	1	0.92	0.85	

(KLOC) 的非线性函数:

$$E = a \times \text{KLOC}^b \times \prod_{i=1}^{17} f_i$$

其中,  $E$  是开发工作量 (以人月为单位);  $a$  是模型系数; KLOC 是估计的源代码行数 (以千行为单位);  $b$  是模型指数;  $f_i (i=1 \sim 17)$  是成本因素。

Boehm 把成本因素划分成产品因素、平台因素、人员因素和项目因素等 4 类, 共 17 个属性。表 4-19 列出了 COCOMO II 模型使用的成本因素及与之相联系的工作量系数。

表 4-19 中等 COCOMO II 模型的成本驱动因子及等级列表

成本驱动因子	级别					
	很低	低	正常	高	很高	极高
产品属性						
可靠性	0.82	0.92	1.00	1.10	1.26	
数据规模		0.90	1.00	1.14	1.28	
复杂性	0.73	0.87	1.00	1.17	1.34	1.74
文档 (DOCU)	0.81	0.91	1.00	1.11	1.23	
复用 (RUSE)		0.95	1.00	1.07	1.15	1.24
平台属性						
执行时间的约束			1.00	1.11	1.29	1.63
存储约束			1.00	1.05	1.17	1.46
平台易变性 (PVOL)		0.87	1.00	1.15	1.30	
人员属性						
分析能力	1.42	1.19	1.00	0.85	0.71	
应用经验	1.22	1.10	1.00	0.88	0.81	
程序员水平	1.34	1.15	1.00	0.88	0.76	
平台经验 (PLEX)	1.19	1.09	1.00	0.91	0.85	
语言与工具经验 (LTEX)	1.20	1.09	1.00	0.91	0.84	
人员连续性 (PCON)	1.29	1.12	1.00	0.90	0.81	
项目属性						
工作地分布程度 (SITE)	1.22	1.09	1.00	0.93	0.86	0.80
使用软件工具的水平 (TOOL)	1.17	1.09	1.00	0.90	0.78	
进度约束 (SCED)	1.43	1.14	1.00	1.00	1.00	

与原始的 COCOMO 模型相比, COCOMO II 模型使用的成本因素的变化有:

1) 新增加了 4 个成本因素, 它们分别是要求的可重用性、需要的文档量、人员连续性 (即人员稳定性) 和多地点开发。这个变化表明, 这些因素对开发成本的影响日益增加。

2) 略去了原始模型中的 2 个成本因素 (计算机切换时间和使用现代程序设计实践)。

3) 某些成本因素 (分析员能力、平台经验、语言和工具经验) 对生产率的影响 (即工作量系数最大值与最小值的比率) 增加了, 另一些成本因素 (程序员能力) 的影响减小了。

工作量方程中模型指数  $b$  的值的确定方法:

原始的 COCOMO 模型把软件开发项目划分成有机式、半嵌入式和嵌入式这样 3 种类型, 并指定每种项目类型所对应的  $b$  值 (分别是 1.05, 1.12 和 1.20)。

COCOMO II 采用了更加精细得多的  $b$  分级模型, 这个模型使用 5 个分级因素  $W_i (1 \leq i \leq 5)$ , 其中每个因素都划分成从甚低 ( $W_i = 5$ ) 到特高 ( $W_i = 0$ ) 的 6 个级别, 然后用下式计算  $b$  的数值:

$$b = 1.01 + 0.01 \times \sum_{i=1}^5 W_i$$

因此,  $b$  的取值范围为 1.01 ~ 1.26。显然, 这种分级模式比原始 COCOMO 模型的分级模式更精细、更灵活。

COCOMO II 使用的 5 个分级因素如下所述:

1) 项目先例性。这个分级因素指出, 对于开发组织来说该项目的新奇程度。诸如开发类似系统的经验, 需要创新体系结构和算法, 以及需要并行开发硬件和软件等因素的影响, 都体现在这个分级因素中。

2) 开发灵活性。这个分级因素反映出,为了实现预先确定的外部接口需求及为了及早开发出产品而需要增加的工作量。

3) 风险排除度。这个分级因素反映了重大风险已被消除的比例。在多数情况下,这个比例和指定了重要模块接口(即选定了体系结构)的比例密切相关。

4) 项目组凝聚力。这个分级因素表明了开发人员相互协作时可能存在的困难。这个因素反映了开发人员在目标和文化背景等方面相一致的程度,以及开发人员组成一个小组工作的经验。

5) 过程成熟度。这个分级因素反映了按照能力成熟度模型(见13.7节)度量出的项目组织的过程成熟度。

### 3. COCOMO 模型扩展及其系列

随着使用范围的扩大及估算需求的增加,Boehm教授及其在美国Southeast California大学软件工程中心(Center for Software Engineering, CSE)的同事、研究生除了进行上述改进之外,还增加了不少扩展模型以解决其他问题,形成了COCOMO模型系列,包括:用于支持增量开发中成本估算的COINCOMO(constructive incremental COCOMO)、基于数据库实现并支持灵活数据分析的DBA COCOMO(database access) doing business as COCOMO II)、用于估算软件产品的遗留缺陷并体现质量方面投资回报的COQUALMO(constructive quality model)、用于估算并跟踪软件依赖性方面投资回报的iDAVE(information dependability attributed value estimation)、支持对软件产品线的成本估算及投资回报分析的COPLIMO(constructive product line investment model)、提供在增量快速开发中的工作量按阶段分布的COPSEMO(constructive phased schedule and effort model)、针对快速应用开发的CORADMO(constructive rapid application development model)、通过预测新技术中最成本有效(most cost effective)的资源分配来提高生产率的COPROMO(constructive productivity improvement model)、针对集成COTS软件产品所花费工作量估算的COOTS(constructive commercial-off-the-shelf cost model)、估算整个系统生命周期中系统工程所花费工作量的COSYSMO(constructive systems engineering cost model)、用于估算主要系统集成人员在定义和集成软件密集型SoS(system-of-system)组件中所花费工作量的COSOSIMO(constructive system of systems integration cost model)等。

因为COCOMO模型应用的日益广泛,其他研究者也纷纷提出有针对性的改进或者扩展方案,不断丰富和完善基于算法模型的估算方法。

#### 4.3.5 专家估算法

专家估算由一个被认为是该任务专家的人来进行估算,并且估算过程的很大一部分是基于不清晰、不可重复的推理过程,也就是直觉(intuition)。对于某一个专家自己所用的估算方法而言,经常使用工作分解结构(WBS),通过将项目元素放置到一定的等级划分中来简化预算估计与控制的相关工作当仅有的可用信息只能依赖专家意见而非确切的经验数据时,专家方法无疑是解决成本估算问题的最直接的选择。并且,专家可以根据自己的经验对实际项目与经验项目的差异作更细致的发掘,甚至可以洞察未来新技术可能带来的影响。但是,其缺点也很明显,就是专家的个人偏好、经验差异与专业局限性都可能为估算的准确性带来风险。

由于专家作为个体,存在很多可能的个人偏好,因此通常人们会更信赖多个专家一起得出的结果,并为达成小组一致。为此,引入了Delphi方法:首先,每个专家在不与其他人讨论的前提下,先对某个问题给出自己的初步匿名评定。第1轮评定的结果收集、整理之后,返回给每个专家进行第2轮评定。这次专家们仍面对同一评定对象,所不同的是他们会知道第1轮总的匿名评定情况。第2轮的结果通常可以把评定结论缩小到一个小范围,得到一个合理的中间范围取值。与专家的这种沟通可以多次进行。Delphi方法的基本步骤是:

- 1) 组织者发给每位专家一份软件系统的规格说明和一张记录估算值的表格,请专家估算。
- 2) 专家详细研究软件规格说明后,对该软件提出3个规模的估算值。
  - 最小值  $a_i$ 。

- 最可能值  $m_i$ 。
- 最大值  $b_i$ 。

3) 组织者对专家表格中的答复进行整理, 计算每位专家的平均值  $E_i = (a_i + 4m_i + b_i)/6$ , 然后计算出期望值:  $E = (E_1 + E_2 + \dots + E_n)/n$ 。

4) 综合结果后, 再组织专家无记名填表格, 比较估算偏差, 并查找原因。

5) 上述过程重复多次, 最终可以获得一个多数专家共识的软件规模。

#### 4.3.6 猜测法

猜测法是一种经验估算法, 进行估算的人有专门的知识和丰富的经验, 据此提出一个近似的数据, 是一种原始的估算方法。此方法只适用于要求很快拿出项目的大概数字的情况, 对于要求详细估算的项目是不适合的。

#### 4.3.7 估算方法综述

实际上, 进行软件规模成本估算时, 会根据不同的时期、不同的状况采用不同的方法。在项目初期, 尤其合同阶段, 项目的需求不是很明确, 而且需要尽快得出估算的结果, 可以采用类比法。在需求确定之后, 开始规划项目的时候, 可以采用自下而上估算法或者参数估算法。随着项目的进展, 项目经理需要时时监控项目的状况, 尤其在项目的不同阶段, 也需要重新评估项目的成本, 一般说, 项目经理根据项目经验的不断积累, 会综合一个实用的评估方法。自下而上估算法费时费力, 参数估算法比较简单, 它们的估计精度相似。但是各种方法不是孤立的, 应该注意相互的结合使用, 类比估算法通常用来验证参数估算法和自下而上估算法的结果。

现在, 介绍一下目前软件项目中常用的软件成本估算方式, 它是一种自下而上和参数法的结合模型, 步骤如下:

##### (1) 任务分解 (WBS)

对项目任务进行分解, 并对分解的任务进行编号 (WBS 编号), 例如分解之后共有  $n$  个任务, 1, 2,  $\dots$ ,  $i$ ,  $\dots$ ,  $n$ 。

##### (2) 规模估算

估算分解后的每个任务的规模 (单位一般是人月), 可以采用如下方法之一:

- 估算任务  $i$  工作量的最大值 Max、最小值 Min、最可能值 Avg, 则任务  $i$  的规模估算  $Q_i = (Max + 4Avg + Min)/6$ ;
- (或者) 估算任务  $i$  工作量的最可能值 Avg, 则任务  $i$  规模估算  $Q_i = Avg$ 。

##### (3) 估算直接成本

估算每个任务  $i$  成本  $E_i$ :

- $E_i = Q_i \times$  人力成本参数 (例如: 一个软件项目的规模是 3 人月, 企业的人力成本参数为 2 万/人月, 则这个项目的直接成本是 6 万元。); 如果任务  $i$  的价值是固定的 (例如采用固定价格外包), 就不用先计算工作量  $Q_i$ , 而是直接给出成本  $E_i$ 。

直接成本 =  $E_1 + E_2 + \dots + E_i + \dots + E_n$ , 直接成本是与本项目直接相关的成本。直接成本包括开发成本、管理成本、质量成本等。

如果任务分解结果 (WBS) 中包括了质量任务和管理任务, 例如 WBS 分解中包含需求评审、配置管理、项目规划、项目评审、项目跟踪等工作量, 在估算成本时, 就会比较的容易。当然, 有的 WBS 任务分解中不包括质量、管理等任务, 只对开发任务进行分解。这时, 可以采用简易估算方法估算管理、质量工作量 (单位: 人月), 例如, 如果 Scale(Dev) 是开发工作量规模, 则管理质量工作量规模  $Scale(Mgn) = a \times Scale(Dev)$ , 其中  $a$  是比例系数, 可以根据企业的具体情况而定, 例如:  $a$  可以是 20% ~ 25% 之间的参数。

##### (4) 估算间接成本

估算成本的时候除了估算直接成本, 还要估算间接成本, 间接成本是指直接成本之外的成本, 例

如企业的日常开销（例如房租水电等）、行政管理费用、员工福利、培训、预防性维护费用、市场宣传及其他等。总估算成本 = 直接成本 + 间接成本，间接成本可以根据企业的具体成本模型计算，如果企业没有成熟的成本模型，可以采用简易的算法计算，例如间接成本 = 直接成本 × 间接成本系数，也就是说可以通过直接成本来计算间接成本，这里就需要确定一个系数，即间接成本系数，这个系数也是根据企业的具体情况而定的，例如：间接成本系数 = 1.5 ~ 3。

#### (5) 项目总估算成本

可以采用简易的算法计算间接成本，间接成本 = 直接成本 × 间接成本系数，而直接成本 = 规模 × 人力成本参数，所以，间接成本 = 规模 × 人力成本参数 × 间接成本系数，而总估算成本 = 直接成本 + 间接成本，那么，总估算成本 = 规模 × 人力成本参数 (1 + 间接成本系数)。所以，可以得到成本估算的简易算法：估算成本 = 规模 × 成本系数，这里，成本系数 = 人力成本参数 (1 + 间接成本系数)，所以可以通过成本系数直接估算项目的总成本，如果成本系数 = 3 万/人月，而项目的规模为 3 人月，则项目的总成本为 9 万元。

如果项目合同额中还包含一些固定价格采购的话，例如硬件设备，系统软件等，就可以单独计算，这个数值一般是固定的，可以在计算总成本的时候直接加入其中。

#### (6) 项目报价

项目的成本不是项目的报价，项目的成本是项目报价的参考值，因为企业要通过项目来赢利的，如果项目报价等于项目成本，那么企业不会赢利的，所以在合同报价的时候，除了评估项目的成本，还有考虑项目的利润，当然这个利润是与风险同在的，所以，项目总报价 = 项目总估算成本 + 风险利润，其中的风险利润包括项目的风险基金、项目税费以及税后的利润等。

同样，项目利润是根据企业的不同而不同的，但是，可以通过项目总估算成本的调整系数而计算，例如项目风险利润 = 项目总估算成本 ×  $a\%$ ， $a$  是利润系数，根据企业、项目的不同而不同，有的企业利润空间大，系数可能大些；有的项目可能风险比较大，系数可能也应该大些等。一般说风险在项目中占有一定的比例，尤其软件行业，很多的项目都带有很多的不确定性，风险就比较大的。所以，要预留一定的风险保证金。

项目总价相当于合同的报价，而估算成本是项目的成本，也就是合同报价的底线，如果超过这个底线，企业就会亏本的。这个估算成本是将来进行项目跟踪控制的一个参照值。而这个估算成本在项目过程中如何花费，就是项目的预算了。

### 4.4 估算的误差度

一般说，进行规模成本估算都存在一定的误差。误差的大小会因项目不同、因项目阶段的不同而有所不同。

#### 1. 估算的误差

项目估算在不同的阶段，会出现不同的误差，表 4-20 是常规估算准确度的统计表，从表可以看出，越是项目前期，估算的准确性越低。随着对项目理解的深入，以及经验的丰富，估算的准确性也在提高。

其中，

- 1) 初步量级估算：这种估算还是比较“粗略”的，一般是在项目的初期中采用，属于自顶向下估算。它的估算误差一般是 -25% ~ 75% 之间。
- 2) 预算级估算：这种估算也有些粗略，在项目的规划过程中采用，也是属于自顶向下估算。它的估算误差一般是 -10% ~ 5% 之间。
- 3) 定义级估算：这种估算最精确的方法之一，在项目规划过程后使用此方法，它与自底向上估算方法有关。估算误差一般是 -5% ~ 10% 之间。

表 4-20 成本估算的误差

类型	准确度	说明
量级估算：合同前	-25% ~ +75%	概念和启动阶段，决策
预算级估算：合同期	-10% ~ +25%	编制初步计划
确定级估算	-5% ~ +10%	任务分解后的详细计划

## 2. 估算不准的主要原因

估算不准的主要原因如下：

- 1) 基础数据不足，准确估算要求估算企业应该有足够的以往项目的经验数据作为估算的基础，但是，很多企业的基础数据远远不足；
- 2) 估算对需求是非常敏感的，必须对项目的需求理解到很深的程度，才能准确估算，而实际项目中，估算前对需求的理解很少达到很深的程度；
- 3) 软件项目存在很多不确定性因素，项目进展过程中会存在很多的变更；
- 4) 缺乏经验的估算人员，例如：过分乐观，考虑不周，方法不一致；
- 5) 签约前后不连贯和低劣的推测技术也是不准的原因。签约前，销售人员为了拿到项目，而夸大了承诺，或者削减了价格，等签约后，项目经理接手项目的时候，可能会处于无法成功的尴尬境地，所以，估算也没法准确的。

因为市场竞争非常激烈，销售人员常作出一些不切实际的承诺，而这些承诺需要实施人员去兑现。因为这些承诺无法实现导致客户不认可，有的项目甚至成了反面典型。

## 3. 对付估算误差的方法

既然估算有误差是在所难免的，那就要采取一定的方法对付有误差的估算。一般说，首先要避免低劣的估算，如果有了低劣估算，就要采取方式处理低劣估算带来的结果。

为了避免低劣的估算，应尽可能使企业的估算机构专业化，进行相关的培训，同时制定一套严格的方法和步骤，以便估算者可以参照执行。另外，应该尽量避免估算中的过分乐观，政治压力，低劣估算模型。下面列出一些避免低劣估算的技巧，例如：

- 1) 避免无准备的估算
- 2) 留出估算的时间，并做好计划
- 3) 使用以前的项目数据
- 4) 使用以开发人员为基础的估算
- 5) 分类法估算（使用多种估算方法分别估算，然后比较结果）
- 6) 详细的较低层次上的估算
- 7) 不要忽略普通任务
- 8) 使用估算工具
- 9) 使用几种不同估算技术，并比较它们的结果

为了更清晰的表达估算的误差，可以在表达估算结果时，采用一定的形式，例如表 4-21 表达的估算结果是 6 人月，但是可能在 4 人月和 9 人月之间。或者如表 4-22 表示了估算结果和最佳以及最差情况。

表 4-21 风险化的估算表

估算：6 个人月，+3 人月，-2 人月	
+1 人月：延迟交付转换子系统	-1 人月：新成员的工作效率高
+1 人月：采用的新工具没有预计的好	-1 人月：采用的新工具比预计的好
+0.5 人月：员工病事假	
+0.5 人月：低估规模	

表 4-22 分情况估算的结果

情况	估算
最佳情况	100 人月
计划情况	110 人月
最差情况	150 人月

如果出现了低劣的估算结果，只好采取相应的应对措施。例如项目经理可以通过有力的数据和图示说明资源不足，以便争取更多资源和资金，来保证项目目标的实现；或者争取削减项目的一些要求。其次，确定目标的优先次序，当项目经理发现资源严重不足的时候，他们必须面对严酷的现实，可能不能完成所有的任务，这时，从长远看，他们必须做出某种让步，或者减少工作量，或者增加人员，或者追加资金等等，必须确定一定的优先次序。最后，应该强化变更管理程序，只有强化变更程序才可以避免多次变更造成的影响。

Glass 在其《软件工程中的真相与假象》一书中，总结了 55 个真相和 10 个假象，其中直接与估算有关的就有 7 个真相、1 个假象：

真相 8：造成软件项目失控最普遍的两个原因之一就是软件估算不足。

真相 9：往往在错误的时间，甚至在没有定义需求也即没有理解问题之前，就进行软件估算。

真相 10：往往由错误的人员，即不是由软件开发人员或者项目经理，而是由高层管理或者市场营销，进行软件估算。

真相 11：由错误的人员、在错误的时间做出的软件估算不但往往是错误的，也很少随着项目的进行做必要的调整。

真相 12：虽然软件估算会发生如此错误，但是人们不去关心产生错误的原因，却还在努力按照错误估算的时间进度执行。

真相 13：管理人员和技术人员对软件估算的认识是完全隔绝的，因此也难以达到一致的目标。

真相 14：很少有真正的可行性研究。

假象 6：要估算软件项目开发时间和开发成本，必须首先估算软件项目的代码行数。

无论从这些作者认为应该属于常识的“真相”，还是可能造成问题的“假象”来看，都可以轻而易举地得出两个结论：软件估算的价值和意义正越来越被大家所认识到；但是同时，软件估算的一些基础性问题一直也没有得到彻底解决，有的甚至还进一步恶化。尤其是，现有数百种软件估算方法、工具和模型中的多数仅仅能够覆盖软件生命周期中的设计、编码和测试阶段，而且都存在着重大缺陷。不过，这也并不意味着人们就束手无策和无所作为。只要坚持按照这些估算方法去做，还是可以获得比以往更好的结果的。

## 4.5 成本预算

成本预算是将项目的总成本按照项目的进度分摊到各个工作单元中去，即将总的成本安排到各个任务中，这些任务项是基于 WBS 分解的结果，所以，WBS、WBS 字典、每个任务的成本估计、进度、资源日历等可以作为成本预算的输入，成本预算的目的是产生成本基线，它可以作为度量项目成本性能的基础。

项目任务编排好了执行的先后关系并分配了资源后，项目中每个任务的成本预算就可以确定了，成本预算是根据项目的各项任务以及分配的相应资源进行计算的。成本预算的作用是提供对实际成本的一种控制机制，为项目管理者控制项目提供一把有效的尺子。所有任务的预算应该在总成本的控制下。这个预算应该与估算是一致的。

分配项目成本预算主要包括三种情况：

1) 分配资源成本。这是最常用的一种方式，即根据每个任务的资源分配情况来计算这个任务的成本预算，而资源成本与资源的基本费率紧密相连，所以要设置资源费率，例如标准费率、加班费率、每次使用费率等。例如，项目中开发人员的费率是 200 元/小时。

2) 分配固定资源成本。当一个项目的资源需要固定数量的资金时，用户可以向任务分配固定资源成本。例如，项目中分配张三的人力资源为固定资源成本，固定资源成本是 2 万元，即张三在这个项目中的成本固定为 2 万，不用计算张三花费的具体工时。

3) 分配固定成本。有些任务是固定成本类型的任务，也就是说，某项任务的成本不变，不管任务的工期有多长，或不管任务使用了哪些资源。在这种情况下，用户向任务直接分配成本。例如：项目中的某些外包任务成本是固定的，即任务为成本固定。假设项目某模块外包的成本是 10 万，则这个任务的成本为 10 万。

编制成本预算过程中应该提供一个成本基线 (cost baseline)，成本基线是每个时间阶段内的成本，它是项目管理者度量和监控项目的依据。例如，项目 A 的成本如表 4-23，是采用自下而上的估算方法得到的成本估算，总成本是 16 万，它的

表 4-23 项目 A 的成本估算

WBS 项	成本	成本	总成本 (万)
项目 A			16
1. 功能 1		8	
1.1 子功能 1	5		
1.2 子功能 2	3		
2. 功能 2		6	
2.1 子功能 1	3		
2.2 子功能 2	2		
2.3 子功能 3	1		
3. 功能 3		2	

每周的成本预算如表 4-24。将表 4-24 的预算数据可以转化为图 4-5 的项目的成本基线。

表 4-24 项目 A 的成本预算

周	任务	费用(万)
1	规划	1
2	需求	3
3	设计	5
4	开发-1	8
5	开发-2	12
6	测试	14
7	验收	16

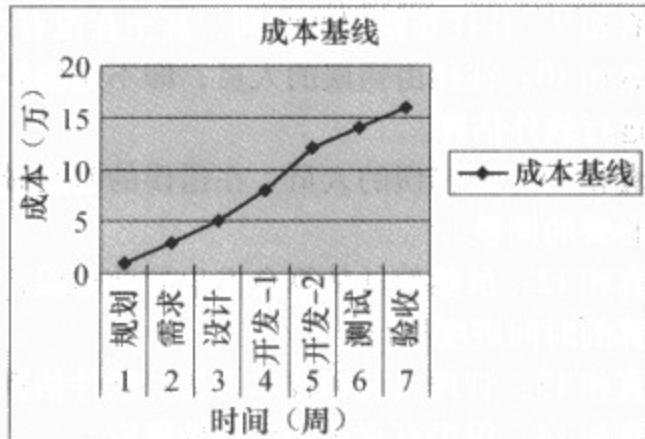


图 4-5 成本基线

成本预算与变更也相关，当发生变更的时候，需要同时变更成本预算。成本为资金需求提供信息。如果在项目进行过程中通过成本基线发现某个阶段的成本超出预算，需要研究必要时采取措施。

## 4.6 校务通系统案例分析

校务通系统估算是一循序渐进的过程，随着项目的不断发展，估算可以重复多次进行的，而且是逐步精确的。本项目采用自下而上和参数法综合的成本估算方法，然后结合进度形成项目成本预算基线。

### 4.6.1 成本估算

#### (1) 签订合同前

签订合同前，根据以往类似项目的经验，采用类比估算方法，进行粗略的估算：根据用户的要求采用 B/S 结构，公司 JSP + SQL Server 的技术比较成熟，以前成功完成过类似的项目，根据 SOW 的说明，基本上需要 2~3 个开发人员，2 个月左右的开发时间，基本上是 4~6 人月的规模，所以，10~15 万可以作为合同的参考价格。这个阶段只需要一个粗略的量级的成本估算，可以不进行详细的成本预算。

#### (2) 合同签署后

合同签署之后，根据现有的资源和 WBS 分解的结果，进一步细化估算，由于 WBS 分解是针对项目功能进行的分解，在成本估算的时候，首先估算每个任务的开发规模，然后通过系数获得相应的质量、管理任务的规模，从而计算直接成本，再计算间接成本，最后计算总成本，具体过程如下。

首先分析资源。

- 人力资源：
  - 2 个开发人员
  - 1 个项目管理人员
  - 1 个项目质量人员
  - 1 个配置管理人员
- 设备资源（作为间接成本计算）：
  - 3 台电脑
  - 1 台服务器

表 4-25 给出了项目规模的估算。

表 4-25 项目规模估算表 注：规模单位为人天

WBS	名称	估计值 (人天)	小计 (人天)	总计 (人天)
1	通用功能		31	103
1.1	电子课表	8		
1.2	会议通知和公告	3		
1.3	个人日记	5		
1.4	通讯录	2		
1.5	教师答疑	5		
1.6	作业布置和批改	8		
2	日常业务管理功能		26	
2.1	招生管理			26
2.1.1	报名	3		
2.1.2	招生	5		
2.1.3	分班	10		
2.1.4	统计查询	8		
2.2	学生日常管理			10
2.2.1	学生档案管理	4		
2.2.2	学生考勤管理	2		
2.2.3	学生奖惩	2		
2.2.4	学生变动	2		
2.3	教务管理		31	
2.3.1	教师日常管理	2		
2.3.2	年级、班级设置	4		
2.3.3	学科设置	2		
2.3.4	年班级课程设计	5		
2.3.5	排课表	9		
2.3.6	考试管理	4		
2.3.7	评价	5		
2.4	教师备课系统	(外包 5000 元)	1	
2.5	资源库系统	(外包 3000)	1	
2.6	网上考试	(外购 3000 元)	1	
2.7	聊天室	(已存在)	1	
2.8	论坛	(已存在)	1	

估算步骤如下：

1) 获取项目分解结果 WBS。

- 任务分解是根据项目的功能进行分解的。

2) 计算开发成本。

- 由于任务分解的结果主要是针对开发任务的分解，管理任务和质量任务可以通过计算开发任务得到，根据以往经验，管理任务和质量任务 = 20% × 开发任务。
- 从表 4-25 得知项目规模是 103 人天，开发人员成本参数 = 480 元/天，则内部的开发成本 = 480 元/天 × 103 天 = 49 440 元。
- 加上外包外购的部分软件成本  $5\ 000 + 3\ 000 + 3\ 000 = 11\ 000$  元，则开发成 =  $49\ 440 + 11\ 000 = 60\ 440$  元。

3) 计算管理、质量成本。

- 项目的管理和质量成本 = 开发成本 × 20% = 12 088 元。

4) 直接成本 =  $60\ 440 + 12\ 088 = 72\ 528$  元。

5) 计算间接成本。

- 间接成本包括前期合同费用、房租水电、培训、员工福利、客户服务等。
- 根据以往经验，采用公式：间接成本 = 25% 直接成本 = 18 132 元。

6) 计算总估算成本。

- 项目总估算成本 =  $72\ 528 + 18\ 132 = 90\ 660$  元。

7) 重新评估项目的报价。

- 重新评估一下项目的报价准确性，当然这时候，项目的合同已经签署了，报价是不能更改的，但是通过再次的评估可以进一步明确企业的项目运作和利润情况等。
- 如果项目的利润是 30%，其中风险基金 10%，利润 15%，税费 5%。则项目的总报价 =  $906\ 60 \times 1.3 = 117\ 858$  元，应该说报价还是比较合适的。

另外，可以采用简便的算法进行估算，企业的报价可以通过开发规模的估算直接得出，例如如果成本系数为 2.5 万元/人月，一个人月 22 人天，则项目报价 =  $2.5 \times 103/22 = 117\ 045$  元。

#### 4.6.2 成本预算

在编制校务通系统项目计划中考虑到，2 个开发人员是全职在这个项目中，而项目经理、质量保证和配置管理人员不是全职在这个项目中，他们同时还在管理其他的项目，进行成本估算的时候，应该根据项目人员付出的时间以及各项任务的具体情况进行成本预算，最后可以得到比较详细的成本分配情况，即成本基准。资源费用比例如表 4-26 所示。

表 4-26 资源费用比例

	资源名称	类型	缩写	最大单位	标准费率	加班费率	每次使用成本	成本累算	基准日历
1	姜岳尊	工时	姜	100%	¥70.00/工时	¥0.00/工时	¥0.00	按比例	标准
2	韩万江	工时	韩	100%	¥80.00/工时	¥0.00/工时	¥0.00	按比例	标准
3	孙 泉	工时	孙	100%	¥70.00/工时	¥0.00/工时	¥0.00	按比例	标准
4	郭天奇	工时	郭	100%	¥45.00/工时	¥0.00/工时	¥0.00	按比例	标准
5	岳 好	工时	岳	100%	¥40.00/工时	¥0.00/工时	¥0.00	按比例	标准

根据每个任务的资源分配和时间安排，确定项目的预算，如表 4-27，预算总成本为 75 160 元，与估算的成本基本持平，这样 75 160 元可以作为项目的成本控制参考。

表 4-27 项目成本预算

甘特图	任务名称		开始时间	完成时间	比较基准
	1	□校务通管理系统	2003 年 4 月 10 日	2003 年 6 月 6 日	¥75 160.00
	2	□软件规划	2003 年 4 月 10 日	2003 年 4 月 11 日	¥3 320.00
	3	项目规划	2003 年 4 月 10 日	2003 年 4 月 10 日	¥1 200.00
	4	计划评审	2003 年 4 月 11 日	2003 年 4 月 11 日	¥2 120.00

(续)

	任务名称	开始时间	完成时间	比较基准
5	需求开发	2003年4月14日	2003年4月18日	¥6 240.00
6	用户界面设计	2003年4月14日	2003年4月14日	¥1 120.00
7	用户需求评审	2003年4月15日	2003年4月15日	¥2 120.00
8	修改需求、修改用户界面	2003年4月16日	2003年4月16日	¥1 120.00
9	编写需求规格说明书	2003年4月17日	2003年4月17日	¥560.00
10	需求验证	2003年4月18日	2003年4月18日	¥1 320.00
11	设计	2003年4月17日	2003年4月22日	¥4 120.00
12	概要设计	2003年4月17日	2003年4月18日	¥1 120.00
13	数据库ER图编制、建库	2003年4月21日	2003年4月21日	¥560.00
14	设计评审	2003年4月22日	2003年4月22日	¥2 440.00
15	实施	2003年4月22日	2003年6月2日	¥54 640.00
16	通用功能 - 增量1	2003年4月22日	2003年4月30日	¥12 520.00
24	招生管理 - 增量2	2003年5月1日	2003年5月7日	¥9 000.00
30	学生日常管理 - 增量3	2003年5月8日	2003年5月12日	¥6 600.00
36	教务管理 - 增量4	2003年5月13日	2003年5月23日	¥16 040.00
45	教师辅助功能 - 增量5	2003年5月26日	2003年5月29日	¥5 800.00
50	聊天室/论坛 - 增量6	2003年5月30日	2003年6月2日	¥4 680.00
54	系统集成	2003年6月3日	2003年6月4日	¥2 920.00
55	系统集成测试	2003年6月3日	2003年6月3日	¥1 120.00
56	环境测试	2003年6月4日	2003年6月4日	¥1 180.00
57	提交	2003年6月5日	2003年6月6日	¥3 920.00
58	完成文档	2003年6月5日	2003年6月5日	¥1 480.00
59	验收、提交	2003年6月6日	2003年6月6日	¥2 440.00

图4-6是成本预算图示，即项目随时间的费用支出图，其中的BCWS（Budgeted Cost of Work Scheduled）就是预算曲线，也是将来项目跟踪控制的费用曲线。

## 4.7 小结

项目规模成本估算也是项目规划的基础，也是项目成本管理的核心，通过成本估算方法，分析并确定项目的估算成本，并以此为基础进行项目成本预算和计划编排，开展项目成本控制等管理活动。软件项目规模成本估算最常用的三种估算方法是类比估算法，参数模型估算法，自下而上估算法，实际应用中可以综合使用。不要盲目的认为存在一个最科学的最先进的数学模型进行项目的估算，如果企业没有足够多的历史数据，就不可能存在有效的数学模型。

成本预算表示项目成本如何分摊的，表示项目成本每个阶段的成本花费。

## 4.8 练习题

### 一、选择题

- ( ) 是用系统的功能数量来测量其规模，与实现产品所使用的语言和技术没有关系的。
  - 功能点
  - 对象点
  - 代码行
  - 用例点
- 如果你是某项目的项目经理，你已经估算出每个单元的成本是¥129。这个项目一共有1 200单元，你采用什么估算方法？( )
  - 自下而上估算法
  - 类比估算法
  - 专家估算法
  - 参数估算法

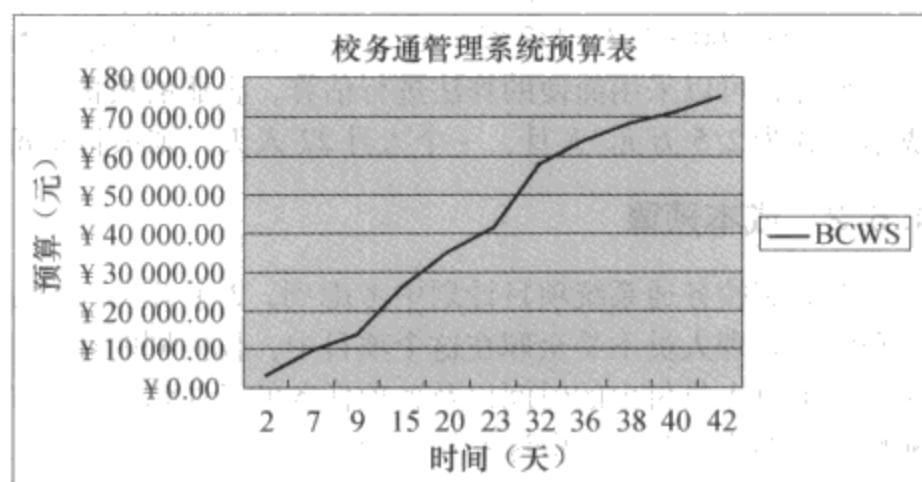


图4-6 成本预算图示

3. 在项目初期，进行竞标合同的时候，一般采用的成本估算方法是（ ）。  
A. 参数估算法      B. 类比估算法      C. 专家估算法      D. 功能点估算
4. 在成本管理过程中，项目经理确定的每个时间段，各个工作单元的成本是（ ）。  
A. 估算      B. 预算      C. 直接成本      D. 间接成本

## 二、判断题

1. 间接成本是与一个具体的项目相关的成本。（ ）
2. 软件项目的估算结果是比较准确的。（ ）
3. 在进行软件项目估算的时候，可以按照其他企业的估算模型进行估算。（ ）
4. COCOMO 81 有 3 个等级的模型：基本（basic）模型、中等（intermediate）模型、高级（详细（detailed））模型。（ ）

## 三、计算题

1. 如果某软件公司正在进行一个项目，预计有 50KLOC 的代码量，项目是中等规模的半嵌入式类型的项目，采用中等 COCOMO 81 模型，项目属性中只有可靠性为很高级别，其他属性为正常，计算项目是多少人月的规模？如果 2 万元/人月，则项目的费用是多少。
2. 项目经理正在进行一个媒体信息查询系统的项目的估算，他采用的 delphi 的成本估算方法，邀请 2 位专家估算，第一个专家给出 1 万、8 万、9 万的估算值，第二个专家给出了 4 万、6 万、8 万的估算，计算这个项目的成本估算值是多少？

# 第5章

## ■ 软件项目质量计划

虽然范围计划、进度计划、成本计划是项目管理中的核心计划，但这还不够，为了更好更全面地进行项目管理，还需要其他的计划，诸如质量计划、沟通计划、风险计划等等。本章进入本书路线图的质量计划，如图 5-1 所示。

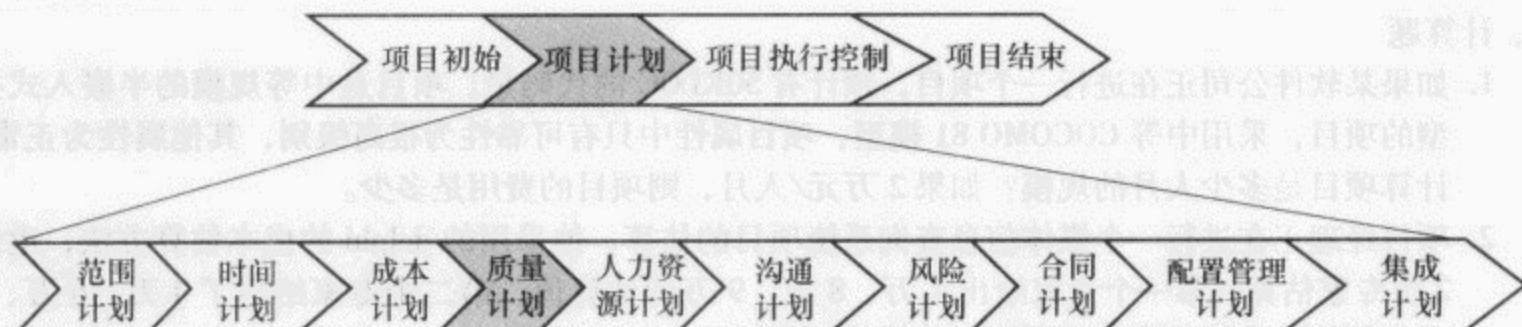


图 5-1 路线图——项目计划 - 质量计划

### 5.1 质量概述

1981 年，由于计算机程序改变而导致的 1/67 秒的时间偏差，使航天飞机上的 5 台计算机不能同步运行，这个错误导致了航天飞机发射失败。1986 年，1 台 Therac25 机器泄露致命剂量的辐射，致使两名医院病人死亡。造成惨剧的原因是一个软件出现了问题，导致这台机器忽略了数据校验。这些惨痛的教训说明，在软件项目开发中认真抓好质量管理，并加强有关软件项目质量管理的研究是摆在我们面前的重要课题。

国际 ISO 定义：“质量是产品或者服务满足明确和隐含需要能力的性能特性的总体”。还有几种比较具有代表性的质量定义，如：

- 质量是符合目的或者用途。
- 用户的感觉就是质量。
- 质量是符合顾客在其合理价格下对产品的要求。

所有的定义都明确或者隐含的与接受产品的顾客有关系，顾客接受产品或者服务，并提供资金，所以，应该以客户为中心。以往对质量的概念，仅局限于符合规定的要求，而忽视了顾客的需要。新项目管理的核心之一是项目管理必须以用户为中心，它强调了用户的满意度。所以质量是满足要求的程度，包括符合规定的要求和满足顾客的需求。归根到底，用户对于应用程序质量的评价是最重要的，当用户做出评价时，第一步要做的不过是用这个软件去完成他们希望软件完成的任务，用过之后，用户会提出该软件好与不好的意见。所以，质量好的一个重要方面是让用户满意，质量管理的目标是满足项目干系人的需求。

一个项目的主要内容是成本、进度、质量。良好的项目管理就是综合三方面的因素，平衡三方面的目标，最终依照目标完成任务。项目的这三个方面是相互制约和影响的，有时对这三方面的平衡策略甚至成为一个企业级的要求，决定了企业的行为，我们知道 IBM 的软件是以质量为最重要目标的，而微软的“足够好的软件”策略更是耳熟能详，这些质量目标其实立足于企业的战略目标。质量管理是项目管理的最高统一（三大目标的统一），如图 5-2 所示。

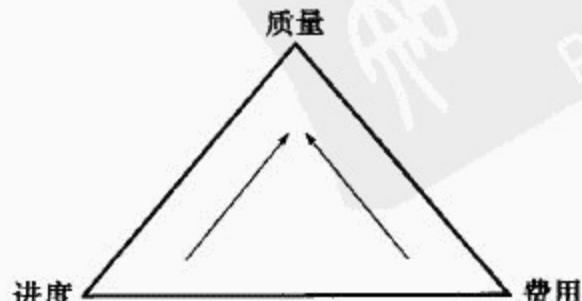


图 5-2 质量目标是最高统一

总之，质量是“一个实体的性能总和，它可以凭借自己的能力去满足对它的明示或暗示的需求”。在项目管理中，质量管理的既定方向就是通过项目范围界定管理体制，将暗示的需求变为明示的需求。

质量与等级（grade）是有区别的，等级是对具有相同功能的实体按照不同技术特征进行分类或者分级。无论产品采用任何等级标准，它都应该具备能满足相应功能要求的各种特征，这些特征的总和就是质量。所以，尽管产品可以有不同等级，但是无论等级高低，都可以实现自己等级内的高质量。所以，低等级不代表低质量。高等级也不代表高质量的。打印机有很便宜的低端产品也有很贵的高端产品。但是，低端的产品也要求是高质量的产品，高端的产品如果打印过程中出现故障，也说明有质量问题。打印机使用的打印纸也有不同的级别。你可以买光滑的照片打印纸，也可以买便宜的复印纸。都是纸，但是纸张的级别是不同的。

质量低通常是个问题，级别低就可能不是。例如，一个软件产品可能是高质量（没有明显问题，具备可读性较强的用户手册），低等级（数量有限的功能特点）；或者是低质量（问题多，用户文件组织混乱），高等级（无数的功能特点）。决定和传达质量与等级的要求层次是项目经理和项目管理小组的责任，其中，低质量就是质量问题了。

这里需要要明确几个概念：质量标准、质量策略和质量责任。

1) 质量标准是企业、国家或者国际制定的对某个方面的规范，与质量政策相比，更侧重质量的细节特征，属于微观的范畴。

2) 质量策略是某个组织针对自身要求制定的一种质量指导方针，更侧重于指导思想，属于宏观的范畴。

3) 质量责任是整个组织都对项目质量负有责任，但是如果明确和细化责任，就会形成人人有责，人人不负责的局面。所以，质量责任包括管理层的责任、最终责任、首要责任等。

### 5.1.1 软件质量

ANSI/IEEE Std 729 - 1983 对软件质量的定义为“与软件产品满足规定的和隐含的需求能力有关的特征或特性的全体”。软件质量反映了以下三方面的问题：

- 1) 软件需求是度量软件质量的基础，不满足需求的软件就不具备质量。
- 2) 不遵循各种标准中定义的开发规则，软件质量就得不到保证。
- 3) 只满足明确定义的需求，而没有满足应有的隐含需求，软件质量也得不到保证。

软件质量是贯穿于软件生存期的一个极为重要的问题，是软件开发过程中采用的各种开发技术和检验方法的最终体现。软件质量是软件满足软件需求规格中明确说明以及隐含的需求的程度。其中明确说明的是指在合同环境中，用户明确提出的需求或需要，通常是合同、标准、规范、图纸、技术文件中做出的明确规定，隐含的则应加以识别和确定，具体来说是顾客或者社会对实体的期望，或者是指人们所公认的，不言而喻的，不需要做出规定的需求。例如，数据库系统必须满足存储数据的基本功能。

根据软件质量国家标准 GB - T 8566—2001G，软件质量评估通常从对软件质量框架的分析开始。如图 5-3 所示，软件质量框架是一个“质量特征—质量子特征—度量因子”的三层结构模型。

在这个框架模型中，上层是面向管理的质量特征，每一个质量特征是用以描述和评价软件质量的一组属性，代表软件质量的一个方面。软件质量不仅从该软件外部表现出来的特征来确定，

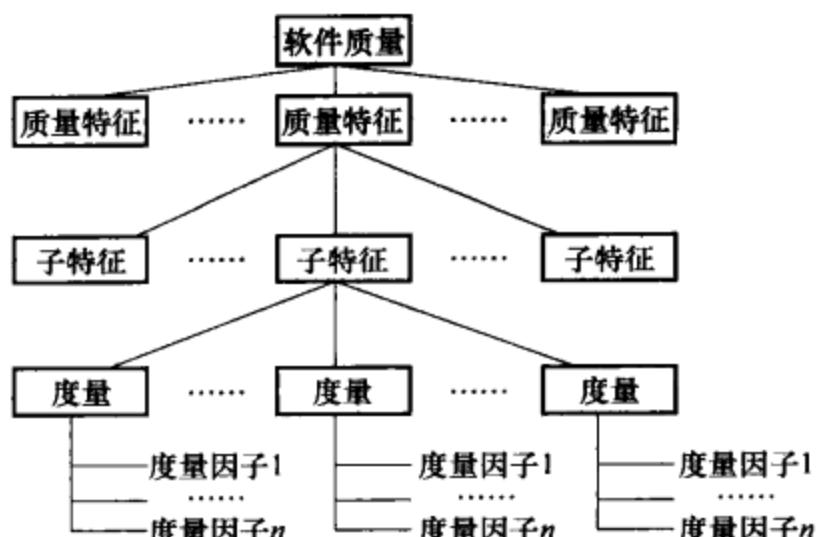


图 5-3 软件质量框架模型

而且必须从其内部所具有的特征来确定。软件的质量属性很多，如正确性、精确性、健壮性、可靠性、容错性、性能、易用性、安全性、可扩展性、可复用性、兼容性、可移植性、可测试性、可维护性、灵活性等。如果某些质量属性并不能产生显著的经济效益，我们可以忽略它们，把精力用在对经济效益贡献最大的质量要素上。简而言之，只有质量要素才值得开发人员下功夫去改善。何为质量要素呢？质量要素包括两方面的内容：1）从技术角度讲，对软件整体质量影响最大的那些质量属性才是质量要素。2）从商业角度讲，客户最关心的、能成为卖点的质量属性才是质量要素。

第二层的质量子特征是上层质量特征的细化，一个特定的子特征可以对应若干个质量特征。软件质量子特征是管理人员和技术人员关于软件质量问题的通信渠道。

最下面一层是软件质量度量因子（包括各种参数），用来度量质量特征。定量化的度量因子可以直接测量或统计得到，为最终得到软件质量子特征值提供依据。

### 5.1.2 软件质量与硬件质量的比较

软件质量与硬件质量有很大的不同，软件是逻辑实体，本质上无形、看不见、难测控；软件的质量问题基本是人为差错引起，又由千变万化的逻辑组合导致不同的质量问题，保证质量的方法是开发全过程采取措施，防错、检错，发生出错时，要进行维护修改，修改时考虑波及面，保证完整一致。而硬件是物理实体，有形、便于测控；硬件质量问题主要是设计、生产和使用的问题，失效由零部件或其结合故障引起，保证质量的方法是开发过程和生产过程加强控制，发生故障时，更换零部件，维修一般不会给其他地方造成影响。

### 5.1.3 软件质量的重要性

软件质量的重要性是不言而喻的，如今社会越来越多的系统是依赖于软件的，软件的不正确运行可能导致灾难性的后果，例如经济损失以至于人员伤亡。低质量的软件像定时炸弹一样，随时可能引起危害的。而且低质量的产品，要增加后期的成本。即使是小的缺陷也可以引起难以预料的后果。例如“千年虫”问题，虽然只是一条语句的问题，却带来很大的麻烦和损害。并为此付出很大的代价。产品的质量影响到开发速度、成本和其他的项目特质。在诸如规格或者设计等上游的错误，如果推到下游修正，比在上游找出错误并修正高出 50~200 倍的成本。另外，质量是企业的生命也是信誉，如果没有质量，企业就无法生存了。

人在做一件事情时，由于存在很多不确定的因素，一般不可能 100% 地达到目标。假设平常人做事能完成目标的 80%。如果某个人的目标是 100 分，那么他最终成绩可达 80 分。如果某个人的目标只是 60 分，那么他最终成绩只有 48 分。做一个项目通常需要多人的协作。项目的总质量（最高为 1）是所有个开发人员的工作质量之积。如果每个人的质量目标是 0.9 分，那么十个人开发团队的累积质量不会超过 0.03。只有每个人都做到 1，项目总质量才会是 1。

### 5.1.4 软件质量管理的发展过程

表 5-1 说明了软件质量管理的发展过程。从表中可以看出，软件质量管理随着开发管理方式的演变逐步向系统化发展。经历了从程序设计、程序系统再到软件工程的时期。

表 5-1 软件质量管理的发展过程

发展时期	年代	成品	开发组织方式	开发技术特点	开发工具	质量管理特征
程序设计	20世纪50年代末	程序	个体	个人设计、个人使用、手工技巧、无维护观念、无系统化方法	机器代码、汇编语言	靠开发者个人技巧
程序系统	20世纪70年代末	软件	项目设计组、软件作坊	程序设计理论深入、模块化、自顶向下，逐步求精，不重视维护问题	高级语言、操作系统、数据管理系统	靠少数人协调把关

(续)

发展时期	年代	成品	开发组织方式	开发技术特点	开发工具	质量管理特征
软件工程	至今	软件产品	软件机构 (软件工厂)	结构化设计理论和方法 面向对象方法、快速原型技术 .....	逐步产生与各种方法相应的工具、开发环境	系统化、规范化、全员化

## 5.2 质量管理

时间、成本与质量在项目管理中常常相提并论。那么如何在时间、成本、质量这三个方面找到均可以满意的模式，并恪守这种模式，持续地进行管理工作呢？这也就是质量管理的最终目标。

质量管理的学派和观点有很多，具有代表性有戴明理论、朱兰理论、克鲁斯比理论、田口玄一理论等。其中：

- 戴明理论的核心是“目标不变、持续改善和知识积累”，预防胜于检验。
- 朱兰理论的核心思想是适用性，适用性是通过遵守技术规范，使项目符合或者超过项目相关人及客户的期望。
- 克鲁斯比理论的核心思想是质量定义符合预先的要求，质量源于预防，质量的执行标准是零缺陷（zero defect），质量是用非一致成本来衡量的。
- 田口玄一核心思想是应用统计技术进行质量管理，通过损失函数来决定产生未满足目标产品的成本。

全面质量管理（TQM）是指通过全体员工的参与、改进流程、产品、服务和公司文化，达到在百分之百时间内生产百分之百的合格产品，以便满足顾客需求，TQM 是一种思想观念，一套方法、手段和技巧。

软件项目的质量管理指的是保证项目满足其目标要求所需要的过程，质量管理的关键是预防重于检查，事前计划好质量，而不是事后检查。

任何软件开发项目中，质量不仅仅拥有发言权，而且对项目的成败拥有表决权甚至最终的否决权。质量不仅会对软件开发项目本身的成绩产生影响，而且会对软件企业的形象、商誉、品牌的褒贬带来冲击和震荡。质量一般通过定义交付物标准来明确定义，这些标准包括各种特性及这些特性需要满足的要求。另外，质量还包含对项目过程的要求，比如规定执行过程应该遵循的流程、规范和标准，并要求提供过程被有效执行的证据。因此，质量管理主要就是监控项目的交付物和执行过程，以确保它们符合相关标准，同时确保不合格项能够按照正确方法排除。还可能对项目的顾客应对质量作出规定，包括应对顾客的态度、速度以及方法。高质量来自满足顾客需求的质量计划、质量保证、质量控制和质量改善活动，来自保证质量、捍卫质量和创造质量的卓越理念、规则、机制和方法。

目前的软件项目质量不太令人满意，而由于软件质量问题导致的损失也不计其数。作为项目管理者，掌握质量管理的技能是必需的。质量管理过程（project quality management）的目的是确保项目满足需要执行的过程。主要过程是质量规划（quality planning）、质量保证（quality assurance）和质量控制（quality control）。质量规划确定与项目相关的质量标准以及如何满足这些标准。质量保证是通过定期的评估项目的整体性能以确保项目满足相关的质量标准。质量控制通过控制特定项目的状态保证完全按照质量标准完成，同时确定质量改进的方法。

### 5.2.1 质量计划

软件质量计划过程是确定项目应达到的质量标准，以及决定如何满足质量标准的计划安排和方法。合适的质量标准是质量计划的关键。只有做出精准的质量计划，才能指导项目的实施，做好质量管理。

质量计划主要指依据公司的质量方针、产品描述以及质量标准和规则等制定出来实施策略，其内容全面反应用户的要求，为质量小组成员有效工作提供了指南，为项目小组成员以及项目相关人员了解在项目进行中如何实施质量保证和控制提供依据，为确保项目质量得到保障提供坚实的基础。

不求质量的开发人员往往凭据经验草草了事，追求完美的开发人员则在该项任务上耗费太多精力。这是由于没有验收标准而导致的情景。因为没有验收标准，你无法知道你要进行的任务需要一个什么样的结果，需要达到什么样的质量标准。作为质量管理最终责任承担者的项目经理来说，只有制定好每个任务的验收标准，才能够严格把好每一个质量关，同时了解项目的进度情况。

### 5.2.2 质量保证

质量保证（Quality Assurance, QA），是“为了提供信用，证明项目将会达到有关质量标准，而开展的有计划、有组织的工作活动”。它是贯穿整个项目生命周期的系统性活动，经常性地针对整个项目质量计划的执行情况，进行评估、检查与改进等工作，向管理者、顾客或其他方提供信任，确保项目质量与计划保持一致。

软件质量保证的目的是验证在软件开发过程中是否遵循了合适的过程和标准。其主要作用是保证软件透明开发的主要环节。它贯穿于整个项目的始终。

质量保证的职责就是确保过程的有效执行，监督项目按照指定过程进行项目活动；同时审计软件开发过程中的产品是否按照标准开发。质量保证的主要方法是质量审计，即产品审计和过程审计，详见 12.4.1 节。

质量保证过程是通过评价项目整体绩效，建立对质量要求的信任。质量保证是为管理人员及相关的各方提供软件项目的过程和项目本身的可视化。质量保证可以用一句英文“Is it done right?”表达，完成的是否正确？即在完成后看是否正确。这个任务本身并不直接提高产品的质量，但是通过质量保证的一系列工作可以间接的提高产品的质量。开发高质量产品是开发组的责任，质量保证人员的职责是规划和维护质量过程，以便实现项目的目标。为此，质量保证人员要定期对项目质量计划执行情况进行评估、审核与改进等工作，在项目出现偏差的时候提醒项目管理人员，它要提供项目和产品可视化的管理报告，质量保证人员通过各种手段来保证得到高质量结果的工作，属于管理职能。例如，产品质量审计就是质量保证的工作过程。

### 5.2.3 质量控制

质量控制（Quality Control, QC）是确定项目结果与质量标准是否相符，同时确定消除不符的原因和方法，控制产品的质量，及时纠正缺陷的过程。质量控制是对阶段性的成果进行检测、验证，为质量保证提供参考依据。质量保证是保证项目能够按照计划的过程执行，按标准的过程去做也不一定能得到一盘完美的“大餐”，因为火候等因素实在很难掌握。万一掌握不好怎么办？软件质量控制主要就是发现和消除软件产品的缺陷。对于高质量的软件来讲，最终产品应该尽可能达到零缺陷。而软件开发是一个以人为中心的活动，所以出现缺陷是不可避免的。因此，要想交付一个高质量的软件，消除缺陷的活动就变得很重要。消除缺陷是通过“评审”和“测试”这类质量控制活动来实现的。质量控制方法有技术评审、走查、测试、返工、控制图、趋势分析、抽样统计、缺陷追踪等，详见 12.4.2 节。

质量控制可以用一句英文“Is it right done?”表达，是否正确完成？即在完成前检查质量。通过质量控制本身可以提高产品的质量，质量控制一般由开发人员实施。是直接对项目工作结果的质量进行把关，属于检查职能。例如，航天飞机发射进入倒计时的时候，可以进行质量保证和质量控制，质量控制是检测各部分是否运行正常，而这时进行质量保证只是对这次的发射过程提交质量保证报告，这个质量保证报告对这次的航天飞机发射已经没有太大的意义了，但是对将来的航天飞机发射是有意义的。

质量保证和质量控制是有区别的，质量控制是检验产品的质量，保证产品符合客户的需求，是产品质量检查者，即挑毛病的。质量保证是审计产品和过程的质量，保证过程被正确执行，是过程质量审计者，审计是来确认项目按照要求进行的证据。QC人员进行质量控制，向管理层反馈质量信息；QA人员则确保QC按照过程进行质量控制活动，按照过程将检查结果向管理层汇报。

如果企业原来具有QC人员并且QA人员配备不足，可以先确定由QC兼任QA工作。但是只能是暂时的，独立的QA人员应当具备，因为QC工作也是要遵循过程要求的，也是要被审计过程的，这种混合情况，难以保证QC工作的过程质量。

### 5.3 质量计划开发

现代质量管理强调：质量是计划出来的，而不是检查出来的。只有制定出切实可行的质量计划，严格按照规范流程实施，才能达到规定的质量标准。尤其软件项目更是预防胜于检验，要求预防、计划、未雨绸缪，而不是后期的补救和打补丁。因为质量是在开发过程中形成的，高质量的开发才能产生高质量的软件产品。当软件完成之后，你就无法再提高它的质量了，好的质量保证开始于好的设计，而且在遵守设计好的编程过程中得以延续。程序员必须在编程过程中重视每一行编码的质量，在测试、运行或者维护中所发现的每个缺陷都是不重视质量的开发人员带来的。一旦一个庞大的软件被开发出来后，保证它没有缺陷是不现实的，测试也是不能保证的，那么保证软件没有错误或者几乎没有错误的最好办法就是做一些事情将错误扼杀在摇篮里，如何做好这些事情就需要做好的质量管理和质量规划过程。

质量形成于产品或者服务的开发过程中，而不是事后的检查（测试）把关等。在开发一个最终的软件产品过程，软件项目可能会产生很多的中间产品，包括需求规格、设计说明书、源程序、测试计划、测试结果等，他们对最终产品的结果起着很重要的作用，所以应该对他们以及最终的产品进行评估和控制，以保证产品最终满足用户需求。

#### 5.3.1 质量成本

与任何管理活动一样，质量管理也是需要成本的，也就是说要采取行动就要有所花费，当然，如果不采取任何行动，则不需要任何花费。质量成本（Cost Of Quality, COQ）是由于产品的第一次工作不正常而衍生的附加花费。在质量概念中还有一个要素是值得注意的，那就是质量的经济性。根据Crosby的质量定义，“符合需求”的代价是指第一次把事情做对所花费的成本，总是最经济的。而“不符合要求”的代价是必须进行补救使企业产生额外的支出，包括时间、金钱和精力，由此产生了质量损失，成本相应增加。

质量成本包括预防成本和缺陷成本。其中预防成本是为确保项目质量而进行预防工作所耗费的费用。而缺陷成本是为确保项目质量而修复缺陷工作所耗费的费用。本着预防重于事后检查的原则，预防成本应该大于缺陷成本。有时，预防成本可以称为一致性成本，而缺陷成本称为非一致性成本。

预防成本包括评估费用和预防费用。评估费用是使项目符合所提要求（第一次）检测缺陷所衍生的成本，例如，质量审计、测试、V&V等。预防费用是使项目符合所提要求预防失败所衍生的成本，例如，用户满意确定、过程评审、改进等。

缺陷成本包括内部费用和外部费用。内部费用是对于不能符合所提要求、尚未发行的软件（返工）所衍生的费用，例如，缺陷标记、返工、重新测试等。外部费用是对于已经发布但是不符合要求的软件所衍生的费用，例如，技术支持、问题估计、修正、索赔等。

软件质量是在开发过程中形成，不是后期测试出来的，如果开发人员认为可以通过后期的测试来提高产品的质量，这是个错误的想法，一个高质量的产品是开发出来的，后期的测试不能真正提高产品的质量，只能靠前期的质量预防和质量检测，例如代码走查，单元测试，对等评审等。前期的质量成本应该高于后期的质量成本，所以在安排计划的时候，应该注意质量活动的时间安排和质量成本的

合理安排。质检/过失比是一个有用的质量测量方法，它可以说明质量管理的程度。质检/过失比 = 预防成本/缺陷成本，这个值大于 2 是努力达到的程度，如果质检/过失比小于 1，则后期测试阶段会发现很多错误。质量保证中的过程审计、产品审计以及质量控制中的测试、对等评审等就是预防成本；而出错后的返工、缺陷跟踪以及诉讼和维护的费用等是缺陷成本。质量成本还包括项目返工的管理时间、丧失的信誉、丧失的商机和客户好感的丧失，丧失的财产等的费用，也许还有更多的其他费用。克鲁斯比的质量理论认为质量用预防成本来衡量，即一致性成本来衡量，因为质量的形成不能靠缺陷成本来拟补。

### 5.3.2 质量计划的方法

质量计划是确定哪种质量标准适合项目并决定如何达到这些标准的过程。质量计划是规划阶段的一个基本过程：每个提交结果都有质量检查的衡量标准。编制项目的质量计划，首先必须确定项目的范围、中间产品和最终产品，然后明确关于中间产品和最终产品的有关规定、标准，确定可能影响产品质量的技术要点，并找出能够确保高效满足相关规定、标准的过程方法。编制质量计划通常对项目进行分析，确定需要监控的关键元素，并制定质量标准。

在进行质量计划时，有很多方法可以采用，例如试验设计、基准对照、质量成本分析、流程图方法、因果分析图等。

1) 试验设计是一种统计学方法，确定哪些因素可能会对特定变量产生影响，是一个不错方法。它是在可选的范围内，对特定要素设计不同的组合方案，通过推演和统计，权衡结果，来寻求优化方案。例如，对成本和时间就可以设计不同的组合方案，并筛选出最优的组合。可以确定在一个项目中的哪些变量是引起项目出现问题的主要原因。

2) 基准对照是一种寻找最佳实践的方法，是利用其他项目的实施情况作为当前项目性能衡量的标准。它通过审查项目的提交结果、项目管理过程、项目成功或者失败的原因等来衡量本项目的绩效。

3) 质量成本分析也是常用的方法，质量成本是为了达到满足用户期望的交付结果的质量要求而花费的所有成本。这包括了为满足质量需求而做的所有工作和解决不合格项而付出的花费。当不合格项需要返工、需要浪费资源时，这个成本是最明显的。所以，质量计划必须进行质量成本的综合分析，以便决定质量活动。

4) 流程图方法可以显示系统的各种成分之间的相互关系，帮助我们预测在何处可能发生何种质量问题，并由此帮助开发处理他们的办法。

5) 因果分析图也称鱼刺图，对于复杂的项目，编制质量计划时可以采用因果分析图，如图 5-4 所示。描述相关的各种原因和子原因如何产生潜在问题或影响，将影响质量问题的“人员、设备、参考资料、方法、环境”等各方面的原因进行细致的分解，方便地在质量计划中制定相应的预防措施。

其次，质量计划中还必须确定有效的质量管理体系，明确质量监理人员对项目质量负责和各级质量管理人员的权限。戴明环（又名 PDCA 循环法）作为有效的管理工具在质量管理中得到广泛的应用，它采用计划—执行—检查—措施的质量环，质量计划中必须将质量环上各环节明确落实到各责任单位，才能保证质量计划的有效实施。

### 5.3.3 质量计划的编写

质量计划应说明项目管理小组如何具体执行它的质量策略。质量计划的目的是规划出哪些是需要被跟踪的质量工作，并建立文档。此文档可以作为软件质量工作指南，帮助项目经理确保所有工作按

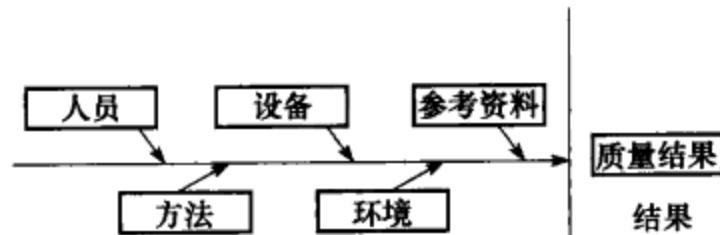


图 5-4 因果分析图

计划完成。开发一个好的、具有针对性的质量计划可以为软件项目带来很多益处：

- 交付一些可靠的、具有特色的、可用的和可维护的产品。
- 项目实施得到控制，降低项目延期交付和成本超出的风险。
- 普遍提高后续软件产品的质量。

严格一致的质量计划不仅有利于产品和客户，而且也有助于项目经理对项目的维护控制和降低成本。质量计划可以是正式的或非正式的，高度细节化的或框架概括型的，皆以项目的需要而定。但是，作为质量计划应满足下列要求：

- 应达到的质量目标和所有特性的要求。
- 确定质量活动和质量控制程序。
- 项目不同阶段，职责、权限、交流方式以及资源分配。
- 确定采用的控制手段，合适的验证手段和方法。
- 确定和准备质量记录。

在质量计划中应该明确项目要达到的质量目标，例如：

- 1) 可用度：可用度指软件运行后在任意随机时刻需要执行规定任务或完成规定功能时，软件处于可使用状态的概率。
- 2) 初期故障率：初期故障率指软件在初期故障期（一般以软件交付给用户后的三个月内为初期故障期）内单位时间的故障数。一般以每100小时的故障数为单位。可以用它来评价交付使用的软件质量与预测什么时候软件可靠性基本稳定。初期故障率的大小取决于软件设计水平、检查项目数、软件规模、软件调试彻底与否等因素。
- 3) 偶然故障率：指软件在偶然故障期（一般以软件交付给用户后的四个月以后为偶然故障期）内单位时间的故障数。一般以每1000小时的故障数为单位，它反映了软件处于稳定状态下的质量。
- 4) 平均失效前时间（MTTF）：指软件在失效前正常工作的平均统计时间。
- 5) 平均失效间隔时间（MTBF）：指软件在相继两次失效之间正常工作的平均统计时间。在实际使用时，MTBF通常是指当n很大时，系统第n次失效与第n+1次失效之间的平均统计时间。对于失效率为常数和系统恢复正常时间很短的情况下，MTBF与MTTF几乎是相等的。国外一般民用软件的MTBF大体在1000小时左右。对于可靠性要求高的软件，则要求在1000~10000小时之间。
- 6) 缺陷密度（FD）：指软件单位源代码中隐藏的缺陷数量。通常以每千行无注解源代码为一个单位。一般情况下，可以根据同类软件系统的早期版本估计FD的具体值。如果没有早期版本信息，也可以按照通常的统计结果来估计。“典型的统计表明，在开发阶段，平均每千行源代码有50~60个缺陷，交付后平均每千行源代码有15~18个缺陷”。
- 7) 平均失效恢复时间（MTTR）：指软件失效后恢复正常工作所需的平均统计时间。对于软件，其失效恢复时间为排除故障或系统重新启动所用的时间，而不是对软件本身进行修改的时间（因软件已经固化在机器内，修改软件势必涉及重新固化问题，而这个过程的时间是无法确定的）。

在质量计划中非常重要的一个任务是提供项目执行的过程程序，例如：项目计划的过程程序、项目跟踪的过程程序、需求分析的过程程序、总体设计的过程程序、详细设计的过程程序、质量审计的过程程序、配置管理的过程程序、测试的过程程序等。

编制一份清晰的质量计划是实施项目质量管理的第一步，而一个清晰的质量计划首先要明确采用的质量标准和质量目标。质量政策和质量标准是编制质量计划的约束条件。质量计划描述项目管理团队如何实施组织的质量策略过程，是项目计划的一个输入。

软件项目的质量计划要根据项目的具体情况决定采取的相应的计划形式，没有统一的定律。有的质量计划只是针对质量保证的计划，有的质量计划既包括质量保证计划也包括质量控质计划。

质量计划可以包括质量保证和质量控制的活动安排，包括质量保证的（审计、评审软件过程、活

动和软件产品等)的方法、职责和时间安排等;质量控制计划可以包含在开发活动计划中,例如代码走查、单元测试、集成测试、系统测试等。

质量计划中要明确质量管理组织,职责和义务。其中质量保证的人员应该有特殊的问题上报渠道,以保证问题的顺利解决。但是这个渠道,作为质量保证人员应该慎用。项目经理是项目质量管理的最终责任承担者。

下面给出一个参考的质量计划模板。

1. 导言
2. 项目概述
  - 2.1 功能概述
  - 2.2 项目生存期模型
  - 2.3 项目阶段划分及其准则
3. 实施策略
  - 3.1 项目特征
  - 3.2 主要工作
4. 项目组织
  - 4.1 项目组织结构
  - 4.2 质量组的权利
  - 4.3 质量组织及职责
5. 质量对象分析及选择
6. 质量任务
  - 6.1 基本任务
  - 6.2 活动反馈方式
  - 6.3 争议上报方式
  - 6.4 采购产品的验证和确认
  - 6.5 客户提供产品的验证
7. 实施计划
  - 7.1 工作计划
  - 7.2 高层管理定期评审安排
  - 7.3 项目经理定期和基于事件的评审
8. 资源计划
9. 记录的收集、维护与保存
  - 9.1 记录范围
  - 9.2 记录的收集、维护和保存

## 5.4 质量体系

在 ISO 9000 的术语中,对质量体系的描述是:“组织结构、责任、工序、工作过程及具体执行质量管理所需的资源”。即质量体系是为实施质量管理所需的组织结构、程序、过程和资源,如图 5-5 所示。企业的质量体系是根据企业质量方针和实施策略建立的,以保证达到企业的质量目标。

质量体系是一个企业质量管理体系的规范,包含的内容比较全,是企业长期遵循和需要重复实施的文件,具有较强的标准性,可以参照一定的标准实施,例如参照 CMM, ISO 等标准。质量计划可以参照质量体系编写,是一次性实施的,项目结束,质量计划的有效性就结束了。质量体系内容的核心是建立、执行和维护(改善)软件生产过程,以保证最终

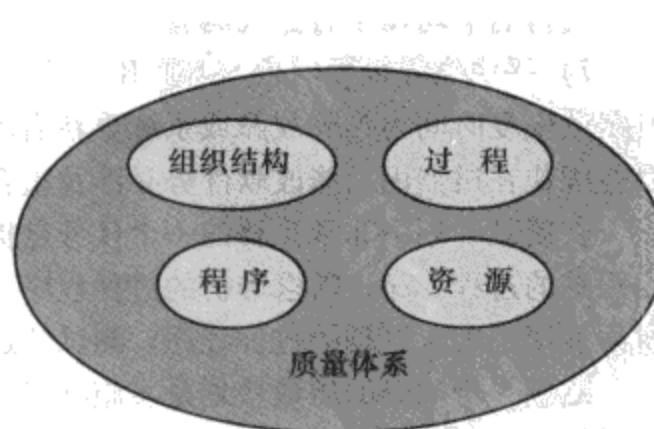


图 5-5 质量体系图示

生产出的软件产品达到用户综合的质量要求。图 5-6 是某公司的质量体系的一个框架，图中表达了质量体系要素和各要素之间的关系。

其中：

- 组织：包括三类组织，即过程改善组织、过程执行组织和过程质量保证组织。过程改善组织负责建立和维护整个企业基于过程的质量体系。过程执行组织具体执行质量体系中相关的过程规范和标准。过程质量保证组织负责过程执行的质量控制和基于过程生产出的产品的质量控制。

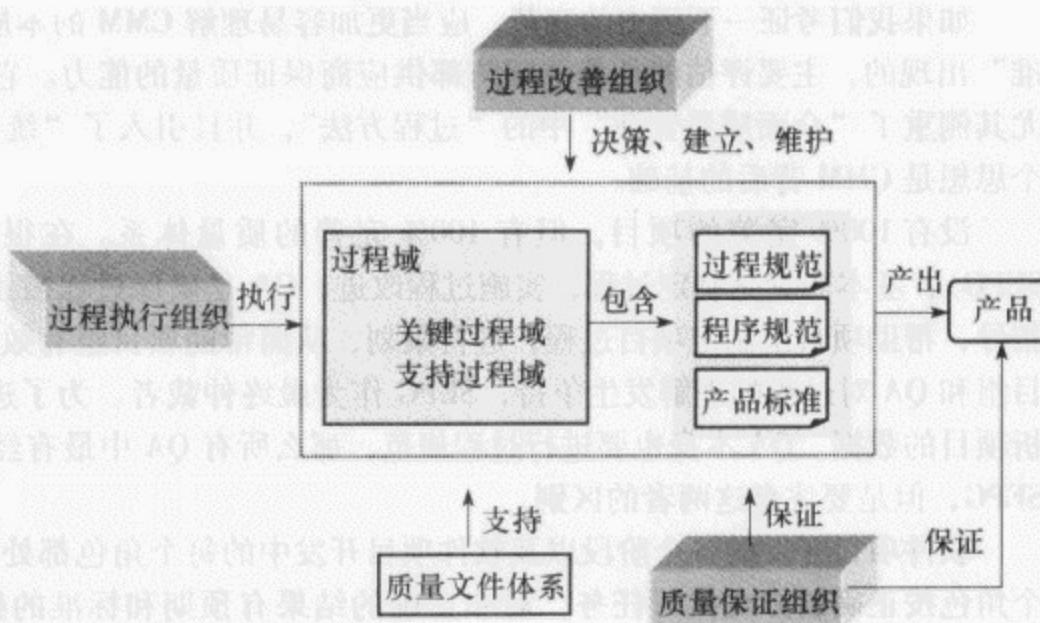


图 5-6 质量体系要素关系图

- 过程域：过程域包括关键过程域和支持过程域。关键过程域的定义包括关键过程实施规范和程序的定义以及工作产品标准的定义。关键过程域的执行是质量保证的对象之一。支持过程域是关键过程域的补充和支持性过程实施规范。
- 质量文件体系：质量文件体系是过程实施的文档化规范和产品标准。
- 产品：产品包括工作产品（即过程阶段中产生出的产品）和提交产品（即向客户提交的最终产品）。所有这些产品需通过确定的过程或步骤产生，并作为质量保证的对象之一。

提高软件质量最好的办法是在开发过程中有效地防止工作成果产生缺陷，将高质量建立于开发过程之中。即苦练内功，通称为“软件过程改进”。

软件过程改进（Software Process Improvement, SPI）帮助软件企业对其软件开发过程的改进进行计划、制定措施以及实施措施。其实施对象就是软件企业的软件过程，也就是软件产品的生产过程，当然也包括软件维护之类的维护过程。

在世界范围内，软件项目需求正以非常快的速度增长，并且这种增长看起来还远未达到顶端。这种增长已经导致软件开发活动急剧性的增长，使得用于构筑软件的过程得到更多的关注。软件过程可以定义为人们用来开发和维护软件以及相关产品（如：工程计划、设计文档、规章、检测事例及用户手册）的一组活动、方法、实践及转换。软件过程重要性的提高已经引起了对软件过程改进的要求，这就需要过程分析和评估的方法。

对于软件企业来说，软件过程是整个企业最复杂、最重要的业务流程，软件产品就是软件企业的生命，改进整个企业的业务流程，最重要的还是要改进它的软件过程。多年以来，人们认识到要想高效率、高质量和低成本地开发软件，必须以改善软件生产过程为中心，全面开展软件工程和质量管理。这是世界各国软件产业都要走的路，我国软件产业之所以落后，不是因为技术落后，而是软件开发的管理落后。CMM 就是结合了质量和软件工程的双重经验而制定的一套针对软件生产过程的规范。由此可见，对软件生产过程的管理在整个软件企业的管理中起了决定性作用。

然而，现在许多实施 CMM 的人员沉溺于 CMM 的理论，过于强调“过程”，这是很危险的倾向。这个思想倾向在国外受到了猛烈抨击，从某种意义上各种敏捷过程（XP）方法的提出就是对强调过程的一种反思。XP 中的一个思想“人比过程更重要”是值得我们思考的。根据现代软件工程对众多失败项目的调查，发现管理是项目失败的主要原因。这个事实的重要性在于说明了“要保证项目不失败，我们应当更加关注管理”，注意这个事实没有说明另外一个问题“良好的管理可以保证项目的成功”。

现在很多人基于一种粗糙的逻辑，从一个事实反推得到的这个结论，在逻辑上是错误的，这种错误形成了更加错误的做法。

如果我们考证一下历史的变革，应当更加容易理解 CMM 的本质。CMM 首先是作为一个“评估标准”出现的，主要评估的是美国国防部供应商保证质量的能力。它引入了“全面质量管理”的思想，尤其侧重了“全面质量管理”中的“过程方法”，并且引入了“统计过程控制”的方法。可以说这两个思想是 CMM 背后的基础。

没有 100% 完美的项目，但有 100% 完善的质量体系。在很多企业中，有过程改进组织（即 SEPG），基本职责是制定过程，实施过程改进；QA 是确保过程被正确执行。SEPG 应当提供过程上的指导，帮助项目组制定项目过程，进行策划，从而帮助项目组有效地工作，有效地执行过程。如果项目组和 QA 对过程的理解发生争持，SEPG 作为最终仲裁者。为了进行有效的过程改进，SEPG 必须分析项目的数据。QA 本身也要进行过程规范，那么所有 QA 中最有经验、最有能力的 QA 人员可以参加 SEPG，但是要注意这两者的区别。

软件项目开发的各个阶段以及软件项目开发中的每个角色都处在质量管理体系之中，因此只要每个角色按正确的方式执行任务，对非正确的结果有预期和标准的处理方法，保证项目质量自然水到渠成。

## 5.5 软件质量改善的建议

软件质量的改善是一个巨大的挑战，以前，人们将软件质量改善都归结为测试的问题，但是现在，一个讲究效率的质量过程远远不只是测试。软件质量改善是对软件质量保证的一种承诺。如果急功近利，不但会做很多浪费人力和物力的无效工作，还会给客户留下不好的印象。为了更好地进行软件质量的改善，现提出如下建议：

- 不但要主观认识到质量的重要性，同时还要落实到行动中。把想法落实到实际工作中是做好软件质量管理的第一原则。
- 软件质量活动必须经过规划、必须明文规定。
- 树立提高质量就是尊重客户的思想。在软件产业发达的今天，已经是客户的买方市场，客户永远会选择质量和服务都表现良好的产品来满足自己的需求。因此，我们应该尊重客户，把客户放在“上帝”的位置上，认真做好质量工作。
- 质量活动必须尽早开始。
- 质量小组尽可能独立存在。
- 质量小组的人应该经过必要的培训。
- 建立规范的质量保证体系，逐步使软件开发进入良性循环状态。在没有开发规范的前提下，软件团队是不能开发出高质量软件的。因此软件团队一定要建立规范的质量保证体系，同时把规范体系逐步落实到工作中。

## 5.6 校务通系统案例分析

在制定项目计划的时候，质量经理参与整个项目计划的制定过程，同时负责制定质量计划，根据项目的特点和企业的质量保证体系，制定项目质量计划。

### 校务通系统质量计划

#### 1. 导言

略。

#### 2. 项目组织

##### 2.1 组织机构

在项目实施期间成立质量保证组织，该组织由质量保证人员和项目经理组成，项目经理负责质量监督

工作及项目进展过程中各环节的质量把关，开发经理负责质量控制的工作，质量保证人员负责质量保证的工作。组织结构图如图1所示：

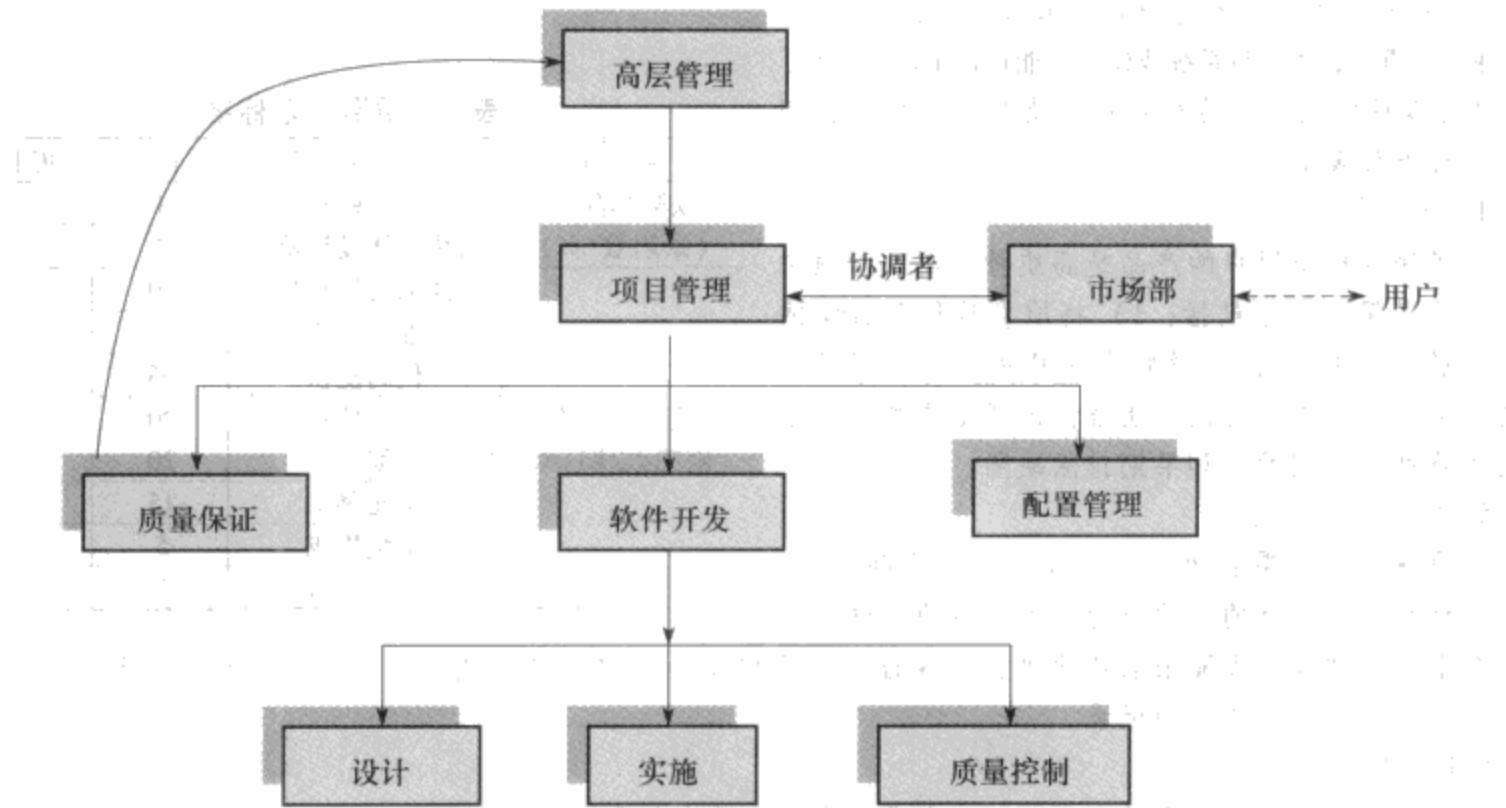


图1 项目的组织结构

## 2.2 职责

在本项目中，质量保证组织的职责如下：

### 2.2.1 高层管理

高层管理是公司负责质量的高级管理，其质量职责如下：

- 受理项目内不能解决的不符合问题，必要时与项目经理协调；
- 负责听取质量保证组的工作报告，评审质量保证活动和结果；
- 参加有关质量保证过程改进的评审。

### 2.2.2 项目的质量保证人员

质量保证人员的质量职责如下：

- 负责项目实施过程中对项目实施情况进行监督，包括对项目实施过程和工作产品进行监督检查；
- 实施项目组成员的质量保证培训；
- 制定质量保证计划；
- 按计划实施审计活动，依照质量保证计划执行评审/审计，并记录执行中发现的不符合项；
- 对不符合问题提交不符合项报告，跟踪并验证纠正措施的执行情况；
- 对项目内不能解决的不符合项问题向高层管理提交报告；
- 向项目经理报告项目质量工作状况和质量度量结果；
- 定期向项目组报告质量活动的结果；
- 制订质量保证的过程改进计划，记录过程数据。

### 2.2.3 项目经理

项目经理的质量职责如下：

- 评审质量计划；
- 与质量保证人员一起协商不符合项问题的纠正措施，并安排资源实施纠正措施；
- 定期或事件驱动的评审质量保证活动和结果。

## 3. 质量目标

根据企业的质量方针和质量目标，结合本项目特点，制定项目的总体质量目标：

- 1) 基于需求的测试覆盖率为 100%；

- 2) 软件功能测试用例通过率不低于 95%;
- 3) 每个阶段评审中发现的问题都已经解决或得到适当处理;
- 4) 产品发布时不存在严重及其以上的缺陷。

注：严重问题指导致系统或模块不能正常工作的问题。

结合以往的项目经验和企业的质量相应标准，制定质量标准如表 1 所示。

#### 4. 质量策略

为了保证提交用户的产品是高质量，实施过程中采取的质量保证措施包括：1) 将质量贯彻到日常的项目进展过程中；2) 应该特别注意项目工作产品质量的早期评审工作，无论是质量保证还是质量控制采取的策略都是早期预防和早期排除缺陷。

#### 5. 质量保证活动

质量保证的主要活动包括过程审计和产品审计。过程审计和产品审计的目的是为了确保在项目进展过程的各个阶段和各个方面采取各项措施来保证和提高提交给用户的产品质量。每一次过程审计和产品审计都应填写相应的报告或活动记录。

##### 5.1 产品审计

产品审计由质量保证人员来进行，检查项目产品是否达到质量目标。

质量保证人员对项目生存期中创建的工作产品可以有选择性的进行审计，以验证是否符合适当的标准，是否进行了质量检查。表 2 便是质量审计一览表。

##### 5.2 过程审计

项目严格按照组织定义的软件过程进行开发，过程审计的具体依据参照企业的过程规范，保证项目中的所有过程活动都在实施范围内。在每次审计之后，要对审计结果做

出明确的决策并形成评审记录。审计可采取文件传阅、评审会等形式。

质量保证人员负责对项目过程进行监督，发现的问题和解决情况在每周的例会上通报，对没有解决的问题进行讨论，对不能解决的问题提交高级管理者处理。

每个周末，进行一次配置管理审核，确认配置管理工作是否正常进行。

根据公司的质量保证体系和本项目的具体特点，确定项目执行过程如下：

- 1) 项目规划过程及产品标准。
- 2) 项目跟踪管理过程。
- 3) 需求分析过程及产品标准。
- 4) 系统设计过程及产品标准。
- 5) 详细设计过程及产品标准。
- 6) 调试运行过程及产品标准。
- 7) 代码走查过程及代码编写标准。
- 8) 产品集成测试过程及产品标准。
- 9) 开发环境中的执行规则。
- 10) 测试环境中的执行规则。
- 11) 质量保证过程及其标准。

表 1 质量计划标准

项目 缺陷排除率 (缺陷数/页)	具体描述	计划	实际
		需求检查	4
缺陷排除率 (缺陷数/KLOC)	系统总体设计检查	2	
	详细设计复核	30	
	详细设计检查	10	
	代码复核	65	
	代码检查	20	
	编译	20	
	单元测试	15	
	系统集成	5	
	系统测试	5	

表 2 审计产品一览表

项	审计对象	审计阶段	参照的标准
1	软件项目计划	计划结束	企业质量体系
2	软件配置管理计划	计划结束	企业质量体系
3	软件质量保证计划	计划结束	企业质量体系
4	总体设计文档	设计结束	企业质量体系和项目计划
5	详细设计文档	设计结束	企业质量体系和项目计划
6	数据库表和编码规范	设计结束	企业质量体系和项目计划
7	产品代码	每个阶段实施结束	企业质量体系和项目计划
8	测试报告	测试结束	企业质量体系和项目计划
9	测试计划	设计结束	企业质量体系和项目计划
10	用户文档	测试结束	企业质量体系和项目计划

12) 配置管理过程及其标准。

具体过程执行标准详见企业质量体系和项目计划的过程附件。

#### 6. 质量控制活动

质量控制活动包括代码走查、单元测试、集成测试、环境测试等，由开发人负责，详见进度计划。编码人员在编写代码时要进行同步单元测试，单元测试要达到分支覆盖，产品通过单元测试和编码检查后，应提交测试部进行集成测试、系统测试。测试部的测试应达到质量目标要求，软件发布时应达到测试通过准则的要求。

#### 7. 质量保证的报告途径

质量保证人员对每次审计活动发现的不符合项，应该和项目经理协商不符合项的纠正措施，及预定完成日期，若和项目经理存在意见分歧，质量保证人员可以上报给高层管理者，高层管理者决定最后的措施。同时不符合项在项目周例会中汇报。

对不符合项，质量保证人员要在预定完成日期内重新审计、验证不符合项的纠正情况，若超过预定完成日期1周仍然有没解决的不符合项，质量保证人员上报给高级管理者，由高级管理者决定最后的措施。

质量保证人员有独立的汇报途径，日常的汇报途径如下：

- 发现的问题通知项目经理，协调纠正措施。
- 将项目组内不能协调的问题汇报给高级管理者，由高级管理者协调解决。
- 日常工作和过程数据要汇报给质量经理统一收集、统计。

#### 8. 记录的收集、维护和保存

项目组应当保留项目执行过程中形成的各类文档、各种记录、各级周报、各级会议记录、对于项目中问题的处理也需要形成记录保存。每周由质量保证人员根据任务清单的审计任务进行审计活动，并收集各活动的过程数据。

### 5.7 小结

软件质量是软件满足用户明确说明或者隐含说明的需求的程度，用户的满意度是质量非常重要的要素。保证软件的质量非常重要，低质量的软件可以导致后期成本的增加。质量的管理过程包括质量计划、质量保证、质量控制三个过程。质量保证是验证软件是否正确，而质量控制是验证软件是否正确构造。所有的质量标准和质量活动都需要质量计划来进行规划，它说明软件项目中的质量活动任务，采用的相应方法、对策，以及出现问题时的处理方式。质量计划最好由独立的质量人员负责。

### 5.8 练习题

#### 一、选择题

1. 项目质量管理的最终责任由谁来承担？（ ）  
A. 项目开发人员      B. 采购经理  
C. 质量经理      D. 项目经理
2. “质量成本”是一个项目管理概念，它说明了下列哪项成本？（ ）  
A. 额外需求的成本      B. 需求变更的成本  
C. 确保符合需求的成本      D. 固定成本
3. 增加有益的活动过程减少没有价值的活动过程是哪类质量活动？（ ）  
A. 质量保证      B. 持续的过程改进  
C. 质量控制      D. 质量改进
4. 质量控制是（ ）。  
A. 对每个工作包增加工作时间      B. 项目生存期的各个阶段都需要实施的  
C. 只需要做一次      D. 只有大的项目才需要的
5. 质量管理计划与质量体系的区别在于（ ）。  
A. 质量计划是针对某一产品、项目、服务和合同制定的

- B. 质量管理是针对单一的产品、项目、服务和合同制定的
  - C. 质量管理是由一个单独的组织实体使用
  - D. 质量计划不属于项目计划的一部分
6. 项目质量管理的目标是满足（ ）的需要。
- A. 老板
  - B. 干系人
  - C. 项目
  - D. 组织
7. 质量与等级的区别是什么？（ ）
- A. 质量是对需求的满足程度，而等级是对质量的排序
  - B. 质量是对需求的满足程度，而等级是对货物和服务的排序
  - C. 在项目管理过程中，质量和等级没有明显的区别
  - D. 质量是项目的最终结果，而等级是项目进展过程中对质量的排序
8. 下面除了哪项都不是质量计划的方法（ ）。
- A. 质量检查
  - B. 对等评审
  - C. 抽样分析
  - D. 试验设计

二、判断题

- 1. 质量计划中可以确定质量保证人员的特殊汇报渠道。（ ）
- 2. 质量计划可以在项目进行过程中编写，不一定是前期。（ ）

## ■ 软件项目人力资源计划

软件产品是众多软件开发人员的智慧结晶，人是软件项目中最重要的因素。如何建设高效的软件项目开发团队是软件项目顺利实施的保证。本章进入本书路线图的人力资源管理计划，如图 6-1。

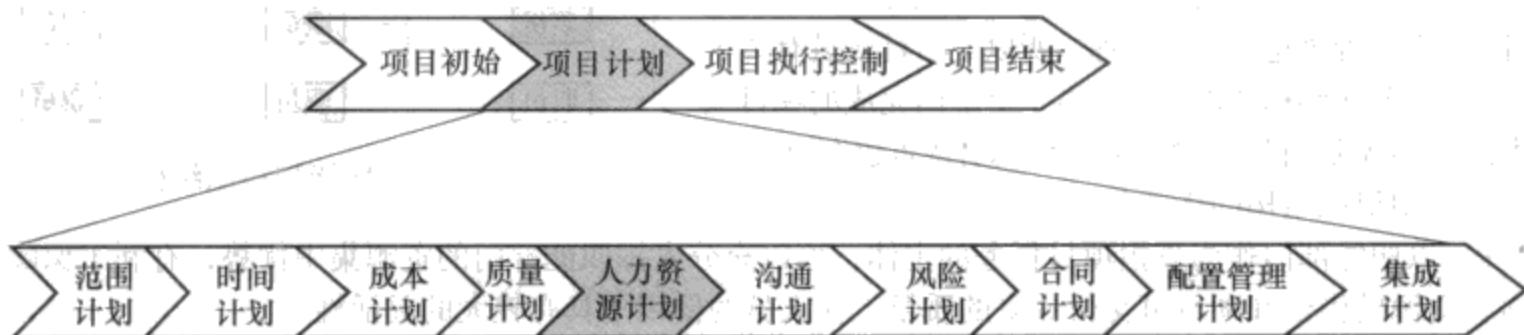


图 6-1 路线图——项目计划 - 人力资源计划

### 6.1 相关概念

一个软件项目涵盖了项目组、客户、客户需求（或者称为项目目标）以及为达到项目目标、满足客户需求所需要的权责、人员、时间、资金、工具、资料、场所等项目资源。而人员无疑是项目资源中最特别最重要的资源，人具备主动性和情感，与社会、家庭、企业、员工等的关系密不可分。影响软件项目进度、成本、质量的因素主要是“人、过程、技术”。首先要明确的是这三个因素中，人是第一位的。“人才是企业的最重要的资源”，人力资源可以决定项目的成败，有效的人力资源管理是项目管理者的一个很大的挑战。

软件项目人力资源管理是项目管理中根本的一项管理。人力资源管理是保证参加项目人员能够被最有效使用所需要的过程，是对项目组织所储备的人力资源开展的一系列科学规划、开发培训、合理调配、适当激励等方面的工作，使项目组织各方面人员的主观能动性得到充分发挥，做到人尽其才、事得其人、人事相宜，同时保持项目组织高度的团结性和战斗力，从而成功的实现项目组织的既定目标。

项目中的人力资源一般是以团队的形式存在的，团队是由一定数量的个体组成的集合，这个团队包括企业内部的人、供应商、承包商、客户等。通过将具有不同潜质的人组合在一起，形成一个具有高效团队精神的队伍来进行软件项目的开发。团队开发是发掘作为个体的个人能力，然后是发掘作为团队的集体能力。当一组人称为团队的时候，他们应该承诺为一个共同的目标工作，每个人的努力必须协调一致，而且能够愉快的在一起合作，从而开发出高质量的软件产品。

本章的主要任务是讨论项目的人力资源计划，包括确定项目中的角色、职责、汇报关系、产生项目的组织结构图、人员管理计划等。

### 6.2 项目组织结构

组建团队时首先要明确项目的组织结构，项目组织结构应该能够增加团队的工作效率，避免摩擦，因此，一个理想的团队结构应当适应人员的不断变化，利于成员之间的信息交流和项目中各项任务的协调。

项目组织是由一组个体成员为实现一个具体项目目标而协同工作的队伍，项目组织的根本使命是在项目经理的领导下，群策群力，为实现项目目标而努力工作，项目组织具有临时性和目标性的特点。但是，项目管理中的组织结构可以总结为三种主要类型：职能型、项目型和矩阵型。具体选择什么样的组织结构要考虑多重因素。在这三种组织结构中，矩阵型沟通最复杂，项目型在项目收尾时，团队

成员和项目经理压力比较大。团队组织和用于管理项目的手段之间应构成默契，任何方法上的失谐都很可能导致项目产生问题。

### 6.2.1 职能型组织结构

职能型组织结构是目前最普遍的项目组织形式。它是一个标准的金字塔型的组织形式。见图 6-2。

职能型组织结构是一种常规的线性组织结构。采用这种组织结构时，项目是以部门为主体来承担项目的，一个项目有一个或者多个部门承担，一个部门也可能承担多个项目。有部门经理也有项目经理，所以，项目成员有 2 个上司。这个组织结构适用于主要由一个部门完成的项目或技术比较成熟的项目。职能型组织结构有优点也有缺点。

职能型组织结构的优点是：

- 1) 以职能部门作为承担项目任务的主体，可以充分发挥职能部门的资源集中优势，有利于保障项目需要资源的供给和项目可交付成果的质量。在人员的使用上具有较大的灵活性。
- 2) 职能部门内部的技术专家可以被该部门承担的不同项目共享，节约人力，减少了资源的浪费。
- 3) 同一职能部门内部的专业人员便于相互交流、相互支援，对创造性的解决技术问题很有帮助。同部门的专业人员易于交流知识和经验，项目成员事业上具有连续性和保障性。
- 4) 当有项目成员调离项目或者离开公司，所属职能部门可以增派人员，保持项目的技术连续性。
- 5) 项目成员可以将完成项目和完成本部门的职能工作融为一体，可以减少因项目的临时性而给项目成员带来的不确定性。

职能型组织结构的缺点如下：

- 1) 客户利益和职能部门的利益常常发生冲突，职能部门会为本部门的利益而忽视客户的需求，精力只集中于本职能部门的活动，项目及客户的利益往往得不到优先考虑。
- 2) 当项目需要多个职能部门共同完成，或者一个职能部门内部有多个项目需要完成时，资源的平衡就会出现问题。
- 3) 当项目需要由多个部门共同完成时，权力分割不利于各职能部门之间的沟通交流、团结协作。项目经理没有足够的权利控制项目的进展。
- 4) 项目成员在行政上仍隶属于各职能部门的领导，项目经理对项目成员没有完全的权利，项目经理需要不断地同职能部门经理进行有效的沟通以消除项目成员的顾虑。当小组成员对部门经理和项目经理都要负责时，项目团队的管理常常是复杂的。对这种双重报告关系的有效管理常常是项目最重要的成功因素，而且通常是项目经理的责任。

### 6.2.2 项目型组织结构

与职能型组织结构相对应的另一种组织是项目型组织结构，项目型组织结构中的部门完全是按照项目进行设置的，是一种单目标的垂直组织方式，存在一个项目就有一个类似部门的项目组，当项目完成之后，这个项目组代表的部门就解散了，这时项目人员的去向就是一个问题了。所以，这种组织结构不存在原来意义上的部门的概念。每个项目以项目经理为首，项目工作会运用到大部分的组织资源，而项目经理也有高度独立性，享有高度的权力。完成每个项目目标所需的全部资源完全划分给该项目单元，完全为该项目服务，如图 6-3 所示。

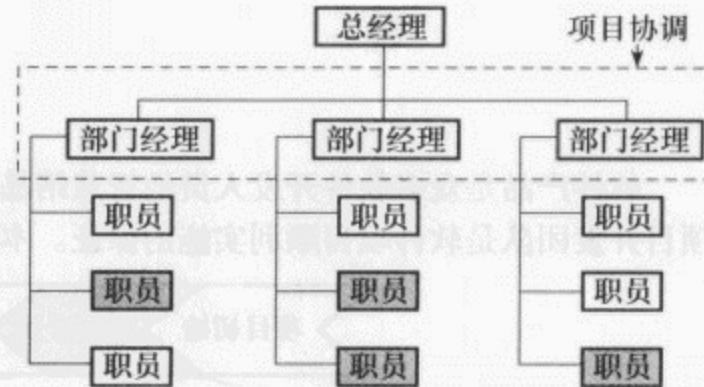


图 6-2 职能型组织结构

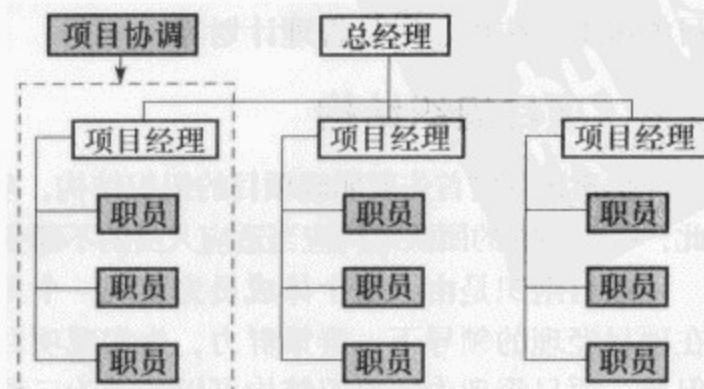


图 6-3 项目型组织结构

项目型组织结构中，项目经理有足够的权力，控制项目的资源。项目成员向唯一领导汇报。这种组织结构适用于开拓型等风险比较大的项目或进度、成本、质量等指标有严格要求的项目，不适合人才匮乏或规模小的企业。项目型组织结构也有自己的优点和缺点。

项目型组织结构的优点是：

- 1) 项目经理对项目可以全权负责。可以根据项目需要随意调动项目组织的内部资源或者外部资源。
- 2) 项目型组织的目标单一，完全以项目为中心安排工作，决策的速度得以加快，能够对客户的要求做出及时响应，项目组团队精神得以充分发挥。有利于项目的顺利完成。
- 3) 项目经理对项目成员有全部权利，项目成员只对项目经理负责，避免了职能型项目组织下项目成员处于多重领导、无所适从的局面，项目经理是项目的真正、唯一的领导者。
- 4) 组织结构简单，易于操作。项目成员直接属于同一个部门，彼此之间的沟通交流简洁、快速，提高了沟通效率，同时也加快了决策速度。

项目型组织结构的缺点是：

- 1) 每一个项目型组织，资源不能共享，即使某个项目的专用资源闲置，也无法应用于另外一个同时进行的类似项目，人员、设施、设备重复配置，会造成一定程度的资源浪费。
- 2) 公司里各个独立的项目型组织处于相对封闭的环境之中，公司的宏观政策、方针很难做到完全、真正的贯彻实施，可能会影响公司的长远发展。
- 3) 在项目完成以后，项目型组织中的项目成员或者被派到另一个项目中去，或者被解雇，对项目成员来说，缺乏一种事业上的连续性和安全感。
- 4) 项目之间处于一种条块分割状态，项目之间缺乏信息交流，不同的项目组很难共享知识和经验，项目成员的工作会出现忙闲不均的现象。

### 6.2.3 矩阵型组织结构

矩阵型组织结构是职能型组织结构和项目型组织结构的混合体，既具有职能型组织的特征又具有项目型组织结构的特征，如图 6-4 所示。它是根据项目的需要，从不同的部门中选择合适的项目人员组成一个临时项目组，项目结束之后，这个项目组也就解体了，然后，各个成员再回到各自原来的部门。团队的成员需要向不同的经理汇报工作。这种组织结构的关键是项目经理需要具备好的谈判和沟通技能，项目经理与职能经理之间建立好工作关系。项目成员需要适应与两个上司协调工作。加强横向连结，充分整合资源，实现信息共享，提高反应速度等方面的优势恰恰符合当前的形势要求。采用该管理方式可以对人员进行优化组合，引导聚合创新，而且同时改变了原有行政机构中固定组合、互相限制的现象。这种组织结构适用于管理规范、分工明确的公司或者跨职能部门的项目。

矩阵型组织结构的优点是：

- 1) 专职的项目经理负责整个项目，以项目为中心，能迅速解决问题。在最短的时间内调配人才，组成一个团队，把不同职能的人才集中在一起。
- 2) 多个项目可以共享各个职能部门的资源。在矩阵管理中，人力资源得到了更有效的利用，减少了人员冗余。研究表明：一般使用这种管理模式的企业能比传统企业少用 20% 的员工。
- 3) 既有利于项目目标的实现，也有利于公司目标方针的贯彻。
- 4) 项目成员的顾虑减少了，因为项目完成后，他们仍然可以回到原来的职能部门，不用担心被解散。而且他们能有更多机会接触自己企业的不同部门。

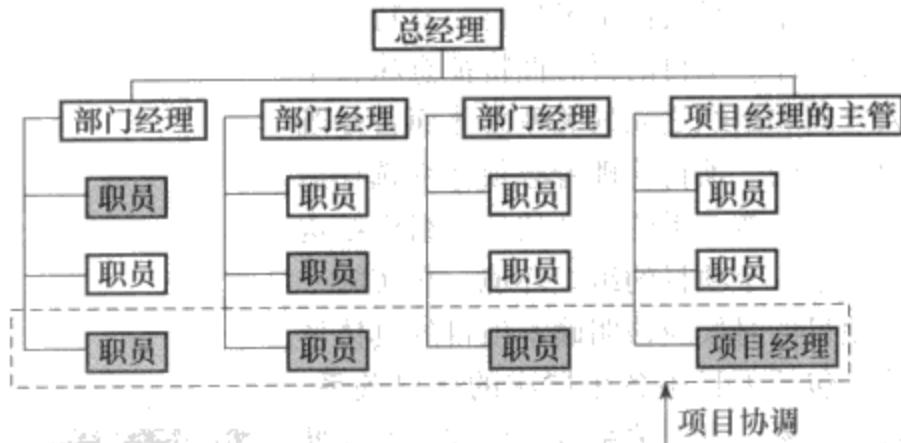


图 6-4 矩阵型组织结构

矩阵型组织结构的缺点是：

- 1) 容易引起职能经理和项目经理权力的冲突。
- 2) 资源共享也能引起项目之间的冲突。
- 3) 项目成员有多位领导，即员工必须要接受双重领导，因此经常能体会到焦虑与压力。当两个经理的命令发生冲突时，他必须能够面对不同指令形成一个综合决策来确定如何分配他的时间。同时，员工必须和他的两个领导保持良好的关系，应该显示出对这两个主管的双重忠诚。

项目是由项目团队完成的，在矩阵型的组织结构中，项目经理和项目成员往往来自不同职能的团队，由于组织职责不同，参与项目的组织在目标、价值观和工作方法上会与项目经理所在部门有所差异，会在项目团队组建之初的磨合阶段出现矛盾，进而产生“对抗”，对于这种由于分工不同，人员相互之间不熟悉产生的对抗，项目经理要能及时识别，并将对抗控制在“建设性”对抗的范围之内，切忌对项目组建初期的磨合放任自流。

矩阵的每一个点其实都有自己的直属上级，都有各自的团队利益，作为项目经理，如果确认了采用矩阵式管理的项目组织模式，就必须能够认同矩阵管理带来的差异，在确保项目整体目标的前提下，务必要和矩阵所涉及的诸多职能部门搞好协同，平衡各自的利益。在沟通中出现问题时，要立刻敏锐感觉到问题并非处在矩阵点，而是出现在矩阵的直线上级，需要花费大量的时间和精力与矩阵的直线上级做好充分沟通。在出现严重问题后，首先要确保自己的站位尽可能超越原来的组织定位，以更高的站位看待各职能的差异，更多的以沟通、认同、甚至妥协的方式顾全大局，确保项目的成功。如果不能超越自己的组织利益，且没有时间或精力做大量的沟通工作，仍旧习惯于高职责、高压力、强绩效导向的管理模式，那么职能型项目组织也许是最好的选择。

其实，很多的组织结构不同程度地具有以上各种组织类型的结构特点，而且会根据项目具体情况执行一套特定的工作程序。项目的暂时性特征意味着个人之间和组织之间的关系总体而言是既短又新的。项目管理者必须仔细选择适应这种短暂关系的管理技巧。组织结构说明了一个项目的组织环境，在实施一个项目的时候应该明确本项目的具体形式，包括项目中各个层次的接口关系、报告关系、责任关系等。例如图 6-5 便是一个网站项目组织结构案例。

为了创建一个组织结构图，项目管理者首先明确项目需要的人员类型，需要熟悉 Oracle 还是 DB2，需要精通 Java 语言的还是 C++ 语言的等等。项目中的人员有不同的背景、掌握不同的技能，管理这样的团队不是件容易的事情，创建一个组织结构图是必须的。

项目经理完成任务分解结构（WBS）分解之后，可能开始考虑如何将各个独立的工作单元分配给相应的组织单元。采用任务分解结构可以与组织分解结构（Organizational Breakdown Structure，OBS）综合使用，建立一个任务职责的对应关系，如图 6-6 所示。

组织分解结构是一种组织结构图，显示组织

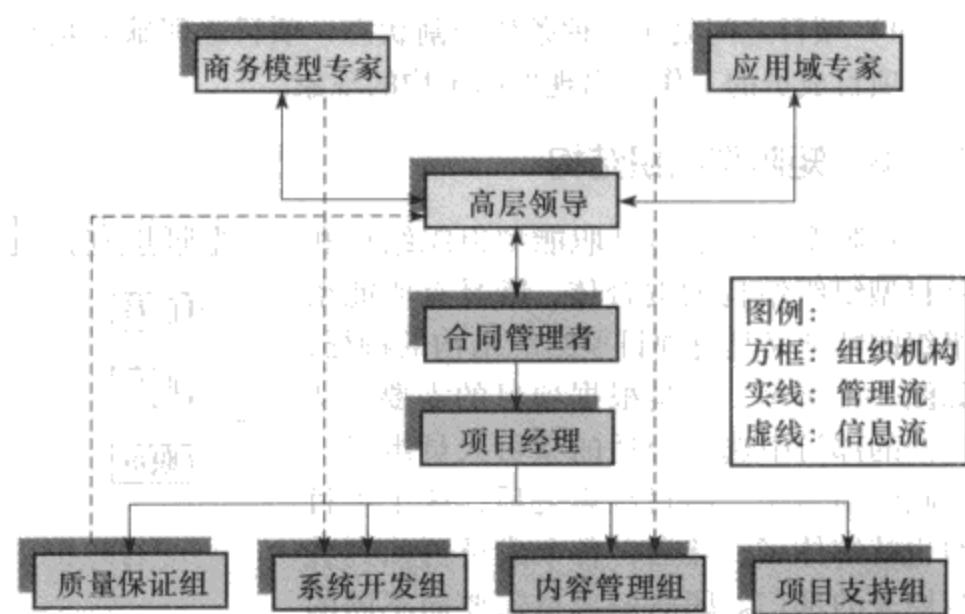


图 6-5 软件项目组织结构

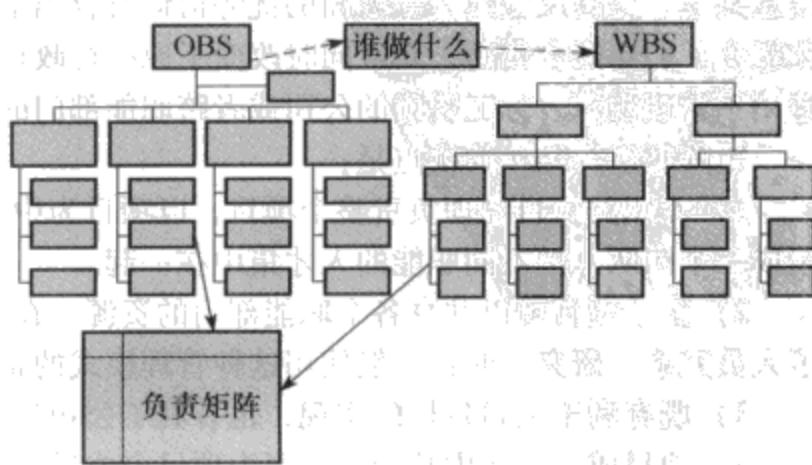


图 6-6 WBS 与 OBS 的对应关系

中哪个单位负责哪项工作任务。也可以是对一般组织结构再进行详细分解。确定了组织结构图之后，项目经理需要确定组织结构中的责任分配。

### 6.3 责任分配矩阵

项目团队内部有时由于各阶段不同角色或同阶段不同角色之间的责任分工不够清晰而造成工作互相推诿、责任互相推卸的现象；有时各阶段不同角色或同阶段不同角色之间的责任分工比较清晰，但是各项目成员只顾完成自己那部分任务，不愿意与他人协作。这些现象都将造成项目组内部资源的损耗，从而影响项目进展。项目经理应当对项目成员的责任进行合理的分配并清楚地说明，同时应强调不同分工、不同环节的成员应当相互协作，共同完善。

责任分配矩阵（responsibility assignment matrix, RAM）是用来对项目团队成员进行分工，明确其角色与职责的有效工具，通过这样的关系矩阵，项目团队每个成员的角色，谁做什么，以及他们的职责等都得到了直观的反映。项目的每个具体任务都能落实到参与项目的团队成员身上，确保了项目的每项任务有人做，每个人有任务做。

责任分配矩阵是一种矩阵图，表6-1是一个责任分配矩阵，横向为工作单元，纵向为组织成员或部门名称。纵向和横向交叉处表示项目组织成员或部门在某个工作单元中的职责。矩阵中的符号表示项目工作人员在每个工作单元中的参与角色或责任。采用责任矩阵来确定项目参与方的责任和利益关系。责任分配矩阵确定了工作职责和执行单位。对于很小的项目这个执行单位最好是分配到了个人，对于大的项目这个执行单位可以是团队或者是一个企业单元。

另外，可以采用责任分配矩阵分配更加详细的工作活动，或者定义一般的角色和职责。例如表6-2就是展示了一个项目的人员是否参加，或者负责项目中的某项活动，是否对项目的某项任务提供输入、进行评审等。

责任分配矩阵可以帮助项目经理标识完成项目需要的资源，同时确认企业的资源库中是否有这些资源。项目后期，项目经理可以使用更加准确的矩阵来标识哪个任务分配给哪个人，明确了项目组织结构，再加上责任分配矩阵，就能够把团队成员的角色和职责，以及汇报关系确定下来，使项目团队能够各负其责、各司其职，进行充分、有效的合作，避免职责不明，为项目任务的完成提供了可靠的组织保证。

### 6.4 人员管理计划

由于项目中所有活动均由人来完成，因此，如何合理化配置项目组成员，充分发挥“人”的作用，对于项目的成败起着至关重要的作用。

软件项目中的开发人员是最大的资源。对人员的配置、调度安排贯穿整个软件过程，人员的组织管理是否得当，是影响软件项目质量的决定性因素。

表 6-1 某项目责任分配矩阵

单位	WBS 任务							
	1.1.1	1.1.2	1.1.3	1.1.4	1.1.5	1.1.6	1.1.7	1.1.8
系统部门	R	RP						R
软件部门			RP					
硬件部门				RP				
测试部门	P							
质量保证部门					RP			
配置管理部门						RP		
后勤部门							P	
培训部门								RP

说明：R：负责人（部门）；P：执行者（部门）。

表 6-2 展示项目人员角色

	项目经理	应用开发人员	网络工程师	专家
建立应用软件	A	C	P	
测试应用软件	A	P	P	
应用软件打包	R		R	P
测试发布应用软件	R	R		C
在工作站上安装应用	A		P	C

说明：A = (Approve) 批准；R = (Reviews) 评审；P = (Participant) 参加；C = (Creator) 建立。

在安排人力资源的时候一定要合理，不能少也不可以过多，否则就会出现反作用，即要控制项目组的规模。人数多了，进行沟通的渠道就多了，管理的复杂度就高了，对项目经理的要求也就高了。在微软的 MSF (Microsoft Solution Framework) 中，有一个很明确的原则就是要控制项目组的人数不要超过 10 人，当然这不是绝对的，也和项目经理的水平有很大关系。但是人员“贵精而不贵多”，这是一个基本的原则。

首先在软件开发的一开始，要合理的配置人员，根据项目的工作量、所需要的专业技能，再参考各个人员的能力、性格、经验，组织一个高效、和谐的开发小组。一般来说，一个开发小组人数在 5 到 10 人之间最为合适，如果项目规模很大，可以采取层级式结构，配置若干个这样的开发小组。

在选择人员的问题上，要结合实际情况来决定是否选入一个开发组员。并不是一群高水平的程序员在一起就一定可以组成一个成功的小组。作为考察标准，技术水平、与本项目相关的技能和开发经验、以及团队工作能力都是很重要的因素。一个编程高手却不能与同事沟通融洽的程序员，未必适合一个对组员之间沟通要求很高的项目。还应该考虑分工的需要，合理配置各个专项的人员比例。例如一个网站开发项目，小组中有页面美工、后台服务程序、数据采集整理、数据库几个部分，应该合理的组织各项工作的人员配比。

在人员组建方面，需要对项目组人员进行规划配置，合理分工，明确责任，保证项目各阶段、各方面的工作能够按计划完成。

例如，某项目经理在一个项目中配置了以下人员：技术组长一名，负责技术难题攻关，组间沟通协调；需求人员 3 名，负责将用户需求转换成项目内的功能需求和非功能需求，编制项目需求规格说明书，针对每个集成版本与用户交流获取需求的细化；设计人员 2 名，负责根据需求规格说明书，进行系统设计；开发人员 5 名，实现设计，完成用户功能；集成人员 1 名，负责整套系统的编译集成，督促小组系统功能提交，及时发现各模块集成问题，起到各小组之间的沟通的纽带；测试人员 2 名，对于集成人员集成的版本进行测试，尽可能的发现程序缺陷，以及未满足需求的设计；文档整理人员 1 名，负责对小组内产生文档的整合、统一。

人员管理计划是人力资源规划的一个输出，它描述了项目团队的人员什么时候如何加入到团队中和离开团队。作为项目计划一部分，详细程度因项目而异。图 6-7 描述了一个软件项目在一年中的不同人员的分布情况，例如 Java 程序员、系统分析员、技术文档编写者、数据库分析员等在每个月中的分布数量。

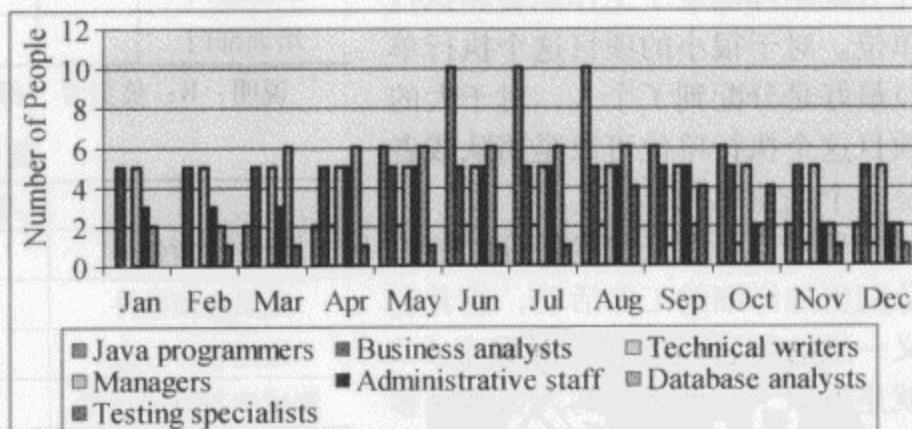


图 6-7 人员分布图

## 6.5 校务通系统案例分析

### 1) 校务通系统的组织结构。

如图 1 所示，它是矩阵型组织结构的一个具体化。

其中：

市场部

- 负责与用户的协调工作
- 负责项目相关的商务活动
- 负责用户需求的接口
- 配合项目经理的资源协调活动
- 负责产品的验收活动
- 负责系统的维护活动

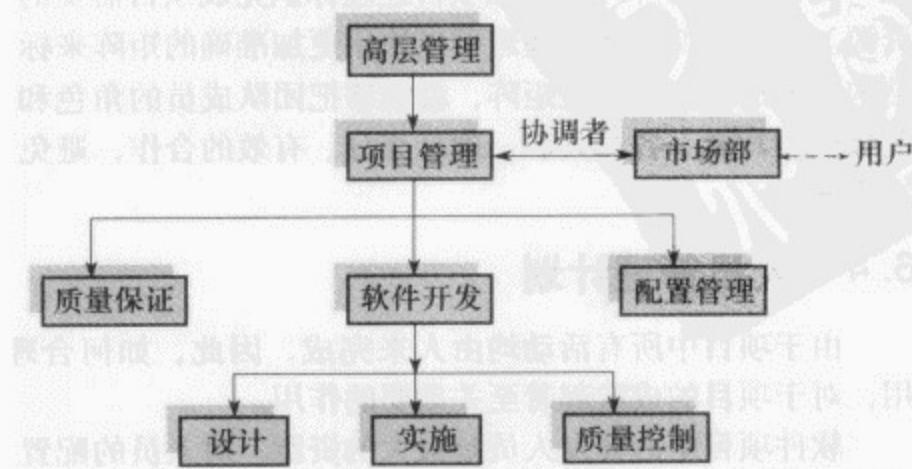


图 1 项目的组织结构

**项目管理**

- 负责项目的组织和规划
- 负责项目计划制定和维护
- 负责项目的跟踪和管理
- 负责资源的分配和协调活动
- 负责各组织和计划之间的协调活动
- 负责与市场部的协调活动

**软件开发**

- 负责项目的软件开发，包括设计，编码，单元测试和集成测试
- 负责产品质量控制的工作
- 负责配合质量保证的活动，如系统测试，文档编制等
- 配合产品验收的相关活动

**质量保证**

- 负责项目过程和产品规范的制定
- 负责项目过程的质量保证活动，
- 过程评审
- 产品审计

**配置管理**

- 负责项目的配置管理活动
- 负责软件产品的提交

**用户**

- 确保相关责任的实施
- 参与项目的组织和规划
- 负责产品的验收工作

2) 校务通系统项目的责任矩阵 (RAM) 如表 1 所示。

**表 1 角色映射表**

	项目管理组	软件开发组	质量保证组	配置管理组
负责人	韩万江	姜岳尊	郭天齐	韩万江
组成人	韩万江	孙权, 姜岳尊, 郭天齐, 岳好	郭天齐, 岳好	韩万江, 岳好

校务通系统具体的人员计划可以通过进度计划中人力资源分配体现出来的，见进度计划。

## 6.6 小结

人力资源管理是项目成功与否的基础，而项目组织形式是团队的基础。项目组织一般有三种典型形式：职能型，项目型和矩阵型。可以根据项目的特点选择合适的项目组织结构。同时还需要确定项目成员的责任，责任分配矩阵是用来对项目团队成员进行分工，明确其角色与职责的有效工具。作为人力资源规划的另外一个输出是人员管理计划，人员管理计划描述了项目团队的人员何时加入到团队中和何时离开团队。其作为项目计划一部分，详细程度因项目而异。

## 6.7 练习题

### 一、选择题

1. 在哪种组织结构中，项目成员没有安全感。（ ）  
A. 职能型      B. 矩阵型      C. 项目型      D. 弱矩阵型
2. 项目经理在一个高科技公司，现在正在为一个新的项目选择合适的组织结构，这个项目涉及很多的领域和特性，他应该选择哪种组织结构？（ ）  
A. 矩阵型      B. 项目型      C. 职能型      D. 组织型

3. 责任分配矩阵是（ ）。  
A. 进行人力资源计划的一个工具      B. 一种组织结构  
C. 与 WBS 类似                          D. 估算成本的方法
4. 人力资源分配图是（ ）。  
A. 展现目前的人力资源质量      B. 展现项目中人力在各个阶段的分布情况  
C. 说明人员分工情况                    D. 说明项目需要的所有资源
5. 人员管理计划描述了（ ）。  
A. 如何获取项目成员  
B. 如何解决冲突  
C. 项目经理的团队建设总结  
D. 项目团队的人员什么时候，加入到团队中和什么时候离开团队

## 二、判断题

1. 组织分解结构是一种特殊的 WBS。（ ）
2. 责任分配矩阵是明确项目团队成员的角色与职责的有效工具。（ ）
3. 职能型组织结构中项目经理的权利最大。（ ）
4. 为了创建一个组织结构图，项目管理者首先明确项目需要的人员类型。（ ）

## ■ 项目沟通计划

及时成功地完成软件项目，并能达到或者超过预先期望的结果，是很不容易的。为了完成所下达的任务，既要有统一的项目计划，还要有一套适合的监控执行方法，但同时又不能扼杀了整个项目开发过程中的创造性和自主性。这样，就必须有一个灵活而且容易使用的沟通方法，从而使一些重要的项目信息实时、最新，做到实时同步，为此项目管理过程中需要沟通计划。本章进入本书路线图的沟通计划，如图 7-1 所示。

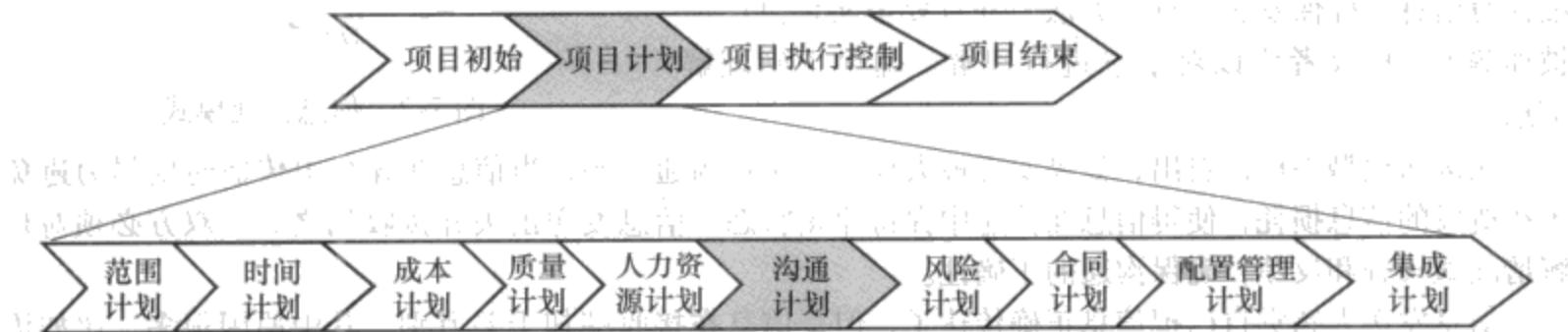


图 7-1 路线图——项目计划 - 沟通计划

### 7.1 项目沟通管理概述

沟通是一个过程，在这个过程中，信息通过一定的符号、标志或者行为系统在人员之间交换，人们之间可以通过身体的直接接触、口头或者符号的描述等方式沟通。统计表明，项目经理 80% 以上的时间做沟通管理。

沟通管理是对传递项目信息的内容、传递项目信息的方法、传递项目信息的过程等几个方面的综合管理，是确定项目干系人的信息交流和沟通需要，确定谁需要信息，需要什么信息，何时需要信息，以及如何将信息分发给他们。沟通管理的基本原则是及时性、准确性、完整性、可理解性。

沟通管理可以确保按时和准确产生项目信息、收集项目信息、发布项目信息、存储项目信息、部署项目信息、处理项目信息所需的过程。项目沟通管理为成功所必须的因素——人（people）、想法（ideas）和信息（information）之间提供了一个关键连接。涉及项目的任何人都应准备以项目“语言”（the project language）发送和接收信息，并且必须理解他们以个人身份参与的沟通怎样影响整个项目。

在软件开发项目中，对于涉及到项目进度和人力资源调度等问题而言，充分的沟通，是一个非常重要的管理手段。尽管项目评估能够在一定程度上解决一些问题，但需要注意的是，如果在计划制定及实行过程中缺乏沟通，不但会从进度上影响项目的进行，也会对项目团队人员的积极性产生不良影响，令供需双方产生彼此的不信任感，从而严重干扰项目开发进度。

在 IT 项目中，许多专家都认为：对于成功，威胁最大的就是沟通的失败。2001 年 Standish Group 研究表明 IT 项目成功的 4 个主要因素分别为：管理层的大力支持，用户的积极参与，有经验的项目管理者，明确的需求表达。而这 4 个要素全部依赖于良好的沟通技巧，特别是技术人员。传统的教育体制注重培养学生的技术技能，而不重视培养他们的沟通与社交技能，很少有沟通（听、说、写）、心理学、社会学和人文科学等软技能方面的课程。

但是在软件项目开发过程中需要进行大量沟通，例如，要开发满足用户需要的软件，必须首先要清楚用户的需求；同时也必须让用户明白你将如何实现这些需求；还要让用户知道为什么有些需求不

能实现，而有些方面可以做得更好；更重要的是要让用户非常愿意地使用你提交的软件，就必须让用户了解它、熟悉它、喜欢它，这些都要充分发挥你听、说、写的本领。

## 7.2 沟通方式

沟通管理的目标是及时并适当地创建、收集、发送、储存和处理项目的信息。沟通是应该占据项目组成员很多时间的工作，他们与客户沟通，与销售人员、开发人员、测试人员等沟通，还需要项目组内部的信息交换。获得的信息量越大，项目现状就越透明，对后续工作的把握就越大。

沟通是一种人与人之间的信息交流活动，所采用的方式应该是双向都可以理解的通用符号和技巧，这样可以保证信息的传送与接收畅通。一般沟通模型至少包括信息发送者、信息、信息接收者，如图 7-2 所示。发送者应该仔细核对信息，确保发送信息的方法，并且要证实信息已经被理解了。接收者应该对信息进行理解，确保正确理解信息。

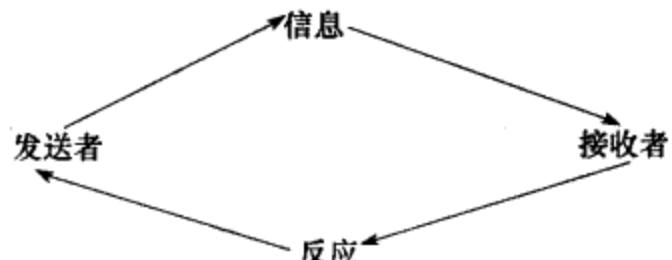


图 7-2 信息沟通模式

从沟通模型中可以看出，如果要想最大程度地保障沟通顺畅，当信息在媒介中传播时要尽力避免干扰造成的信息损耗，使得信息在传递中保持原始状态。信息发送出去并接收到之后，双方必须对理解情况做检查和反馈，确保沟通的正确性。

信息发送人感觉自己把信息正确传达了，但是信息在接收端却千差万别。其中原因很多，比如语言、文化、语义、知识、信息内容、道德规范、名誉、权利、组织状态等，在项目执行中经常碰到由于背景不同而在沟通中产生理解差异的情况。保证沟通渠道畅通是前提，传递有效信息是关键。很多人以为沟通就是多说说话，结果是说到令人生厌也没完，漫无目的的沟通是无效的沟通。

沟通前，项目经理要弄清楚沟通的真正目的是什么？要对方理解什么？确定了沟通目标，沟通的内容就围绕沟通要达到的目标组织规划，根据不同的目的选择不同的沟通方式。

沟通方式主要有书面沟通和口头沟通、语言沟通和非语言沟通、正式沟通和非正式沟通、单向沟通和双向沟通、网络沟通等。将项目管理的信息正确传达到相应的人员，是相当重要并有一定的困难的，经常发生的事情是信息发送人感到自己把信息正确传达了，但实际的结果却是信息没有传达到或者是被错误的理解了。太多的人还是不太习惯成堆的文件或者通篇的 EMAIL 传送的邮件，如果能利用非正式的方式或者是双方会谈的方式来听取重要的信息，就会来得又快又准确更能让人接受，就像传统里的一纸书信在某些场合还是比任何一种现代化的联系方式来得好一样，价值取向不同，沟通的方式也就在使用效果上全然不一样了。德鲁克提出的四个基本法则：沟通是一种感知，沟通是一种期望，沟通产生要求，信息不是沟通。

对于紧急的信息应该通过口头的方式沟通，对于重要的信息采用书面的方式沟通。项目人员应该了解以下内容：

- 许多非技术专业人员，从同事到领导，更愿意以非正式的形式和双向的会谈来听取重要的项目信息。
- 有效地发送信息依赖于项目经理和项目组成员良好的沟通技能。口头沟通还有助于在项目人员和项目干系人之间建立较强的联系。
- 人们还有不愿报告坏消息的倾向，报喜不报忧的状况要引起注意。
- 对重大的事件、与项目变更有关的事件、有关项目和项目成员利益的承诺等要采用正式方式发送和接受。
- 与合同有关的信息要以正式方式发送和接受。

## 7.3 沟通渠道

沟通渠道就像连接每个人的电话线一样，随着项目的进展和范围的增加，项目团队成员的数量也

在不断的增加，人越多沟通的渠道就越多，我们知道3个人有3条沟通渠道，5个人有10条沟通渠道，如图7-3所示。

沟通渠道的公式为  $N(N - 1)/2$ ，其中N为人员总数。可见如果团队中的人越多，存在的沟通渠道就越多，管理者的管理难度也加大了。

为保证沟通的良好效果，必须保持沟通渠道的畅通和单一。例如，作为客户项目经理，应该是唯一的客户接口，所有针对客户的信息也只能通过客户项目经理来传递，所有跟客户相关的会议，客户项目经理必须在场，这样才能保证客户需求和客户信息的一致性。

## 7.4 项目沟通计划

保证项目成功必须进行沟通，为了有效的沟通，需要创建一个沟通计划。沟通计划决定项目相关人的信息和沟通需求：谁需要什么信息、什么时候需要、怎样获得、选择的沟通模式——什么时候采用书面沟通和什么时候采用口头沟通、什么时候使用非正式的备忘录和什么时候使用正式的报告等等。沟通计划常常与组织计划紧密联系在一起，因为项目的组织结构对项目沟通要求有重大影响。

在项目初始阶段就应该包含沟通计划，根据对项目相关人员的分析，项目经理和项目团队的成员可以确定沟通的需求。提供人们不需要的信息是没有用处的，花费在编制和发送无用信息的时间是一种资源的浪费。沟通计划可以将沟通的过程、沟通的类型和沟通的需求进行组织和文档化。

### 7.4.1 沟通计划

项目沟通计划是对项目全过程的沟通内容、沟通方法、沟通渠道等各个方面的计划与安排。就大多数项目而言，沟通计划的内容是作为项目初期阶段工作的一个部分。由于项目相关人员有不同的沟通需求，所以应该在项目的早期，与项目相关人员一同确定沟通管理计划，并且评审这个计划，可以预防和减少项目进行过程中存在的沟通问题。同时，项目沟通计划还需要根据计划实施的结果进行定期检查，必要时还需要加以修订，所以项目沟通计划管理工作是贯穿于项目全过程的一项工作。尤其，企业在同时进行多个项目的时候，制定统一的沟通计划和沟通方式，有利于项目的顺利进行。例如，公司所有的项目有统一的报告格式，有统一的技术文档格式，有统一的问题解决渠道，起码给用户的感觉是公司的管理是有序的。编制沟通计划的具体步骤如下。

#### 1. 准备工作

##### (1) 收集信息

收集沟通过程中的信息，包括：

- 项目沟通内容方面的信息。
- 项目沟通所需沟通手段的信息。
- 项目沟通的时间和频率方面的信息。
- 项目信息来源与最终用户方面的信息。

##### (2) 加工处理沟通信息

对收集到的沟通计划方面的信息进行加工和处理也是编制项目沟通计划的重要一环，而且只有经过加工处理过的信息，才能作为编制项目沟通计划的有效信息使用。

#### 2. 确定项目沟通需求

项目沟通需求的确定是在信息收集的基础上，对项目组织的信息需求做出的全面决策，其内容包括：

- 项目组织管理方面的信息需求。

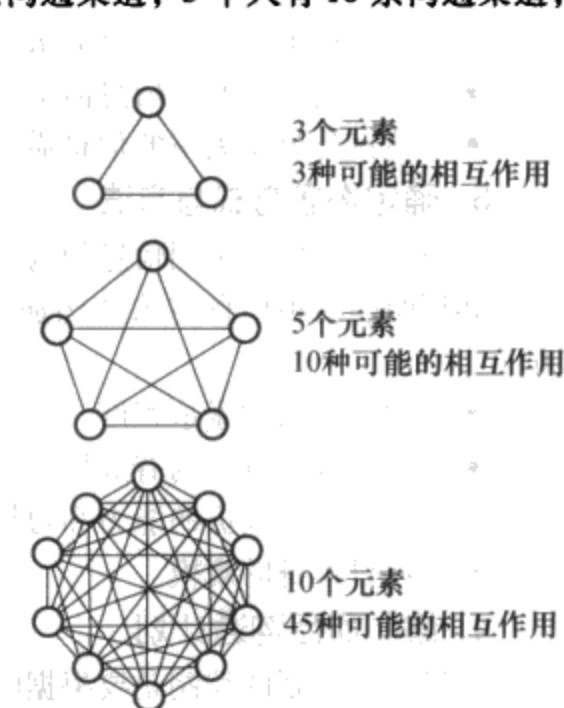


图7-3 沟通渠道

- 项目内部管理方面的信息需求。
- 项目技术方面的信息需求。
- 项目实施方面的信息需求。
- 项目与公众关系的信息需求。

### 3. 确定沟通方式与方法

在项目沟通中，不同信息的沟通需要采取不同的沟通方式和方法，因此在编制项目沟通计划过程中还必须明确各种信息需求的沟通方式和方法。影响项目选择沟通方式方法的因素主要有以下几个方面：

- 沟通需求的紧迫程度。
- 沟通方式方法的有效性。
- 项目相关人员的能力和习惯。
- 项目本身的规模。

### 4. 编制项目沟通计划

项目沟通计划的编制是要根据收集的信息，先确定出项目沟通要实现的目标，然后是根据项目沟通目标和沟通需求确定沟通任务，进一步根据项目沟通的时间要求去安排这些项目沟通任务，并确定出保障项目沟通计划实施的资源和预算。

制定一个协调的沟通计划非常重要，清楚地了解什么样的项目信息要报告、什么时候报告、如何报告、谁来负责编写这些报告非常重要。项目经理要让项目组人员和项目干系人都了解沟通管理计划，对各自负责的部分要让他们根据相关规范来编制。

在制定项目计划时，可以根据需要制定一个沟通计划，沟通计划可以是正式的或者非正式的，可以是详细的或提纲式的。沟通计划没有固定的表达方式，它是整个项目计划的一个部分。沟通管理计划内容主要包括：

1) 沟通需求，分析项目相关人需要什么信息，确定谁需要信息，何时需要信息。对项目干系人的分析，有助于确定项目中各种参与人员的沟通需求。

2) 沟通内容，确定沟通内容包括沟通的格式、内容、详细程度等。如果可能的话可以统一项目文件格式，统一各种文件模板，并提供编写指南。

3) 沟通方法，确定沟通方式、沟通渠道等，保证项目人员能够及时获取所需的项目信息。信息如何收集、如何组织，详细描述沟通类型，采用的沟通方式，沟通技术等。例如检索信息的方法，明确信息保存方式、信息读写的权限、会议记录、工作报告、项目文档（需求、设计、编码、发布程序等）、辅助文档等的存放位置，及相应的读写权利，约束条件与假设前提等。明确表达项目组成员对项目经理或项目经理对上级和相干人员的工作汇报关系和汇报方式，明确汇报时间和汇报形式。例如，项目组成员对项目经理通过 EMAIL 发送周报；项目经理对直接客户和上级按月通过 EMAIL 发月报的方式；紧急汇报通过电话及时沟通；每两周项目组进行一次当前工作沟通会议；每周同客户和上级进行一次口头汇报等等。包括问题的解决程序、途径等。

4) 沟通职责，谁发送信息，谁接收信息，制定一个收集、组织、存储和分发适当信息给适当人的系统。这个系统也包括对发布的错误信息进行修改和更正。详细描述项目内信息的流动图。这个沟通结构描述了沟通信息的来源、信息发送的对象以及信息的接收形式。传送重要项目信息的格式、权限。

5) 沟通时间安排，创建沟通信息的日程表，类似项目进展会议的沟通，应该定期进行，设置沟通的频率等。其他类型的沟通可以根据项目的具体条件进行。

6) 沟通计划维护，当项目进展时，沟通计划如何修订的指南，明确本计划在发行变化时，由谁进行修订，并对相关人员发送。

其实，沟通计划也包括很多其他的方面，例如，应该有一个专用于项目管理中所有相关人员的联系方式的小册子，其中包括项目组成员、项目组上级领导、行政部人员、技术支持人员、出差订房订票等人员的相关联系信息等。联系方式做到简洁明了，最好能有对特殊人员的一些细小标注。

沟通规划在项目规划的早期进行并且贯穿项目生存期，项目干系人会发生变化，他们的需求可能也会发生变化，沟通计划也需要定期的审核和更新。

#### 7.4.2 沟通建议

在项目中一些重要信息没有进行充分和有效的沟通，会造成各做各事、重复劳动，甚至造成不必要的损失。例如沟通过程中的沉默、没有反应等可能表示项目人员还尚未弄清楚问题，为此项目管理者还需要同开发人员进行充分沟通，了解开发人员的想法。在对项目没有一个共同的一致的理解的前提下，一个团队是不可能成功的。为此，需要制定有效的沟通制度和沟通机制，对由于缺乏沟通而造成的事件进行通报作为教训提醒，以提高沟通意识，提高大家对沟通作用的认识；沟通方式应根据内容而多样化，讲究有效率的沟通。例如通过制度规定对由于未及时收取邮件而造成损失的责任归属；对于特别重要的内容要采用多种方式进行有效沟通以确保传达到位，除发送邮件外还要电话提醒、回执等，重要的内容还要通过举行各种会议进行传达。

为解决沟通中的问题，建议采取多种、灵活、经济的沟通方式，例如，对一般的小问题或者是简单问题进行电话交流，复杂的、必要的、重要的沟通需要以会议解决，形成书面的会议记要。在保证效果的前提下节省时间，提高工作效率。规定项目组成员在每天工作过程中遇到问题，将其记录下来，然后在以邮件方式发送给需要沟通或者询问者，对于可以直接回答的问题则直接以邮件方式回复，对于无法直接答复而只需与提出问题者讨论的问题，进行商议确定，而需要众人一起讨论的问题，则召开会议讨论，较紧急的问题召开临时性会议。通过以上方法，可以及时发现问题、解决问题，避免了因各方立场不一致造成严重对立而影响项目进度，避免了因交流不畅形成重大质量问题。

### 7.5 校务通系统案例分析

为了保证项目开发过程的顺利进行和信息的有效沟通，校务通系统项目要求如下的沟通计划。

- 1) 项目进行过程中，项目开发人员需要及时了解项目中的各种技术信息，以及相关的管理信息；项目经理需要全面了解项目所有信息；高层领导需要了解项目管理中的进度信息、成本信息、质量信息；客户需要了解项目管理中的进展和提交情况。
- 2) 项目信息实时查询：通过给定的权限实时查询配置管理系统的项目信息，由配置管理员进行项目资料的实时更新和发布。
- 3) 项目组成员之间的及时沟通，项目进行过程中，在开发环境方面保证开发人员集中在一起开发，便于有问题及时交流沟通。同时，确定项目组的沟通和汇报渠道。
- 4) 项目交流会议。
  - 每天 17:00 ~ 17:30 项目组成员进行口头交流。
  - 每周五的 14:00 前提交周报告，格式见模板。
  - 每周五的 15:00 ~ 17:00 召开项目周例会，会后发布会议纪要给相关的项目人员，其中说明项目的进展和存在的问题。
- 5) 及时提交问题报告，问题可以通过网络提交，项目经理会及时获取问题信息。
- 6) 项目报告：每周向项目组及其公司领导层发布项目评审报告，每周向客户发布项目简报和问题反馈结果。

### 7.6 小结

沟通是项目成功的前提，确定谁需要信息，需要什么信息，何时需要信息，需要什么信息，以及如何将信息分发给他们。沟通是项目经理在项目管理中最多的任务，沟通可以根据具体情况选择适当的沟通形式，以保证沟通的有效性，必要时，可以建立沟通计划，保证团队内的沟通渠道的畅通。

## 7.7 练习题

### 一、选择题

1. 项目经理花在沟通上的时间是（ ）。  
A. 20% ~ 40%      B. 75% ~ 90%      C. 60%      D. 30% ~ 60%
2. 大量使用（ ）沟通最有可能协助解决复杂的问题。  
A. 口头      B. 书面      C. 正式      D. 信息管理系统
3. 对于项目中比较重要的通知，最好采用（ ）沟通方式。  
A. 口头      B. 网络方式      C. 书面      D. 电话
4. 项目团队原来有6个成员，现在又增加了6个成员，这样沟通渠道增加了多少？（ ）  
A. 4.4倍      B. 6倍      C. 2倍      D. 6条
5. 在项目进行过程中，老板突然有个紧急的通知告知项目经理，要求项目经理告诉团队成员，这时项目经理应该采取（ ）沟通方式。  
A. 口头      B. 书面      C. 正式      D. 检索
6. 下面各项对沟通计划的描述，哪项是错误的？（ ）  
A. 确定沟通需求      B. 确定沟通的内容      C. 确定沟通方法      D. 对项目管理是没有必要的

### 二、判断题

1. 在项目管理过程中，沟通是项目管理者的一部分工作，而且是很少的一部分工作。（ ）
2. 沟通计划是项目计划的一部分。（ ）
3. 沟通管理的基本原则是及时性、准确性、完整性、可理解性。（ ）
4. 在IT项目中，许多专家都认为：对于成功，威胁最大的就是沟通的失败。（ ）

## ■ 软件项目风险计划

任何项目都有一定的不确定性，所以，项目具有一定的风险，如果没有很好的风险管理，项目就可能遇到麻烦。所以，软件项目管理过程中，风险计划也是一个重要的计划，只有进行合理的风险管理，制定及时的风险计划，才能防患于未然，做到主动控制风险，而不是被动地被风险所控制。本章进入本书路线图的风险计划，如图 8-1 所示。

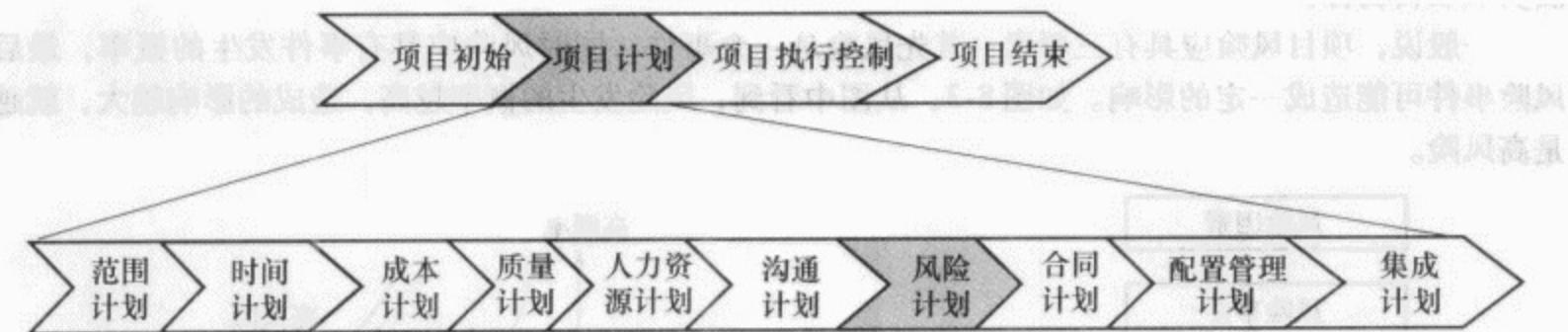


图 8-1 路线图——项目计划 - 风险计划

### 8.1 风险概念

项目中的风险有很多种，没有风险的项目几乎是不存在的，只是风险的多少、严重程度大小不同而已。在软件项目管理当中，很多项目管理者，乃至经验丰富的项目管理者经常会更关注于项目范围、进度、成本，并且认为完成这三项就能完成项目管理的目的。但是，在项目管理过程中，还需要风险管理，没有完善的风险管理机制，项目将困难重重。

软件项目风险管理是一种特殊的规划方式，当对软件项目有较高的期望值时，一般都要进行风险分析。进行过大中型项目开发的人都亲身体验到许多事情可能出错，成功的项目需要采取积极的步骤对要发生或即将发生的风险进行管理。

项目风险管理过程中，存在多种风险管理方法与工具，软件项目管理只有找出最适合自己的方法与工具并应用到风险管理中，才能尽量减少软件项目风险，促进项目的成功。

#### 8.1.1 风险定义

风险是损失发生的不确定性，是对潜在的，未来可能发生损害的一种度量，如果它确实发生了，则它的发生会对项目产生有害的或者负面影响。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。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。

软件项目风险会影响项目计划的实现，如果项目风险变成现实，就有可能影响项目的进度，增加项目的成本，甚至使软件项目不能实现。如果对项目进行风险管理，就可以最大限度的减少风险的发生。但是，目前国内的软件企业不太关心软件项目的风险管理，结果造成软件项目经常性的延期、超过预算，甚至失败。

软件项目的开发过程中，必然要使用一些新技术、新产品，同时由于软件系统本身的结构和技术复杂性的原因，需要投入大量人力、物力和财力。这就造成开发过程中存在某些“未知量”或者说“不确定因素”，也必然给项目的开发带来一定程度的风险，也就可能会使项目计划失败或不能完全达

到预期目标。因此，对项目风险进行科学、准确的判别，为项目决策层和管理人员提供科学的评估方法是十分必要的。

一个项目的损失可能有不同的后果形式，例如软件质量的下降，成本费用的超出，项目进度的推迟等等。

风险发生的过程如图 8-2 所示，首先有风险因素的存在，风险因素导致风险事件的发生，从而造成损失，而损失又引起了实际和计划之间的差异，从而得到风险的结果。

风险因素是指能够引起或者增加风险事件发生的机会或影响损失严重程度的因素，是造成损失的直接或者间接的原因。例如，需求的变化、设计错误、疏漏和理解错误、狭隘定义或理解错误、不充分估计、不胜任的技术人员等等都是风险因素。

风险事件是指那些特殊的、不确定的、人们不愿意它发生的或者没有规划的事件。它可能导致无法实现项目目标。

一般说，项目风险应具有三要素：首先风险是一个事件，同时风险应具有事件发生的概率，最后风险事件可能造成一定的影响。如图 8-3，从图中看到，风险发生的概率越高，造成的影响越大，就越是高风险。

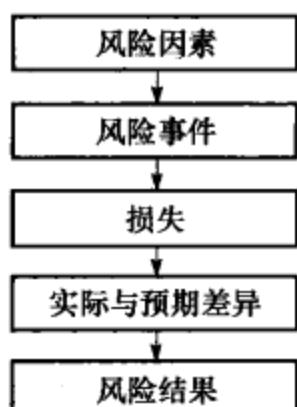


图 8-2 风险发生过程

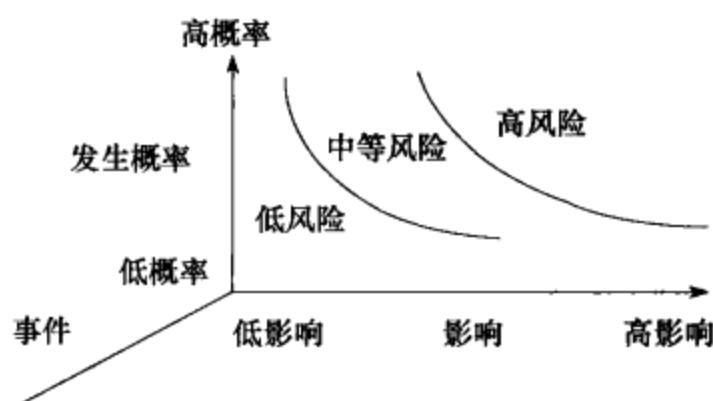


图 8-3 风险图示

风险是一种概率事件——它可能发生也可能不发生。因此，我们通常会表现出很乐观，不是看不到风险就是希望它们不会发生。如果风险真的出现了，这种态度会使项目陷入困境，这是一个大型项目中很可能发生的事情。因此，风险管理被认为是管理大型软件项目的最佳实践。

风险管理旨在识别出风险，然后采取措施使它们对项目的影响最小。风险管理是软件管理中相对较新的领域，它首次出现于勃姆（Boehm）关于风险管理的指南中。自那以后，软件的风险管理逐渐被人们所认识。

当没有办法消除风险，甚至连试图降低该风险也存在疑问时，这些风险就是真正的风险了。

### 8.1.2 风险类型

从范围角度上看，风险主要分：商业风险、管理风险、人员风险、技术风险、开发环境风险、客户风险、过程风险、产品规模风险等等。

1) 商业风险。商业风险是指与管理或市场所加诸的约束相关的风险，主要包括市场风险、策略风险、管理风险和预算风险等。例如：如果开发的软件不是市场真正所想要的，就发生了市场风险；如果开发的软件不再符合公司的软件产品策略，就发生了策略风险；由于重点转移或者人员变动而失去上级管理部门的支持，就发生了管理风险；如果没有得到预算或者人员的保证，就发生了预算风险。

2) 管理风险。管理风险是指潜在的预算、进度、个人（包括人员和组织）、资源、用户和需求方面的问题，例如时间和资源分配的不合理、项目计划质量的不足、项目管理原理使用不良所导致的风险、资金不足、缺乏必要的项目优先级等。项目的复杂性、规模的不确定性和结构的不确定性也是构成管理风险的因素。管理风险的因素还包括项目的复杂性、规模、结构的不确定性。

3) 人员风险。人员风险是与参与项目的软件人员稳定性、总体技术水平及项目经验相关的风险。

作为先决条件的任务不能按时完成；开发人员和管理层之间关系不佳，导致决策缓慢，影响全局；缺乏激励措施，士气低下，降低了生产能力；某些人员需要更多的时间适应还不熟悉的软件工具和环境；项目后期加入新的开发人员，需进行培训并逐渐与现有成员沟通，从而使现有成员的工作效率降低；由于项目组成员之间发生冲突，导致沟通不畅、设计欠佳、接口出现错误和额外的重复工作；不适应工作的成员没有调离项目组，影响了项目组其他成员的积极性；没有找到项目急需的具有特定技能的人等。

4) 技术风险。技术风险是与待开发软件的复杂性以及系统所包含技术的“新奇性”相关的风险，例如，潜在的设计、实现、接口、检验和维护方面的问题。规格说明的多义性、技术上的不确定性、技术陈旧、以及“过于先进”的技术等都是技术风险因素。复杂的技术，项目执行过程中使用技术或者行业标准发生变化所导致的风险也是技术风险。技术风险威胁要开发的软件质量及交付时间。如果技术风险变成现实，则开发工作可能变得很困难或者不可能。

5) 开发环境风险。开发环境风险是与用以开发产品的工具的可用性及质量相关的风险。设施未及时到位；设施虽到位，但不配套；开发工具未及时到位；开发工具不如期望的那样有效，开发人员需要时间创建工作环境或者切换新的工具；新的开发工具的学习期比预期的长，内容繁多。

6) 客户风险。客户风险是与客户的素质以及开发者和客户定期沟通的能力相关的风险。客户对于最后交付的产品不满意，要求重新设计和重做；客户的意见未被采纳，造成产品最终无法满足用户要求，因而必须重做；客户对规划、原型和规格的审核决策周期比预期的要长；客户没有或不能参与规划、原型和规格阶段的审核，导致需求不稳定和产品生产周期的变更；客户答复的时间（如回答或澄清与需求相关问题的时间）比预期长；客户提供的组件质量欠佳，导致额外的测试、设计和集成工作，以及额外的客户关系管理工作。

7) 产品风险。产品风险是与质量低下的不可接受的产品相关的风险，矫正这些产品的质量，需要比预期更多的测试、设计和实现工作；开发额外的不需要的功能（镀金），延长了计划进度；严格要求与现有系统兼容，需要进行比预期更多的测试、设计和实现工作；要求与其他系统或不受本项目组控制的系统相连，导致无法预料的设计、实现和测试工作；在不熟悉或未经检验的软件和硬件环境中运行所产生的未预料到的问题；开发一种全新的模块将比预期花费更长的时间；依赖正在开发中的技术将延长计划进度。

8) 过程风险。过程风险是与软件过程被定义的程度以及它们被开发组织所遵守的程度相关的风险。大量的纸面工作导致进程比预期的慢；前期的质量保证行为不真实，导致后期的重复工作；太不正规（缺乏对软件开发策略和标准的遵循），导致沟通不足，质量欠佳，甚至需重新开发；过于正规（教条地坚持软件开发策略和标准），导致过多耗时于无用的工作；向管理层撰写进程报告占用开发人员的时间比预期的多；风险管理粗心，导致未能发现重大的项目风险。

从预测角度看，风险可分为下面三种类型：

1) 已知风险。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、以及其他可靠的信息来源（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。可预测风险是指能够从过去项目的经验中推测出来的风险（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等）。

3) 不可预测风险。不可预测风险是可能、也会真的出现，但很难事先识别出来的风险。

项目管理者只能对已知和可预测风险进行规划，不可预测的风险，只能靠企业的能力来承担了。

### 8.1.3 风险的基本性质

项目风险具有如下的基本性质：

1) 客观性。风险客观性首先表现在它的存在是不以人的意志为转移的，因为决定风险的各种因素对风险主体是独立存在的，不管风险主体是否意识到风险的存在，在一定的条件下就可能变为现实，其次，风险客观性还表现在风险无时不有，无所不在，它潜在各种活动之中。

2) 不确定性。风险发生的不确定性,表现为风险的程度多大,风险何时何地有可能变为现实是不肯定的,由于人们对客观世界的认识受到各种条件的限制,不可能准确的预测风险的发生,不确定性要求我们运用各种方法进行测度。

3) 不利性。风险一旦产生时,就会使风险主体产生挫折、失败、甚至损失,对风险主体不利,因此我们在承认风险、认识风险的基础上,做好决策,尽可能避免风险,将风险的不利性降到最低。

4) 可变性。风险的可变性表现在一定条件下风险可以进行转化,风险可以转化为非风险,非风险的事件可以转化为风险事件。例如,项目初始经过评估确认人员流动是风险事件,但是随着项目进展,人员比较稳定了,这样“人员流动”就不是风险事件了。

5) 相对性。风险的相对性是针对风险的主体而言的,在相同的风险情况,不同的风险主体对风险的承受能力不同,不同的组织和个人往往对风险有着不同的容忍限度。例如:一个高利润高收益的公司也许愿意为一个10亿美元的合同花费50万做一个投标书,而一个收支相抵的公司则不会。一个组织也许认为15%的误差机率是高风险的,而其他组织却认为这个机率风险很低。

6) 风险和利益的对称性。风险和利益是同时存在的,风险是利益的代价,利益是风险的报酬,没有利益只有风险,没人会做,实现利益必须承担一定的风险。

## 8.2 风险管理

风险管理是项目管理中最容易被忽略而且是最难以管理的环节,它是在项目进行过程中不断对风险进行识别、评估、制定策略、监控风险的过程。通过风险识别、风险分析和风险评价去认识项目的风险,并以此为基础合理的使用各种风险应对措施、管理方法、技术和手段对项目的风险进行有效的控制,妥善处理风险事件造成的不利后果,以最小的成本保证项目总体目标的实现。

目前,风险管理被认为是软件项目中减少失败的一种重要手段。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。

我们可能无法根除风险,但是我们如果加强对风险产生的认识,对项目产生的风险进行有效的管理,就可以从最大限度上减少风险的发生。了解和掌握项目风险的来源、性质和发生规律,强化风险意识,进行有效的风险管理对项目的成功具有很重要的意义。

一般说,在风险实施前需要进行风险规划,即决定采用什么方式方法、如何计划项目风险的活动,指导对于特定项目如何进行风险管理。例如确定风险管理执行的方法、采用的工具、数据等;风险管理过程中的角色、职责;风险管理的预算、时间等要求;确定风险的分类;如何评估风险的概率、影响等,如何生成风险管理文档。

作为一个优秀的风险管理者应该采取主动的风险管理策略,而不是弥补和解决了多少问题(风险未被及时识别或妥善处理,就会转换成问题),即着力预防和消灭根源的管理策略。而不应该采取被动的方式,被动风险管理是直到风险变成真正的问题时,才会拨出资源来处理它们,更普遍的是,软件项目组对风险不闻不问,直到发生了错误才赶紧采取行动,试图迅速地纠正错误。这种管理模式常常被称为“救火模式”。当补救的努力失败后,项目就处在真正的危机之中了。主动风险管理策略的目标是预防风险。但是,因为不是所有的风险都能够预防,所以,项目组必须建立一个应付意外事件的计划,使其在必要时能够以可控的及有效的方式作出反应。

只有进行很好的风险管理才能有效地控制项目的成本、进度、产品需求,同时可以阻止意外的发生,这样项目经理可以将精力更多地放到项目的及时提交上,不用像救火队员一样,处于被动状态。同时,风险管理可以防止问题的出现,即使出现,也可以降低程度。可以说你不跟踪风险,风险就跟踪你。正如Tom Gilb所说“如果你不主动攻击风险,风险就会主动攻击你。”

风险管理是在风险尚未产生、形成之前,对风险进行识别,并且评估风险出现的概率以及经它们能够产生的影响,按风险从高到低排序,有计划地管理。这种做法正好和被动风险管理方式相反,我们在这里采取了主动出击,就是这样做,我们也并不能从根本上防止未知风险的产生,我们能做的只是减少风险发生的可能,所以说风险的管理和我们的认知以及经验有一点的联系。在计划项目

书中写清如何进行风险管理。在项目的预算中必须包含风险解决所需要的经费，以及它们可能产生的影响。

### 8.3 风险识别

风险识别是试图系统化地确定对项目计划（估算、进度、资源分配）的威胁，识别已知和可预测的风险，只有识别出这些风险，项目管理者才有可能避免这些风险，且当必要时控制这些风险。在项目的前期及早识别出风险是很重要的。风险识别在项目开始时就要进行，并在项目执行过程中不断跟进。

风险识别包括确定风险的来源，确定风险产生的条件，描述风险特征和确定哪些风险事件有可能影响本项目。风险识别相当于确定风险三要素中的风险事件。每一类风险可以分为两种不同的情况：一般性风险和特定性风险。一般性风险对每一个软件项目而言都是一个潜在的威胁。特定性风险只有那些对当前项目的技术、人员及环境非常了解的人才能识别出来。为了识别特定性风险，必须检查项目计划及软件范围说明，从而了解本项目中有什么特性可能会威胁到项目计划。一般性风险和特定性风险都应该被系统化地标识出来。

项目风险识别应凭借对“因”和“果”（将会发生什么和导致什么）的认定来实现，或通过对“果”和“因”（什么样的结果需要予以避免或促使其发生，以及怎样发生）的认定来完成。

风险识别不是一次性行为，而应有规律的贯穿整个项目中。风险识别过程见图 8-4，其中，风险识别的输入可能是项目的 WBS、工作陈述（SOW）、项目相关信息、项目计划假设、历史项目数据、其他项目经验文件、评审报告、公司目标等。风险识别常用方法是建立“风险条目检查表”，利用一组提问来帮助项目风险管理了解在项目和技术方面有哪些风险。此外，还有德尔菲方法、头脑风暴法、情景分析法、问询法（座谈法、专家法）法等。当然，风险识别还有很多其他的方法，例如流程图法、现场观察法、相关部门配合法和环境分析法等。风险识别的输出是风险列表。

#### 8.3.1 德尔菲方法

德尔菲方法又称专家调查法，是获得多数专家一致意见的方法，以便预测未来的开发问题。它起源于 20 世纪 40 年代末期，最初由美国兰德公司首先使用，很快就在世界上盛行起来，目前此法的应用已遍及经济、社会、工程技术等各领域。我们在进行成本估算的时候也用到了这个方法。用德尔菲方法进行项目风险识别的过程，是由项目风险小组选定与该项目有关的领域专家，并与这些适当数量的专家建立直接的函询联系，通过函询收集专家意见，然后加以综合整理，再匿名反馈给各位专家，再次征询意见。这样反复经过四至五轮，逐步使专家的意见趋向一致，作为最后预测和识别的根据。

#### 8.3.2 头脑风暴法

所谓头脑风暴法，就是以专家的创造性思维来获取未来信息的一种直观预测和识别方法。此法是由美国人奥斯本于 1939 年首创的，从 20 世纪 50 年代起就得到了广泛应用。头脑风暴法一般是在一个专家小组内进行，通过专家会议，激发专家的创造性思维来获取未来信息。这就要求主持专家会议的人在会议开始时的发言应能激起专家们的思维“灵感”，促使专家们感到急需和本能地回答会议提出的问题，通过专家之间的信息交流和相互启发，从而诱发专家们产生“思维共振”，以达到互相补充并产生“组合效应”，获取更多的未来信息，使预测和识别的结果更准确。



图 8-4 风险识别过程

### 8.3.3 情景分析法

情景分析法是根据项目发展趋势的多样性，通过对系统内外相关问题的系统分析，设计出多种可能的未来前景，然后用类似于撰写电影剧本的手法，对系统发展态势作出自始至终的情景和画面的描述。当一个项目持续的时间较长时，往往要考虑各种技术、经济和社会因素的影响，对这种项目进行风险预测和识别，就可用情景分析法来预测和识别其关键风险因素及其影响程度。情景分析法对以下情况是特别有用的：提醒决策者注意某种措施或政策可能引起的风险或危机性的后果；建议需要进行监视的风险范围；研究某些关键性因素对未来过程的影响；提醒人们注意某种技术的发展会给人带来哪些风险。情景分析法是一种适用于对可变因素较多的项目进行风险预测和识别的系统技术，它在假定关键影响因素有可能发生的基础上，构造多重情景，提出多种未来的可能结果，以便采取适当措施防患于未然。

### 8.3.4 风险条目检查表

“风险条目检查表”是最常用也是比较简单风险识别方法，它是利用一组提问来帮助管理者了解项目在各个方面有哪些风险。在风险条目检查表中，列出了一些可能的与每一个风险因素有关的提问，使得风险管理者集中来识别常见的、已知的和可预测的风险，如产品规模风险、依赖性风险、需求风险、管理风险及技术风险等。

风险条目检查表一般根据风险要素进行编写，包括项目的环境、管理层的重视度、技术情况以及内部因素（如团队成员的技能或技能缺陷）等。风险条目是项目经验的积累。

风险条目检查表可以以不同的方式组织，通过判定分析或假设分析，给出这些提问的回答，就可以帮助管理或计划人员估算风险的影响。这个风险条目检查表是以往项目经验的总结。当然，检查表的类别和条目可以根据企业或者项目的具体情况，来选择或者自行开发。下面介绍几种检查表。

#### 1. 基于关键域的检查表

一般来说，作为项目经理可以将主要的精力放在以下几个关键域方面来关注风险。其中每一关键域包含很多的风险检查条目，通过对每个条目的回答，可以识别项目可能存在的风险。

##### (1) 产品规模风险检查表

项目风险是直接与产品规模成正比的。下面的风险检查表中的条目标识了与软件规模相关的常见风险：

- 对于估算出的产品规模的信任程度如何？
- 产品规模与以前产品的规模的平均值的偏差百分比是多少？
- 产品创建或使用的数据库大小如何？
- 产品的用户数有多少？
- 产品的需求改变多少？交付之前有多少？交付之后有多少？
- 复用的软件有多少？

##### (2) 商业影响风险检查表

下面的风险检查表中的条目标识了与商业影响相关的常见风险：

- 本产品对公司的收入有何影响？
- 本产品是否得到公司高级管理层的重视？
- 交付期限的合理性如何？
- 将会使用本产品的用户数及本产品是否与用户的需要相符合？
- 本产品必须能与之互操作的其他产品/系统的数目？
- 最终用户的水平如何？
- 政府对本产品开发的约束是什么？
- 延迟交付所造成的成本消耗是多少？
- 产品缺陷所造成的成本消耗是多少？

对于上述产品规模和商业影响的风险检查表中的每一个回答都必须与过去的经验加以比较。如果出现了较大的百分比偏差或者如果数字接近过去很不令人满意的结果，则风险较高。

### (3) 需求相关风险检查表

很多项目在确定需求时都面临着一些不确定性和混乱。如果在项目早期容忍了这些不确定性，并且在项目进展过程当中得不到解决，这些问题就会对项目的成功造成很大威胁。例如：

- 对产品缺少清晰的认识。
- 对产品需求缺少认同。
- 在需求分析中客户参与不够。
- 没有优先需求。
- 不确定的需要。
- 新的市场不断变化需求。
- 缺少有效的需求变化管理过程。
- 对需求的变化缺少相关分析。

如果对于上述问题中的任何一个答案是肯定的，则需要进一步的研究，以评估潜在的风险。

### (4) 客户相关风险检查表

不同的客户有不同的需要，有些人只知道他们需要什么，而有些人知道他们不需要什么。一些客户希望进行详细的讨论，而另外一些客户则满足于模糊的承诺。不同的客户有不同的个性，一些人喜欢享受客户的身份，而另一些人则根本不喜欢客户的身份。一些人会高兴地接受几乎任何交付的产品，并能充分利用一个不好的产品，而另一些人则会对质量差的产品猛烈抨击。一些人会对质量好的产品表示赞赏，而另一些人则不管怎样都抱怨不休。

客户和供应商之间也有各种不同的通信方式。一些人非常熟悉产品及生产厂商；而另一些人则可能素未谋面，仅仅通过信件来往和电话与生产厂商沟通。

一个“不好的”客户可能会对一个软件项目组能否在预算内完成项目产生很大的影响。对于项目管理者而言，“不好的”客户是对项目计划的巨大威胁和实际的风险。下面的风险检查表中的条目标识了与客户特征相关的常见风险：

- 你以前是否曾与这个客户合作过？
- 该客户是否很清楚需要什么？他能否花时间把需求写出来？
- 该客户是否同意花时间召开正式的需求收集会议，以确定项目范围？
- 该客户是否愿意建立与开发者之间的快速通信渠道？
- 该客户是否愿意参加复审工作？
- 该客户是否具有该产品领域的技术素养？
- 该客户是否愿意让你的人来做他们的工作？
- 该客户是否了解软件过程？

如果对于这些问题中的任何一个答案是否定的，则需要进一步的调研，以评估潜在的风险。

### (5) 过程风险检查表

如果软件过程定义得不清楚；如果分析、设计、测试以无序的方式进行；如果质量是每个人都认为很重要的概念，但没有人切实采取行动来保证它，那么这个项目就处在风险之中。

过程问题包括：

- 高级管理层是否有一份已经写好的政策陈述，该陈述中强调了软件开发标准过程的重要性？
- 开发组织是否已经拟定了一份已经成文的、用于本项目开发的软件过程的说明？
- 开发人员是否同意按照文档所写的软件过程进行开发工作，并自愿使用它？
- 该软件过程是否可以用于其他项目？
- 管理者和开发人员是否接受过一系列的软件工程培训？
- 是否为每一个软件开发者和管理者提供了软件工程标准？

- 是否为作为软件过程一部分而定义的所有交付物建立了文档概要及示例?
- 是否定期对需求规约、设计和编码进行正式的技术复审?
- 是否定期对测试过程和测试情况进行复审?
- 是否对每一次正式技术复审的结果建立了文档,其中包括发现的错误及使用的资源?
- 有什么机制来保证按照软件工程标准来指导工作?
- 是否使用配置管理来维护系统/软件需求、设计、编码、测试用例之间的一致性?
- 是否使用一个机制来控制用户需求的变化及其对软件的影响?
- 对于每一个承包出去的子合同,是否有一份文档化的工作说明、一份软件需求规约和一份软件开发计划?
- 是否有一个可遵循的规程,来跟踪及复审子合同承包商的工作?

技术问题包括:

- 是否使用方便易用的规格说明技术来辅助客户与开发者之间的通信?
- 是否使用特定的方法进行软件分析?
- 是否使用特定的方法进行数据和体系结构的设计?
- 是否90%以上的代码都是使用高级语言编写的?
- 是否定义及使用特定的规则进行代码编写?
- 是否使用特定的方法进行测试用例的设计?
- 是否使用配置管理软件工具控制和跟踪软件过程中的变化活动?
- 是否使用工具来创造软件原型?
- 是否使用软件工具来支持测试过程?
- 是否使用软件工具来支持文档的生成和管理?
- 是否收集所有软件项目的质量度量值?
- 是否收集所有软件项目的生产率度量值?

如果对于上述问题的答案多数是否定的,则软件过程是薄弱的,且风险很高。

#### (6) 技术风险检查表

采用新技术是具有挑战性和令人兴奋的,但这也是有风险的。下面的风险检查表中的条目标识了与建造的技术相关的常见风险:

- 该技术对于你的公司而言是新的吗?
- 客户的需求是否需要创建新的算法?
- 待开发的软件是否需要使用新的或未经证实的硬件接口?
- 待开发的软件是否需要与开发商提供的未经证实的软件产品接口?
- 待开发的软件是否需要与功能和性能均未在本领域得到证实的数据库系统接口?
- 产品的需求是否要求采用特定的用户界面?
- 产品的需求中是否要求开发某些程序构件,这些构件与你的公司以前开发的构件完全不同?
- 需求中是否要求采用新的分析、设计、测试方法?
- 需求中是否要求使用非传统的软件开发方法?
- 需求中是否有过分的对产品的性能约束?
- 客户能确定所要求的功能是可行的吗?

如果对于这些问题中的任何一个答案是肯定的,则需要进一步的调研,以评估潜在的风险。

#### (7) 开发环境风险检查表

软件工程环境支持项目组、过程及产品,但是,如果环境有缺陷,它就有可能成为重要的风险源。

下面的风险检查表中的条目标识了与开发环境相关的风险:

- 是否有可用的软件项目管理工具?

- 是否有可用的软件过程管理工具?
- 是否有可用的分析及设计工具?
- 分析和设计工具是否适用于待建造产品?
- 是否有可用的编译器或代码生成器?
- 是否有可用的测试工具?
- 是否有可用的软件配置管理工具?
- 环境是否利用了数据库或数据仓库?
- 项目组的成员是否接受过每个所使用工具的培训?
- 是否有专家能够回答有关工具的问题?
- 工具的联机帮助及文档是否适当?

如果对于上述问题的答案多数是否定的，则软件开发环境是薄弱的，且风险很高。

#### (8) 与人员数目及经验相关的风险检查表

软件人员也是项目中的需要关注的风险因素，例如：

- 是否有优秀的人员可用?
- 人员在技术上是否配套?
- 是否有足够的人员可用?
- 开发人员是否能够自始至终地参加整个项目的工作?
- 项目中是否有一些人员只能部分时间工作?
- 开发人员对自己的工作是否有正确的期望?
- 开发人员是否接受过必要的培训?
- 开发人员的流动是否仍能保证工作的连续性?

如果对于上述问题中的任何一个答案是否定的，则需要进一步的调研，以评估潜在的风险。

## 2. 基于三层结构的检查表

美国的软件工程研究所（简称 SEI）的一个研究报告对于软件风险提出用类（Class）、元素（Element）、属性（Attribute）三个层次描述风险列表（见图 8-5）。对于类层分三组即产品工程、开发环境和项目限制。每个类组下包含若干元素，每个元素组下又包含若干属性。它们共同构成了风险检查表条目，通过属性值来识别、评估风险。具体条目要素见表 8-1。

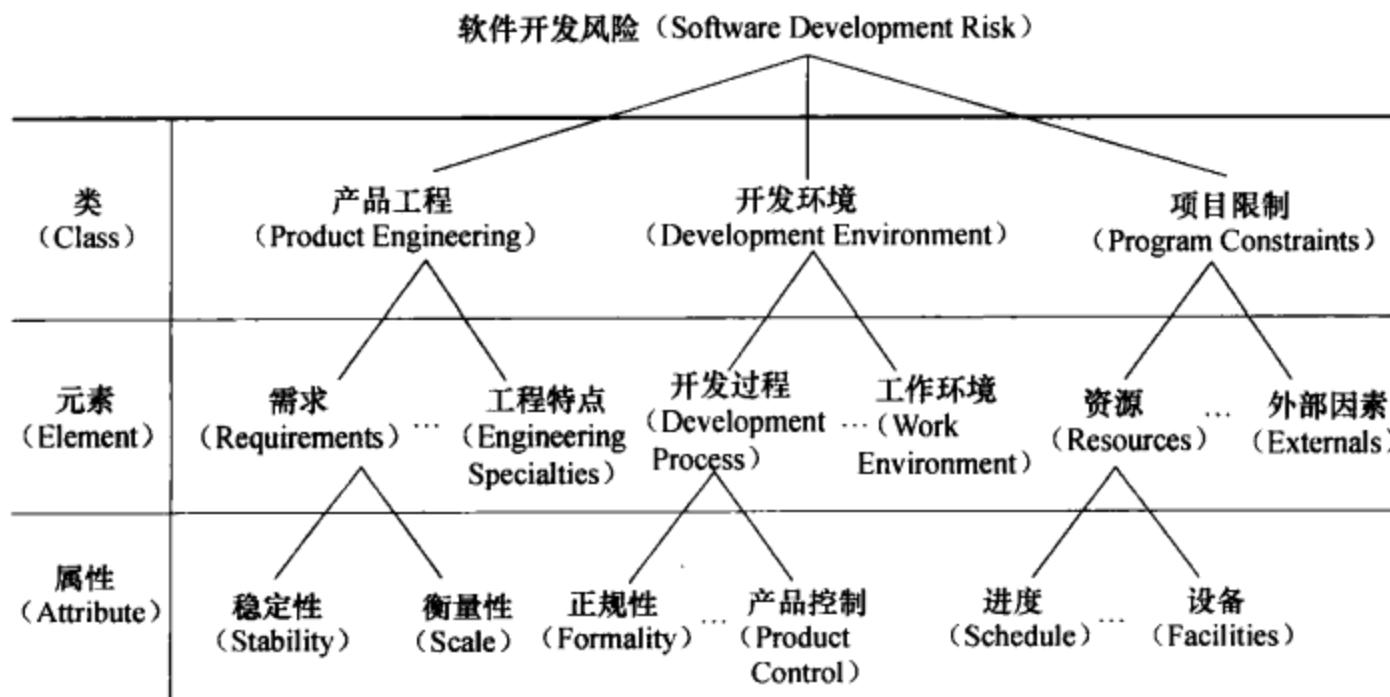


图 8-5 风险三层分析结构

表 8-1 三层风险检查表

类 (Class)	元素 (Element)	属性 (Attribute)
产品工程 (Product Engineering)	需求 (Requirements)	稳定性 (Stability) 完整性 (Completeness) 清晰性 (Clarity) 有效性 (Validity) 可行性 (Feasibility) 前瞻性 (Precedent) 衡量性 (Scale)
	设计 (Design)	功能性 (Functionality) 难度 (Difficulty) 接口 (Interfaces) 性能 (Performance) 易测性 (Testability) 硬件的限制 (Hardware Constraints) 非开发软件 (Non Developmental software)
	编码和单元测试 (Code and Unit test)	可行性 (Feasibility) 可测性 (Testing) 编码 (Coding/Implementation)
	集成和测试 (Integration and Test)	环境 (Environment) 产品 (Product) 系统 (System)
开发环境 (Development Environment)	工程特点 (Engineering Specialties)	可维护性 (Maintainability) 可靠性 (Reliability) 安全性 (Security) 人为因素 (Human Factors) 规范 (Specification)
	开发过程 (Development Process)	正规性 (Formality) 适合性 (Suitability) 过程控制 (Process Control) 过程的熟知程度 (Familiarity) 产品控制 (Product control)
	开发系统 (Development System)	容量 (Capacity) 适合性 (Suitability) 可用性 (Usability) 了解程度 (Familiarity) 可靠性 (Reliability) 系统的支持性 (System Support) 供应能力 (Deliverability)
	管理过程 (Management Process)	计划性 (Planning) 项目组织 (Project Organization) 管理经验 (Management Experience) 管理程序之间的接口 (Program Interfaces)
	管理方法 (Management Methods)	监控 (Monitoring) 人员管理 (Personnel Management) 质量保证 (Quality Assurance) 配置管理 (Configuration Management)
	工作环境 (Work Environment)	质量的态度 (Quality Attitude) 合作性 (Cooperation) 沟通 (Communication) 团队士气 (Morale)

(续)

类 (Class)	元素 (Element)	属性 (Attribute)
项目限制 (Program Constraints)	资源 (Resources)	进度 (Schedule) 人员 (Staff) 预算 (Budget) 设备 (Facilities)
	合同 (Contract)	合同类型 (Type of Contract) 约束条件 (Restriction) 前提 (Dependence)
	项目接口 (Program Interfaces)	客户 (Customer) 相关合同人 (Associate Contractors) 子合同 (Subcontractors) 总承包人 (Prime Contractor) 公司管理机构 (Corporate Management) 供方 (Vendors) 合同条款 (Politics)

### 3. 基于生存期的检查表

一般来说，项目生存期的各个阶段都可能存在风险，例如：

- 1) 初始阶段：在这个阶段进行大部分需求分析和部分设计工作，可能的风险事件包括：
  - 项目目标不清。
  - 项目范围不明确。
  - 用户参与少或与用户沟通少。
  - 对业务了解不够。
  - 对需求了解不够。
  - 没有进行可行性研究。
- 2) 设计阶段：在这个阶段进行大部分设计和部分编码工作，可能的风险事件包括：
  - 项目队伍缺乏经验，如缺乏有经验的系统分析员。
  - 没有变更控制计划，以至于变更没有依据，该变更的不变，不该变的也变，这样得来的设计势必会失败或者偏离用户需求。
  - 仓促计划，可能带来进度方面的风险。
  - 漏项，由于设计人员的疏忽，某些功能没有考虑进去。
- 3) 实施阶段：在这个阶段进行大部分编码和测试，也涉及少部分设计，可能的风险事件包括：
  - 开发环境没有具备好。
  - 设计错误带来的实施困难。
  - 程序员开发能力差，或程序员对开发工具不熟。
  - 项目范围改变（突然要增加或修改一些功能，需要重新考虑设计）。
  - 项目进度改变（要求提前完成任务等）。
  - 人员离开，在一个项目内软件开发工作有一定的连续性，需要移交和交接，有时人员离开对项目的影响会很大。
  - 开发团队内部沟通不够，导致程序员对系统设计的理解上有偏差。
  - 没有有效的备份方案。
  - 没有切实可行的测试计划。
  - 测试人员经验不足。
- 4) 收尾阶段：在这个阶段进行安装及维护（大部分部署），可能的风险事件包括：
  - 质量差。
  - 客户不满意。
  - 设备没有按时到货，资金不能回收。

### 8.3.5 其他方法

进行风险识别的时候，项目经理不一定能预测到所有的风险情况，而且有时当局者迷，与不同的项目相关人员进行有关风险的面谈，或者以电话、邮件等方式讨论将有助于识别那些在常规计划中未被识别的风险。在进行可行性研究时获得的项目前期面谈记录，往往是风险识别的很好素材。另外，SWOT（强项、弱项、机会、威胁）分析也是一个风险识别的方法。

### 8.3.6 风险识别的结果

风险识别的结果可以是一个风险清单表，如表 8-2 所示，表的第一列列出识别出来的风险，第二列是对风险进行的分类。风险清单表可以包括风险管理过程的所有结果，例如风险名称、风险标识、风险描述、风险类别、风险负责人、概率、影响、状态等，随着风险管理过程的进行，风险清单表的内容将逐步完善。对不同项目可能发生的风险事件不同，应该对具体项目识别出真正有可能发生在该项目的风险事件。而且还要对这些风险事件进行描述，如：可能性、可能后果范围、预计发生时间、发生频率等。

表 8-2 风险识别列表

风险	类别
规模估算可能非常低	产品规模
用户数量大大超出计划	产品规模
复用程度低于计划	产品规模
最终用户抵制该计划	商业影响
交付期限将被紧缩	商业影响
资金将会流失	客户特性
用户将改变需求	产品规模
技术达不到预期的效果	技术情况
缺少对工具的培训	开发环境
人员缺乏经验	人员数目及其经验
人员流动频繁	人员数目及其经验

## 8.4 风险评估

风险评估就是对识别出来的风险事件做进一步分析，对风险发生的概率进行估计和评价，对项目风险后果的严重程度进行估计和评价，对项目风险影响范围进行分析和评价，以及对于项目风险发生时间进行估计和评价。是衡量风险概率和风险对项目目标影响程度的过程。风险评估是评估风险三要素的另外两个要素：概率和影响。

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。

通过对风险及风险的相互作用的估算来评价项目可能结果的范围，从成本、进度及性能三个方面对风险进行评价，确定哪些风险事件或来源可以避免，哪些可以忽略不考虑（包括可以承受），哪些要采取应对措施。

在实践中，通常把风险评估的结果用风险发生的概率以及风险发生后对项目目标的影响来表示，风险 R 是该风险发生的概率 P 和影响程度 I 的函数：即  $R = f(P, I)$ 。然后按风险的严重性排序，确定最需要关注的前几个（TOP 10，一般说前 10 个，具体多少个可以视项目的具体情况而定）风险。

风险评估的方法包括定性风险评估和定量风险评估。

### 8.4.1 定性风险评估

定性风险评估主要是针对风险概率及后果进行定性的评估；例如采用历史资料法、概率分布法、风险后果估计法等。历史资料法主要是应用历史数据进行评估的方法，通过同类历史项目的风险发生情况，进行本项目的估算。概率分布法主要是按照理论或者主观调整后的概率进行评估的一种方法。另外，可以对风险事件后果进行定性的评估。按其特点划分为相对的等级，形成一种风险评价矩阵，并赋以一定的加权值来定性地衡量风险大小。例如，根据风险事件发生的概率度，将风险事件发生的可能性定性地分为若干等级。风险概率值是介于没有可能和确定之间的。风险概率度量也可以采用高、中、低或者极高、高、中、低、极低，以及不可能、不一定、可能和极可能等等不同方式表达。风险后果是风险影响项目目标的严重程度，可以从无影响到无穷大影响，风险后果的影响度量可以采用高、中、低或者极高、高、中、低、极低，以及灾难、严重、轻微、可忽略等方式表达。

在评估概率和风险后果时可以采用一些原则，例如小中取大原则、大中取小原则、遗憾原则、最大数学期望原则、最大可能原则等。

表8-3所示将风险发生的概率分为五个等级。同时可以将风险的影响程度分为若干等级。表8-4所示将风险后果的影响程度分为四个等级。

表8-3 风险发生概率的定性等级

等级	等级说明
A	极高
B	高
C	中
D	低
E	极低

表8-4 风险后果影响的定性等级

等级	等级说明
I	灾难性的
II	严重的
III	轻度的
IV	轻微的

将上述风险的后果影响和发生概率等级编制成矩阵并分别给以定性的加权指数，可形成风险评价指数矩阵，表8-5为一种定性风险评估指数矩阵的实例。

矩阵中的加权指数称为风险评估指数，指数1到20是根据风险事件可能性和严重性水平综合而确定的，通常将最高风险指数定为1，相对应于风险事件是频繁发生的并是有灾难性的后果的。最低风险指数20，对应于风险事件几乎不可能发生并且后果是轻微的。数字等级的划分具有随意性，但要便于区别各种风险的档次，划分得过细或过粗都不便于风险的决策，因此需要根据具体对象制定。

项目管理者可以根据项目的具体情况确定风险接受准则，这个准则没有统一的标准，例如可以对风险矩阵中的指数给出四种不同类别的决策结果：指数为1~5，为不可接受的风险；6~9为不希望有的风险，需由项目管理者们决策；10~17是有控制的接受的风险，需要项目管理者们评审后方可接受；18~20是不经评审即可接受的风险。

由于这种风险评估指数通常是主观制定的，而且定性的指标有时没实际意义，这是定性评估的一大缺点。因为无论是对风险后果的严重性或是风险发生的概率做出严格的定性量度都是很困难的。

当然，风险评价指数矩阵可以采用更为简单的方法，表8-6给出一种简易的定性评估的表格。其中，将风险发生的概率分为高、中、低三个等级，将风险后果的影响也分为高、中、低三个等级，通过表8-6矩阵定性地确定风险的评估结果。

风险后果影响和发生概率从风险管理的角度来看，它们各自起着不同的作用。一个具有高影响但低概率的风险因素不应当占用太多的风险管理时间，而具有中到高概率、高影响的风险和具有高概率及低影响的风险，就应该进行风险分析。例如图8-7是某项目的定性风险分析的一个结果，从图中可以看到Risk1, Risk2, Risk4, Risk9, Risk10是高风险，需要高度重视的。

表8-6 风险评估表

影响 概率	低	中	高
高	低	高	高
中	低	高	高
低	低	中	中

表8-7 项目定性分析结果

影响 概率	低	中	高
高	Risk3	Risk4	Risk9
中	Risk6 Risk7	Risk1, Risk2	Risk10
低		Risk8	Risk5

定性风险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。通过定性风险评估，人们能对项目风险有一个大致了解，可以了解项目的薄弱环节。但是，有时需要了解风险发生的可能性到底有多大，后果到底有多严重等。回答这些问题，就需要对风险进行定量的评价分析。

#### 8.4.2 定量风险分析

定量风险分析是在定性分析的逻辑基础上，给出各个风险源的风险量化指标及其发生概率，再通

过一定的方法合成，得到系统风险的量化值。它是基于定性风险分析基础上的数学处理过程。

定量风险分析是一种广泛使用的管理决策支持技术。一般在定性风险分析之后就可以进行定量风险分析。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后果，也分析项目总体风险的程度。定量风险评估可以包括访谈、盈亏平衡分析、敏感性分析、决策树分析、模拟法等方法。

### 1. 访谈

访谈技术用于量化对项目目标造成影响的风险概率和后果。访谈可以邀请以前搞过与本项目相类似项目的专家，这些专家运用他们的经验做出风险度量，其结果相当准确和可靠，甚至有时比通过数学计算与模拟仿真的结果还要准确和可靠。如果风险损失后果的大小不容易直接估计出来，可以将损失分解为更小的部分，再对其进行评估，然后将各部分评估结果累加，形成一个合计评估值。例如，如果使用3种新编程工具，可以单独评估每种工具未达到预期效果的损失，然后再把损失加到一起，这要比总体评估容易多了。

### 2. 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开发时间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。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。

盈亏平衡分析主要确定项目的盈亏平衡点。在平衡点上收入等于成本。此点是用以标志项目不亏不赢的开发量，用来确定项目的最低生产量。盈亏平衡点越低，项目盈利的机会就越大，亏损的风险越小。因此该点表达了项目生产能力的最低容许利用程度。一种对风险评估的常用技术是定义风险的参照水准，对绝大多数软件项目来讲，风险因素——成本、性能、支持和进度就是典型的风脸参照系。也就是说对成本超支、性能下降、支持困难、进度延迟都有一个导致项目终止的水平值。如果风险的组合所产生的问题超出了一个或多个参照水平值时，就终止该项目的工作，在项目分析中，风险水平参考值是由一系列的点构成的，每一个单独的点常称为参照点或临界点。如果某风险落在临界点上，可以利用性能分析、成本分析、质量分析等来判断该项目是否继续工作。图8-6表示了这种情况。

### 3. 决策树分析

决策树分析（decision tree analysis）是一种形象化的图表分析方法，它提供项目所有可供选择的行动方案以及行动方案之间的关系、行动方案的后果以及发生的概率，为项目经理提供选择最佳方案的依据。决策树分析采用损益期望值（Expected Monetary Value, EMV）作为决策树的一种计算值，它是根据风险发生的概率计算出一种期望的损益。

决策树是以一种便于决策者理解的，能说明不同决策之间和相关偶发事件之间的相互作用的图表来表达的。决策树的分支或代表决策或代表偶发事件，图8-7是一个典型的决策树

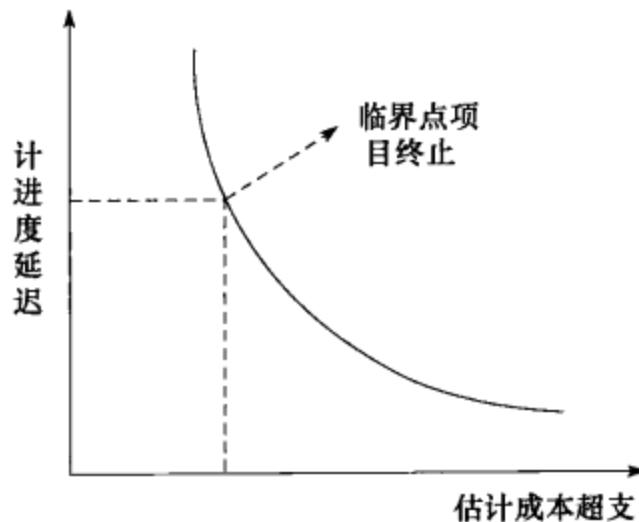


图8-6 项目临界点

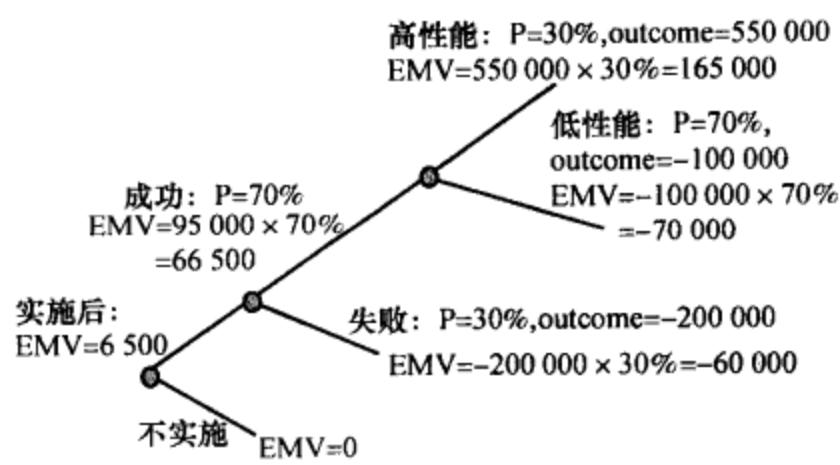


图8-7 决策树

图，是对实施某计划的风险分析。它用逐级逼近的计算方法，从出发点开始不断产生分枝以表示所分析问题的各种发展可能性，并以各分支的损益期望值中最大者（如求极小，则为最小者）作为选择的依据。从这个风险分析来看，实施计划后有 70% 概率的成功，30% 概率的失败。而成功后有 30% 概率是项目有高性能的回报  $outcome = 550\,000$ ，同时有 70% 概率是亏本的回报  $outcome = -100\,000$ ，这样项目成功的  $EMV = (550\,000 \times 30\% - 100\,000 \times 70\%) \times 70\% = 66\,500$ ，项目失败的  $EMV = 60\,000$ （概率为 30%），则实施后的  $EMV = 66\,500 - 60\,000 = 6\,500$ ，而不实施此项计划的  $EMV = 0$ 。通过比较，应该实施这个计划。

#### 4. 模拟法

模拟（Simulation）法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。例如蒙特卡罗技术（Monte Carlo）。

大多模拟项目日程表是建立在某种形式的“蒙特洛”分析基础上的。这种技术往往由全局管理所采用，对项目“预演”多次以得出如图 8-8 所示计算结果的数据统计分。图中的曲线显示了完成项目的累计可能性与某一时间点的关系。比如说，虚线的交叉点显示：在项目启动后 150 天之内完成项目的可能性为 50%。项目完成期越靠左则风险愈高，反之风险愈低。

蒙特洛分析也可能用来估算项目成本可能的变化范围。

#### 5. 敏感性分析

敏感性分析（sensitivity analysis）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。通过敏感性分析，使我们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，有助于我们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。

敏感性分析是判断不同变量的变化对结果的影响。在生活中，我们也会使用到这个方法，例如在进行贷款的时候，大家很注意不同利率、不同年限的贷款对月供的影响，从而判断哪个风险最小，以便做出决定。

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。

#### 8.4.3 风险评估的结果

通过风险评估分析，可以得到量化的、明确的、需要关注的风险管理清单，见表 8-8，清单上列出了风险名称、类别、概率、该风险所产生的影响以及风险的排序。可以从风险清单中选择排序靠前的几个风险（简称 TOP 10）作为风险评估的最终结果。

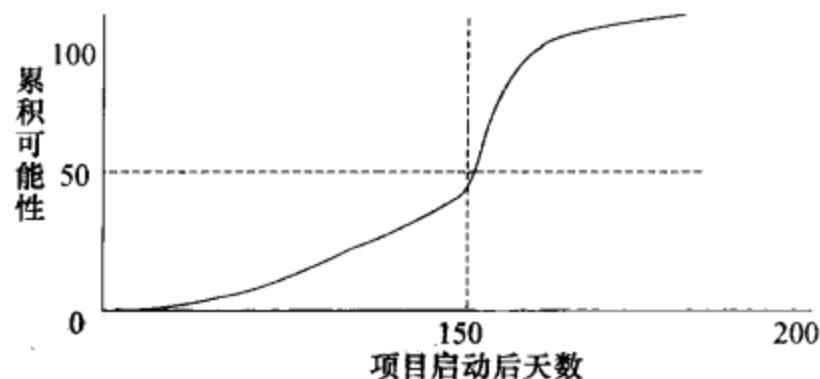


图 8-8 模拟分析

表 8-8 风险分析结果

风险	类别	概率	影响	排序
用户变更需求	产品规模	80%	5	1
规模估算可能非常低	产品规模	60%	5	2
人员流动	人员数目及其经验	60%	4	3
最终用户抵制该计划	商业影响	50%	4	4
交付期限将被紧缩	商业影响	50%	3	5
用户数量大大超出计划	产品规模	30%	4	6
技术达不到预期的效果	技术情况	30%	2	7
缺少对工具的培训	开发环境	40%	1	8
人员缺乏经验	人员数目及其经验	10%	3	9

在风险管理过程中，也可以通过量化风险条目检查表中的条目来评估风险，图 8-9 是根据多个项目的实际数据得出对三层结构检查表的量化风险分析结果：其中产品工程占 30% 的风险，开发环境占 33% 的风险，项目限制占 37% 的风险。而产品工程中的各个因素占有的风险比例分别是需求为 53%，设计为 27%，集成和测试为 14%，工程特点为 6%，编码和单元测试为 2%，可见需求和设计是风险的主要来源。而需求的各个属性的风险来源比例分别为复杂性为 36%，稳定性为 21%，可行性为 14%，有效性为 10%，先例性为 8%，规模方面为 7%，清晰性为 4%。可见需求的复杂性和稳定性是需求风险的主要来源。

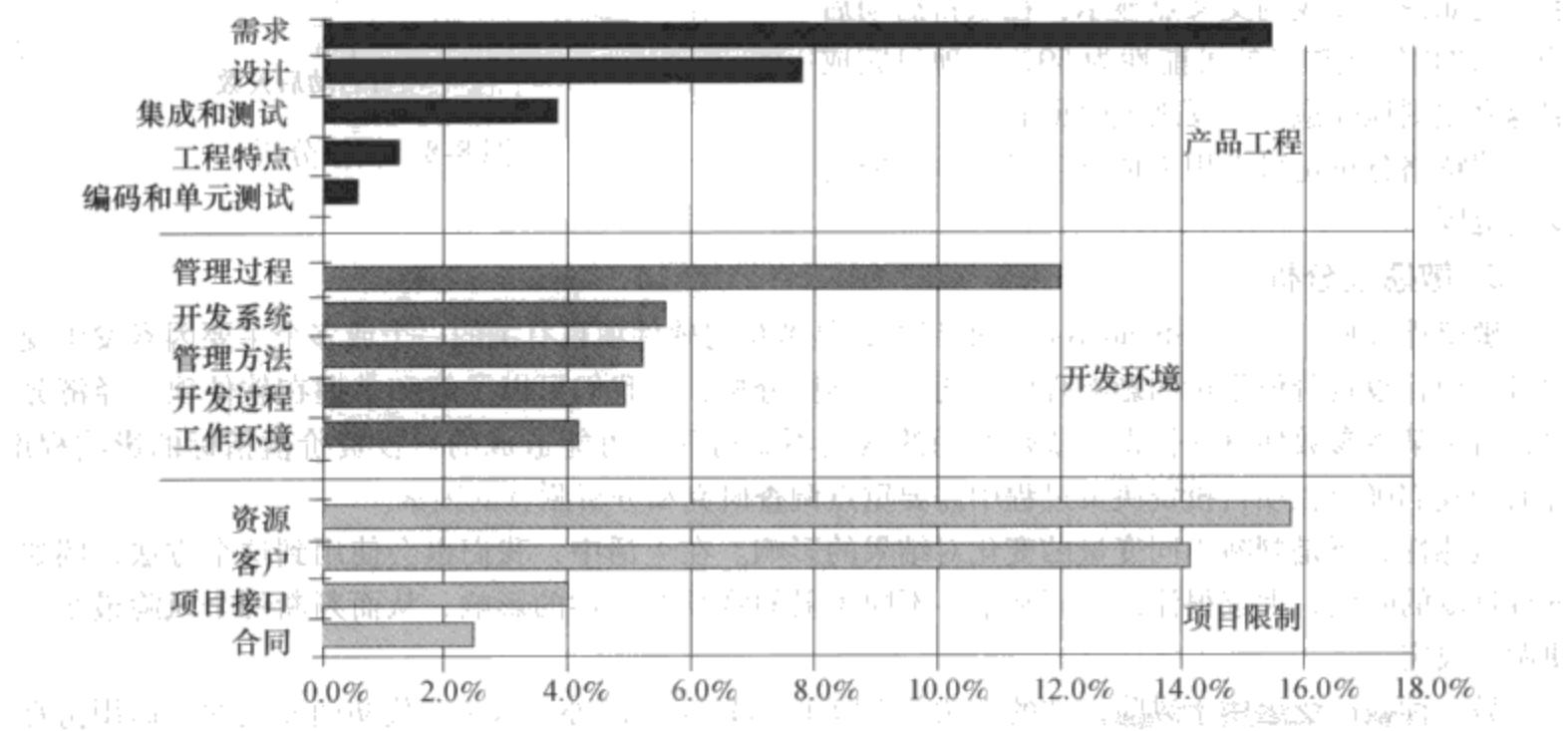


图 8-9 风险识别结果

## 8.5 风险应对计划

风险规划是针对风险分析的结果，为提高实现项目目标的机会，降低风险的负面影响而制定风险应对策略和应对措施的过程，即通过制定一系列的行动和策略来对付、减少以至于消灭风险事件。

项目管理永远不能消除所有的风险，但是常常通过一定的风险规划，采取必要的风险控制策略可以消除特定的风险事件。风险规划的输出是项目风险计划。

项目开发是一个高风险的活动，如果项目采取积极的风险管理策略，就可以避免或降低许多风险，反之，就有可能使项目处于瘫痪状态。规划降低风险的主要策略是回避风险、转移风险、损失控制以及自留风险。

### 8.5.1 回避风险

风险事件常常可以通过及时改变计划来制止或避免。回避风险是通过分析找出发生风险事件的原

因，对可能产生的风险尽可能的规避，采取主动放弃或拒绝使用导致风险的方案。这样可以直接消除风险损失，回避风险具有简单、易行、全面、彻底的优点，能将风险的发生概率保持为零，因为已经将风险的起因消除了，从而保证项目的安全运行。回避风险也称替代战略。排除特定风险往往靠排除风险起源。项目管理组不可能排除所有风险，但特定的风险事件往往是可以排除的。在采取回避风险时，应注意以下几点：

- 1) 对风险有足够认识，而且风险发生概率极高、风险后果影响很严重时可以采用这个策略。
- 2) 当其他的风险策略不理想的时候，才可以考虑这个策略。
- 3) 不是所有的风险都可以采取回避策略的，如地震或者洪涝灾害等是无法回避的。
- 4) 由于回避风险只是在特定范围内及特定的角度上才有效，因此，避免了某种风险，有可能产生另一种新的风险。例如避免采用新的技术，可能导致开发出来的产品技术落后的风险。

### 8.5.2 转移风险

转移风险是指为避免承担风险损失，而有意识的将损失或与损失有关的财务后果转嫁给另外的单位或个人去承担。例如将有风险的一个软件项目分包给其他的分包商，或者通过免责合同等开脱手段说明不承担后果。

转移风险有时也称通过采购转移风险，即从本项目组织外采购产品和服务，常常是针对某些种类风险的有效对策。比如，与使用特殊科技相关的风险就可以通过与有此种技术经验的组织签定合同减缓风险。采购行为往往将一种风险置换为另一种风险。比如，如果销售商不能够顺利销售，那么用制定固定价格的合同来减缓成本风险会造成项目时间进程受延误的风险；而相同情形下，将技术风险转嫁给销售商又会造成难以接受的成本风险。

投保也是一种转移风险的策略，保险或类似保险的操作对一些风险类别是行之有效的。在不同的应用领域，险种的类别和险种的成本也相应不同。

### 8.5.3 损失控制

损失控制是指风险发生前消除风险可能发生的根源，并减少风险事件的概率，在风险事件发生后减少损失的程度。故损失控制的基本点是在于消除风险因素和减少风险损失。

根据目的不同，损失控制分为损失预防和损失抑制。

#### 1. 损失预防

损失预防是指风险发生前为了消除或减少可能引起风险的各种因素而采取的各种具体措施，制定预防性计划，也就是设法消除或减少各种风险因素，以降低风险发生的概率。预防性计划包括针对一个确认的风险事件的预防方法以及风险发生后的应对步骤。

例如经过风险识别发现，项目组的程序员对所需开发技术不熟，可以事先进行培训来减轻对项目的影响，这是损失预防方法。

再如，为了避免“客户不满意”这个风险事件，可以采用如下的损失预防方法：

- 1) 业务建模阶段要让客户参与。
- 2) 需求阶段要多和客户沟通，了解客户真正的需求。
- 3) 采用目标系统的模型向客户演示，并得到反馈意见。
- 4) 要有双方认可的验收方案和验收标准。
- 5) 做好变更控制和配置管理。

#### 2. 损失抑制

损失抑制（也称风险减缓）是指风险发生时或风险发生后为了缩小损失幅度所采用的各项措施。通过降低风险事件发生的概率或得失量来减轻对项目的影响。

为了避免自然灾害造成的后果，在一个大的软件项目中考虑了异地备份来进行损失抑制。如果一个系统有问题，可以启动另外一个系统。再如：有一个软件集成项目中包括了设备，而且计划在部署

阶段之前设备必须到位，而这些设备从厂家直接进货。经过分析发现有可能不能按时进货，那就应该考虑备选方案。又比如：在一个软件开发项目中，某开发人员有可能离职，离职后会对项目造成一定的影响，则应该对这个风险事件开发应对计划如下（针对损失控制策略）：

- 1) 进行调研，确定流动原因。
- 2) 在项目开始前，把缓解这些流动原因的工作列入风险管理计划。
- 3) 项目开始时，做好计划一旦人员离开时便可执行，以确保人员离开后项目仍能继续进行。
- 4) 制定文档标准，并建立一种机制，保证文档及时产生。
- 5) 对所有工作进行细微详审，使更多人能够按计划进度完成自己的工作。
- 6) 对每个关键性技术人员培养后备人员。

#### 8.5.4 自留风险

自留风险又称承担风险，它是一种由项目组织自己承担风险事件所致损失的措施。这种接受可以是积极的（如制定预防性计划来防备风险事件的发生），一般是经过合理判断和谨慎研究后，决定承担风险；或者是不知道风险因素的存在而承担下来，这是消极的自我承担（如某些工程运营超支则接受低于预期的利润，或者由于疏忽而承担的风险）。

#### 8.5.5 风险规划的结果

针对风险事件采取应对措施，开发应对计划，一旦风险事件发生，就可以实施风险应对计划。在考虑风险成本之后，决定是否采用上述策略。例如，在软件测试期间经常会发现故障，因此一个合理的项目必须作好发现故障时对它们进行修复的计划。同样，项目开发过程中几乎总是会出现某些变更申请，因此项目管理必须相应地准备好变更计划，以处理这些的事件。

风险规划的结果是一个项目风险计划或者说风险管理方案。在整个项目进程中都应将管理风险的过程记录下来。除了记录风险识别和风险量化程序的结果外，还应记录包括：谁对处理某些风险负责，怎样保留初步风险识别和风险量化的输出项，预防性计划怎样实施，以及储备如何分配等。

一个风险应对计划方案可以是正式的或非正式的。可以是细致入微或框架性的，这主要依据项目而定。

风险应对计划应该包括项目风险的来源、类型、项目风险发生的可能时间及范围、项目风险事件带来的损失，以及项目风险可能影响的范围等。见表 8-9 的例子。它是一个项目的风险分析的结果，通过输入风险识别项，然后对风险进行分析，得出风险的风险值，然后按照风险的大小排序，给出需要关注的 TOP 10 风险表，同时，确定风险应对策略。

表 8-9 风险分析表

排序	输入	风险事件	可能性	影响	风险值	采取的措施
1	系统设计评审	没有足够的时间进行产品测试	70%	50%	35%	1) 采取加班的方法 2) 修改计划去掉一些任务 3) 与客户商量延长一些时间
2	WBS	对需求的开放式系统标准没有合适的测试案例	20%	80%	16%	找专业的测试公司完成测试工作
3	需求和计划	采用新技术可能导致进度的延期	50%	30%	15%	1) 培训开发人员 2) 找专家作指导 3) 采取边开发边学习的方法，要求他们必须在规定的时间内掌握技术
.....						

项目经理在制定项目计划的时候，应该制定一个需要关注的风险管理计划，至于采用什么形式风险管理计划，项目经理可以自行决定，但是 TOP 10 风险清单是很重要的工具。当然，主要看项目的具体要求。

一个大型软件的开发大约存在 30~40 种风险。如果每种风险都需要多个风险管理步骤，那么风险管理本身也可以构成软件开发过程的一个子项目。人们必须懂得，风险管理不仅需要人力资源，而且还需要经费的支持。所以在软件项目开发过程中，当对软件的期望很高时，一般都会进行项目风险分析、预测、评估、管理及监控等风险管理。通过风险管理可以使项目进程更加平稳，可以获得很高的跟踪和控制项目的能力，并且可以增强项目组成员对项目如期完成的信心。风险管理是项目管理中很重要的管理活动，有效的实施软件风险管理是软件项目开发工作顺利完成的保证。

在软件项目管理中，应该任命一名风险管理者，该管理者的主要职责是在制订与评估规划时，从风险管理的角度对项目规划或计划进行审核并发表意见，不断寻找可能出现的任何意外情况，试着指出各个风险的管理策略及常用的管理方法，以随时处理出现的风险。

## 8.6 风险管理建议

软件项目管理从某种意义上讲，就是风险管理。我们尽量去定义明确不变的需求，以便进行计划并高效管理，但商业环境总是快速变化的，甚至是无序的变化。所以，软件企业在进行项目管理的过程中，必须采用适合自己的风险管理方法进行风险管理，以确保软件项目在规定的预算和期限内完成项目。

有的项目有风险，有的是高风险，而有的项目则是在冒险。很少有没有风险的项目。风险和快速开发是一对矛盾，也是一对协调组合。风险延长进度，高风险更如此的。如果你管理的项目需求含糊、人员没有培训、不熟悉产品应用领域、技术很新等，这就要求你开发出一个挑战性的风险计划，风险计划可以帮助你摆脱困境，用风险分析的结果说明问题。

为此，项目经理应该通过学习掌握项目风险管理所必备的知识，总结项目中常见的风险及其对策。下面简单罗列软件项目常见风险及其预防措施：

### (1) 合同风险

签订的合同不科学、不严谨，项目边界和各方面责任界定不清等是影响项目成败的重大因素之一。

预防这种风险的办法是项目建设之初项目经理就需要全面准确地了解合同各条款的内容、尽早和合同各方就模糊或不明确的条款签订补充协议。

### (2) 需求变更风险

需求变更是软件项目经常发生的事情。一个看似很有“钱途”的软件项目，往往由于无限度的需求变更而让项目开发方苦不堪言，甚至最终亏损。

预防这种风险的办法是项目建设之初就和用户书面约定好需求变更控制流程、记录并归档用户的需求变更申请。

### (3) 沟通不良风险

项目组与项目各干系方沟通不良是影响项目顺利进展的一个非常重要的因素。

预防这种风险的办法是项目建设之初就和项目各干系方约定好沟通的渠道和方式，项目建设过程中多和项目各干系方交流和沟通，注意培养和锻炼自身的沟通技巧。

### (4) 缺乏领导支持风险

上层领导的支持是项目获得资源（包括人力资源、财力资源和物料资源等）的有效保障，也是项目遇到困难时项目组最强有力的“后台支撑”。

预防这种风险的办法是主动争取领导对项目的重视，确保和领导的沟通渠道畅通，经常向领导汇报工作进展。

### (5) 进度风险

有些项目对进度要求非常苛刻（进度要求不高的项目，我们同样要考虑该风险），项目进度的延迟意味着违约或市场机会的错失。

预防这种风险的办法一般是分阶段交付产品、增加项目监控的频率和力度、多运用可行的办法保证工作质量避免返工。

### (6) 质量风险

有些项目，用户对软件质量有很高的要求，如果项目组成员的开发经验不足，则需要密切关注项目的质量风险。

预防这种风险的办法一般是经常和用户交流工作成果、采用符合要求的开发流程、认真组织对产出物的检查和评审、计划和组织严格的独立测试等。

### (7) 系统性能风险

有些软件项目属于多用户并发的应用系统，系统对性能要求很高，这时项目组就需要关注项目的性能风险。

预防这种风险的办法一般是在进行项目开发之前先设计和搭建出系统的基础架构并进行性能测试，确保架构符合性能指标后再进行后续工作。

### (8) 工具风险

软件项目开发和实施过程，所必须用到的管理工具、开发工具、测试工具等是否能及时到位，到位的工具版本是否符合项目要求等，是项目组需要考虑的风险因素。

预防这种风险的办法一般是在项目的启动阶段就落实好各项工具的来源或可能的替代工具，在这些工具需要使用之前（一般需要提前一个月左右）跟踪并落实工具的到位事宜。

### (9) 技术风险

在软件项目开发和建设的过程中，技术因素是一个非常重要的因素。项目组一定要本着项目实际要求，选用合适、成熟的技术，千万不要无视项目的实际情况，而选用一些虽然先进但并非项目所必须且自己又不熟悉的技术。如果项目所要求的技术项目成员不具备或掌握不够，则需要重点关注该风险因素。

预防这种风险的办法是选用项目所必须的技术，在技术应用之前，针对相关人员开展好技术培训工作。

### (10) 团队成员能力和素质风险

团队成员的能力（包括业务能力和技术能力）和素质，对项目的进展、项目的质量具有很大的影响，项目经理在项目的建设过程需要实时关注该因素。

预防这种风险的办法是在用人之前先选对人、开展有针对性的培训、将合适的人安排到合适的岗位上。

### (11) 团队成员协作风险

团队成员是否能齐心协力为项目的共同目标服务，是影响进度和质量的关键因素。

预防这种风险的办法是项目在建设之初项目经理就需要将项目目标、工作任务等和项目成员沟通清楚，采用公平、公正、公开的绩效考评制度，倡导团结互助的工作风尚等。

### (12) 人员流动风险

项目成员特别是核心成员的流动给项目造成的影响是非常可怕的。人员的流动轻则影响项目进度，重则导致项目无法继续甚至被迫夭折。

预防这种风险的办法是尽可能将项目的核心工作分派给多人（而不要集中在个别人身上）、加强不同类型人才的培养和储备。

### (13) 工作环境风险

工作环境（包括办公环境和人文环境）的好坏直接影响项目成员的工作情绪和工作效率。

预防这种风险的办法是在项目建设之前就选择和建设好适合项目特点和满足项目成员期望的办公环境，在项目的建设过程中不断培育和调整出和谐的人文环境。

### (14) 系统运行环境风险

目前，大部分项目系统集成和软件开发是分开进行的（甚至由不同公司承接）。因此，软件系统赖以运行的硬件环境和网络环境的建设进度对软件系统是否能顺利实施具有相当大的影响。

预防这种风险的办法是和用户签定相关的协议、跟进系统集成部分的实施进度、及时提醒用户等。

### (15) 分包商风险

有些项目可能会涉及到将系统的部分功能分包出去，这时项目组就需要关注项目的分包商风险。

预防这种风险的办法一般是指定分包经理全程监控分包商活动、让分包商采用经认可的开发流程、督促分包商及时提交和汇报工作成果、及时审计分包商工作成果等。

世间万物总是发展变化的，风险亦可能随时出现和变化。项目经理应该将“防患于未然”牢记于心。项目经理不断培养和强化项目整个团队成员的风险意识，是确保项目顺利进展的最有效方法之一。

以上列举的这些风险，应该是软件项目建设中经常出现的主要风险，但由于项目本身的个性化特征，针对具体的项目，肯定会出现一些我们上面没有列举，甚至是事先根本无法预期的风险，这就需要我们项目经理有敏锐的“嗅觉”去识别它们，从而更好地预防和控制它们。

## 8.7 校务通系统案例分析

校务通系统项目的主要风险是开发人员对客户需求中的学校管理环境不是很熟悉，另外，客户要求的进度比较紧，而且具体需求不是很明确，下面的这个风险列表就是通过一系列的风险识别、风险评估、风险应对，最后得出项目风险计划（Top 风险清单）见表 8-10。

表 8-10 风险计划

排序	输入	风险事件	可能性	影响	风险值	风险应对措施
1	客户的 SOW	需求不明确，增加需求，导致需求蔓延	70%	50%	35%	1) 采取加班的方法 2) 修改计划去掉一些任务 3) 与客户商量延长一些时间
2	合同	进度要求紧，合同金额有限	30%	50%	15%	可以请一些实习的学生做辅助工作，一来成本不高，二来可以加快进度
3	历史项目信息	开发人员对测试工作不重视	30%	40%	12%	1) 强制性要求每段代码保留测试单元，由 SQA 检查 2) 加入专门的测试人员
4	WBS	供货商、外包商的质量问题	20%	50%	10%	多选择几个可以作为备份的外包商和供应商
5	历史项目信息	开发人员的流动	15%	60%	9%	1) 注意项目团队的沟通，及时了解开发人员的动态 2) 控制好项目过程中的文档 3) 从其他的项目组借调人员 4) 从外部招聘有过此类开发经验人员

## 8.8 小结

风险是伴随着软件项目过程而产生的，在软件项目中进行风险管理是必须的，如果忽略了风险，风险就会导致项目的失败。风险管理过程包括：风险识别、风险评估、风险规划、风险控制等步骤，而且这些步骤是循环进行的。“风险条目检查表”是风险识别和风险评估中常用的一种方法。风险评估包括定性风险分析和定量风险分析，其中决策树是一个很重要的方法，风险应对策略主要包括：回避风险、转移风险、损失控制以及自留风险。风险计划是软件项目计划的一个重要部分，它是项目经理在软件项目进行过程中需要不断监视的依据。对任何一个软件项目，可以有最佳的期望值，但更应该要有最坏的准备，“最坏的准备”在项目管理中就是进行项目的风险分析。

## 8.9 练习题

### 一、选择题

1. 在风险分析中使用灵敏度分析可以（ ）。
  - A. 在风险定量分析中取代不确定性分析
  - B. 估算管理层对风险的厌恶级别
  - C. 估算一个项目变量变更对整个项目的影响
  - D. 理解达到项目目标的概率

2. 以下各项关于回避风险的陈述除了（ ）都是正确的。
- A. 消除引起风险的因素
  - B. 决定不对风险过高的项目进行投标
  - C. 风险倘若发生，就接受后果
  - D. 决定取消采用具有高风险的新技术，而采用原来熟悉的技术
3. 下面哪项不是风险管理的过程。（ ）
- A. 风险评估
  - B. 风险识别
  - C. 风险规划
  - D. 风险收集
4. 在靠近河边的某建筑工地，洪水毁坏了所有挖掘的地基，这是发生了什么类型的风险？（ ）
- A. 已知风险（Known knowns）
  - B. 可预测风险（Known unknowns）
  - C. 不可预测风险（Unknown unknowns）
  - D. 其他
5. 在风险分析过程中，确定已经识别的一个风险事件是无法避免的，也是不能减轻的，也不能投保，这是一个关键的风险事件，一旦发生可能造成项目的失败，项目经理最佳的选择是（ ）。
- A. 降低风险的级别，项目团队将找到一个克服故障的方法
  - B. 特别关注，加强管理该风险事件和所有的相关事件
  - C. 让风险评估小组继续分析该风险事件，直到降低预期影响
  - D. 忽略风险评估，因为不管赋予什么值，都只是一个估算。
6. 在风险分析过程中，蒙特卡罗技术是一种（ ）。
- A. 模拟分析方法
  - B. 盈亏平衡分析
  - C. 敏感性分析
  - D. 决策树风险分析
7. 风险的三个属性是（ ）。
- A. 风险发生的时间、地点、负责人
  - B. 风险事件、时间、影响
  - C. 风险事件、概率、影响
  - D. 风险数量、风险影响程度、概率
8. 购买保险是（ ）类型的风险策略。
- A. 风险转移
  - B. 风险规避
  - C. 风险抑制
  - D. 风险自担
9. 在什么情况下可以采用德尔菲方法进行风险识别。（ ）
- A. 存在模拟模型
  - B. 需要大家的统一意见
  - C. 有大量历史项目
  - D. 风险概率明确
- 二、利用决策树风险分析技术来分析如下两种情况的，以便决定你会选择哪种方案？（要求画出决策树）
- 方案 1：随机投掷硬币两次，如果两次投掷的结果都是硬币正面朝上，你将获得 10 元；投掷的结果背面每朝上一次你需要付出 1.5 元。
  - 方案 2：随机投掷硬币两次，你需要付出 2 元；如果两次投掷的结果都是硬币正面朝上，你将获得 10 元。

## ■ 软件项目合同计划

合同是规定项目执行各方行使其权利和义务、具有法律效力的文件。合同应该是一个项目合法存在的标志，围绕合同，存在合同签署之前的一些工作以及合同签署之后的一些工作。这章进入本书路线图的合同计划，如图 9-1。

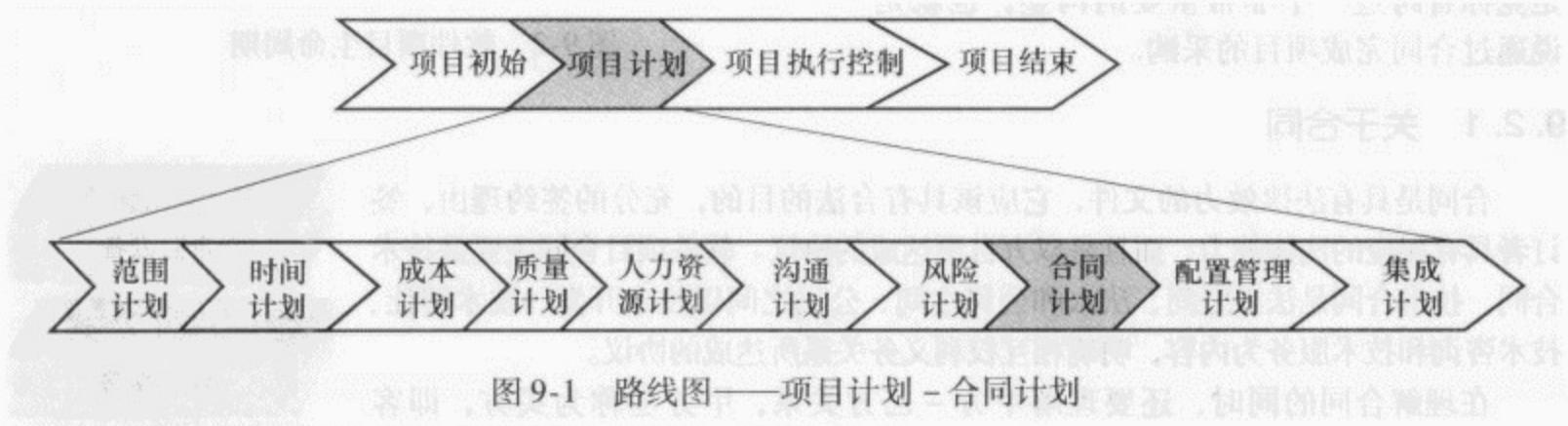


图 9-1 路线图——项目计划 - 合同计划

### 9.1 项目采购

为了执行项目而从项目团队外部采购或者获取产品、服务或者结果的过程，称为采购。对于软件产品，一般采购可以分为两大类，一类是对已经在市场流通的软件产品进行采购。例如，某企业想做信息化建设项目，涉及到数据库，就可以在目前市面通用的几种厂家数据库中选择。然后根据自己的需求，通过询价、签合同、安装培训等过程来购买此类产品。这种采购过程基本已经形成几套通用的解决方案，比较简单，中国企业在处理这类产品的采购时，大部分都处理的较好。个别的企业由于需求分析不清晰，培训工作不到位等原因，也会产生购买的产品不适用，或不会用的情况。另外一类软件产品采购的形式是外包采购。它是指在市场上没有现成的产品或者没有适合自己企业需求的产品的情况下，需要以定制的方式把项目（功能模块）承包给其他企业。例如某企业需要实施企业资源计划项目（ERP），虽然可以购买一些现成的商用软件，但是基于本企业业务流程的管理软件必须定制，必须自己开发或外包给别的公司。

企业在执行项目过程中需要评判需要采购什么，什么时候需要采购，从哪里采购，是否采购。在决定是采购还是自制的过程中可以选择自制/外购分析（Make- or Buy analysis）方法或者专家方法（Expert judgment）来决策。自制/外购分析是决定是一个产品或者服务是外购还是企业内部自己来完成的方法，通常需要考虑一些财务分析数据。例如成本、可利用的内部产能、控制水平、供应商的可获利性、保密性要求、持续需求等。

可以基于 WBS 来决定是自己开发还是购买。如果选择不确定时，还要进一步研究。一些情况，购买是划算的，而另外的情况自己制造是更合适的。如果项目决定采购（外包），则需要选择合适的乙方，进行合同规划。

### 9.2 合同规划

一个软件项目如果选择通过外购（包）方式完成的话，一般以招投标的形式开始，如图 9-2 所示。作为软件的客户（需求方）根据自己的需要，提出软件的基本需求，并编写招标书，同时将招标书以某种方式传递给竞标方，所有的竞标者们都会认真地编写投标书。每一个竞标者都会思考如何以较低的费用和较高的质量来解决客户的问题，然后都会交付一份对问题理解的说明书以及相应的解决方案，

同时也会附上一些资质证明和自己参与类似项目的经验介绍，以向客户强调各自的资历和能力。有时竞标者为了最后中标，会花大力气开发一个系统原型。在众多能够较好满足客户需要的投标书中，客户会选择一个竞标者。其间，竞标者会有与客户之间进行各种公开和私下的讨论，以及各种公关活动。这是售前的任务，此时，作为竞标方的项目经理已经参与其中的工作，经过几个回合的切磋，得到用户的认可，并中标后，就要开始着手合同书的编写等相关事宜，此时，就有质量保证人员和相关的法律人员的介入。合同签订之后，这是一个重要的里程碑，也是竞标者跨过一个非常重要的沟壑，也就是说通过合同完成项目的采购。

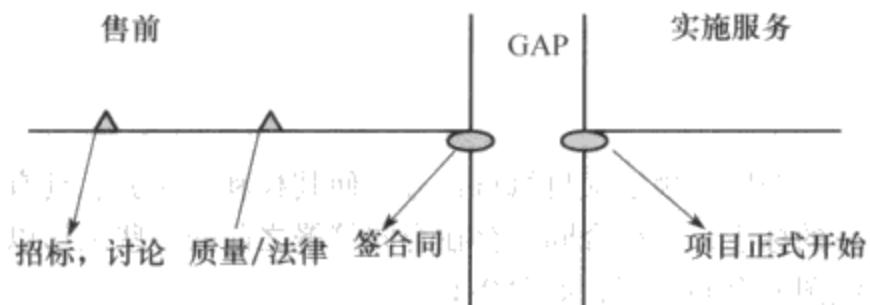


图 9-2 软件项目生命周期

### 9.2.1 关于合同

合同是具有法律效力的文件，它应该具有合法的目的，充分的签约理由，签订者具有相应的法律能力，而且是双方自愿达成的协议。软件项目合同主要是技术合同，技术合同是法人之间、法人和公民之间、公民之间以技术开发、技术转让、技术咨询和技术服务为内容，明确相互权利义务关系所达成的协议。

在理解合同的同时，还要理解甲方 - 乙方关系，甲方也称为买方，即客户，是产品的接受者。乙方也称为卖方，又称分包商、卖主、供应商，是产品的提供者。甲乙双方之间存在的法律合同关系称为合同当事人。

技术合同管理是围绕合同生存期进行的。合同生存期划分为四个基本阶段，即：合同准备、合同签署、合同管理与合同终止，见图9-3。合同双方当事人在不同合同环境下承担不同角色，这些角色包括：甲方、乙方。

一般说，在合同的管理过程中甲乙双方可以各自确定一个合同管理者，负责合同相关的所有管理工作，称其为合同管理者。

企业在甲方合同环境下的关键要素是提供准确、清晰和完整的需求、选择合格的乙方并对采购对象（“采购对象”包括产品、服务、人力资源等）进行必要的验收。

企业在乙方合同环境下的关键要素是了解清楚甲方的要求并判断企业是否有能力来满足这些需求。

### 9.2.2 合同类型

当项目的甲乙双方达成一致后，应当将这个项目所涉及的一些事情，包括价格、质量标准、项目时间等以合同的方式写下来。如果必要的话，实地考察、请律师审查，或者做出合适的修改。

合同有很多不同类型，可以根据项目的工作、预计的项目时间和甲乙双方的关系来确定合同的类型。表9-1是常用的合同类型及其属性的一览表。

表 9-1 合同类型一览表

合同类型	属性	风险
成本加成本百分比 (Cost Plus Percentage of Cost, CPPC)	实际成本加上乙方利润	甲方承担成本超出的风险。这是一个对买方而言很危险的合同类型
成本加固定费用 (Cost Plus Fixed Fee, CPFF)	实际成本加上乙方利润	甲方承担成本超出的风险。甲方的风险比较大
成本加奖金 (Cost Plus Incentive Fee, CPIF)	实际成本加上乙方利润	甲方承担成本超出的风险
固定价格 (Fixed Price, FP)	甲乙双方就合同产品协商的价格，其中包括对乙方的奖励金	乙方承担风险

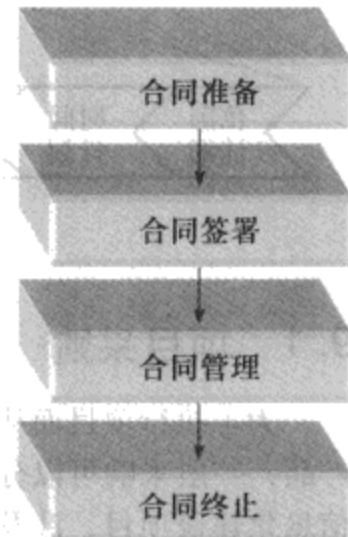


图 9-3 合同生存期

(续)

合同类型	属性	风险
一次付清 (lump sum)	甲乙双方就合同产品协商的价格，其中也包括对乙方的奖励金	乙方承担风险
固定总价 (Firm Fixed Price, FFP)	甲乙双方就合同产品协商的价格	乙方承担风险
固定价格加奖励费 (Fixed Price Plus Incentive Fee, FPIF)	甲乙双方就合同产品协商的价格，其中也包括对乙方的奖励金	乙方承担风险
时间与材料合同 (time and materials)	按照乙方使用的时间和材料来计算价格	没有最大开销约束的合同可以导致成本超支
单价合同 (unit price)	一个产品或者时间度量单位的价格 (例如：一个工程师的工时是130美元)	产品不同风险也不同。如果合同中没有明确时间长度，时间将是最大的风险

下面结合例子具体说明合同类型：

1) CPPC 合同类型：如果一个项目的成本百分比是 15%，假设估计的成本是 10 万，则合同金额为  $10 + 10 \times 15\% = 11.5$  万。而如果实际成本是 20 万的话，则合同金额应为  $20 + 20 \times 15\% = 23$  万。即乙方花费的成本越高，获得的合同金额也越高，如果乙方故意拖延工期，增加成本，则合同的金额也加大了，所以这是一个对甲方而言很危险的合同类型。

2) CPFF 合同类型：如果一个项目固定费用是成本的 15%，假设估计的成本是 10 万，则合同金额为  $10 + 10 \times 15\% = 11.5$  万。而实际的成本是 20 万的话，则合同金额为  $20 + 10 \times 15\% = 21.5$  万。即固定费用不变。这样，对乙方故意拖延工期，有一定的控制，但是没有完全控制，因为拖延工期，虽然固定费用不变，但是合同金额还是增加的，所以对甲方还是有比较大的风险。

3) CPIF 合同类型，它增加了激励的机制：如果一个项目激励的比例是 80/20，即将节约成本的 20% 作为激励。假设估计的成本是 10 万，利润是 1 万，则如果实际成本是 10 万，则合同金额为 11 万；如果实际成本是 8 万，则合同金额为  $8 + 1 + 2 \times 20\% = 9.4$  万，即将节约的 2 万成本的 20% 作为激励。这种合同类型甲方还是承担成本超出的风险，但是又进一步约束了乙方，甲方的风险在减低，乙方的风险在增加。合同激励可以协调双方的目标和利益。

4) FPI 合同类型：如果一个项目的假设目标成本是 100 万，最高价格是 110 万，利润是 10 万，激励的比例是 70/30，即，乙方获得节约成本的 30%，当实际的成本是 80 万时，即节约成本 20 万，则合同金额为  $80 + 10 + 20 \times 30\% = 96$  万；当实际的成本是 150 万时，则合同金额为 110 万。即乙方节约成本有奖励，超出成本的部分自己承担，这样乙方的风险增加，甲方的风险降低了。

5) FFP 合同类型：如果一个项目的合同价格 100 万，不管成本是 80 万还是 150 万，合同的金额 100 万是固定的，所以，这种合同类型，甲方的风险相对最小，乙方的风险相对最大。

综上所述，几种合同类型的风险关系可以用图 9-4 表达出来。

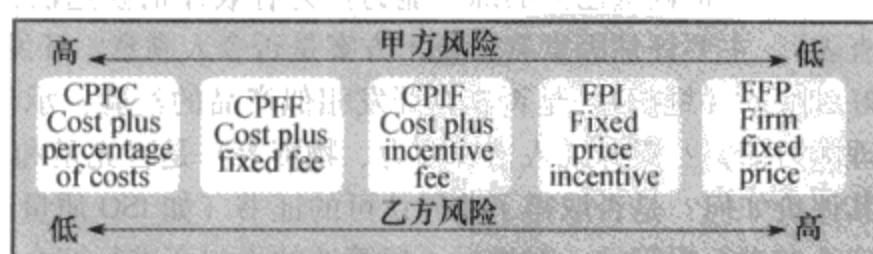


图 9-4 合同类型的风险比较

### 9.2.3 合同计划

当企业需要采购，例如需要将软件项目委托开发或者外包，企业需要从最有利于项目工期、成本、质量的角度出发来制定采购或者合同计划，计划中需要明确如何进行、委托什么项目、何时进行、费用如何等，其中，“资金”常常是一个约束条件。所以，应该对采购的产品或者委托的项目进行描述，选择需要的合同类型，同时，应该考虑市场条件，其他计划工作成果，例如供应商、承包商的情况，以及成本、工期、质量、人员等情况，采用的招标方式、合同形式以及招标的评估标准等。这个过程涉及到复杂的合同管理的相关知识，可以听取相关专家的意见等。这个阶段更多是通过招标书或者类

似招标书的形式体现，详见第一章的甲方合同环境下的招标书定义。

潜在的卖（乙）方获招标书后，被邀请参加竞标，卖（乙）方如果认为可以参与竞标，就可以组织编写投标书并提交投标书，然后参加竞标。

### 9.3 软件外包

软件项目外包其实质是软件开发过程从企业内部部分或全部延伸到外部的管理规范与管理技术。与内部实施相比，管理难度有过之而无不及。

项目外包有很多优点，但最主要的一点也许并不是为了降低开发成本，而是为了解决企业内部人力资源的限制，使得企业不用招聘新员工就可以开发大型项目。外包费用是一次性的营运开支，不像雇员薪资这样成为企业的长期营运成本。假如企业有些一次性的大型项目需要马上启动，但缺乏足够的资源，或者企业本身没有相应的技术人员来执行的时候，外包不失为一个可行的解决办法。

欧美企业愿意向印度、爱尔兰、中国等软件生产“蓝领”国家进行软件外包时，并非意味着它们不能开发，而是他们不开发，原因很简单，就是节省成本和控制质量。此类外包的发包方位处强势，全程可控，也形成了严格而规范的流程。而回首国内企业的软件工程外包，背景就复杂得多。

基于软件外包管理的复杂性，要确保软件外包的主要目标实现，必须合理地设计与外包相关的组织结构。目前，部分外包项目管理失控现象的主要原因是企业在软件项目外包管理方面职责不是很清晰，外包的策划、承包商的选择、监理执行等过程中协同工作出现问题。

参考CMM理念，结合国内软件工程特点，外包（或外包为主）项目还是采用外包管理部门领导下的项目监理负责制比较合适。类似内部实施项目的业务主管部门领导下的项目经理负责制，外包管理部门负责根据项目需求，定义外包需求、策划外包承包商的选择、外包合同的拟定、推荐并管理各外包项目监理、执行外包合同、监控项目进展、积累外包获取的财富、量化评估外包承包商的业绩、维护外包承包商关系记录等。

各项目的高级管理人员根据项目特点，审批决策外包项目需要实现的主要目标、任命评审项目监理、确定监理策略、选定外包承包商、外包合同批准与中止等。

一般情况下，软件外包的基本步骤如下：

- 1) 竞标邀请。首先由外包管理小组负责人起草“项目竞标邀请书”，然后与候选乙方建立联系，分发“外包项目竞标邀请书”及相关材料。然后，候选乙方与委托方有关人员进行交流，进一步了解外包项目，撰写投标书，并将投标书及相关材料交付给项目合同管理小组负责人（用于证明自身能力）。投标书的主要内容有：技术解决方案、开发计划、维护计划、报价等。

- 2) 评估候选乙方的综合能力。为有效评估候选乙方的综合能力，合同管理小组应制定“评估检查表”，主要评估因素有：技术方案是否令人满意？开发进度是否可以接受？性价比如何？能否提供较好的服务（维护）？是否具有开发相似产品的经验？承包商以前开发的产品质量如何？开发能力与管理能力如何？资源（人力、财力、物资等）是否充足和稳定？信誉如何？地理位置是否合适？外界对其评价如何？是否取得了业界认可的证书（如ISO质量认证、CMM2级以上认证）？项目管理小组对候选乙方进行粗筛选，对通过了粗筛选的承包商进行综合评估。

项目合同管理小组要和候选乙方进行多方面的交流，依据“评估检查表”评估候选承包商的综合能力，将评估结论记录在承包商的“能力评估报告”中。

- 3) 确定承包商。项目管理小组给出候选承包商的综合竞争力排名，并逐一分析与其建立外包合同的风险，选出最合适的承包商。将结论记录在“承包商能力评估报告”中。甲方确定最终选择的乙方名单。

选择适合的外包商，并不能单以价格来做最终决定。优质的服务需要付出较高的代价。企业应根据自身对软件质量的要求来决定服务的代价。按照国际企业的衡量指标，外包投入比本身开发的净投资多付15%~20%。也就是说，如果企业本身开发需要30万元的话，那么合理的外包服务价格大概是34万元到36万元。

一些项目经理往往认为外包开发项目与企业内部开发项目的管理没有多大分别，唯一不同是外包

项目需要更多时间去沟通、协调、跟进和监控。总体来说，这种想法是对的，但事实上外包项目的管理比企业内部开发项目的管理更复杂，担负更大的风险，需要更紧密的进度和质量监控。所以，外包项目尤其需要在如下方面做好工作：

1) 保障沟通。内部开发项目所需人力资源大致分为两组：一组是技术人员，另一组是配合技术人员的业务人员（他们是所建信息系统的潜在用户）。外包项目除了需要部分技术人员和用户群体参与外，更增加了一组外包商的资源。有些外包商更会指派一名联络人员负责联系与协调，而他们的技术人员只在后方负责项目的开发。这种运作模式要尽量避免，因为外包商指派负责联系的人员往往是业务人员的背景，对技术的细节不能全面把握，把有关信息传达到技术人员的时候便会有所差异。所以我们的首要任务是让外包商明白负责项目联系的人员必须是开发小组的主管。这名开发小组主管是直接参与开发项目的主要人员，如此才能够有效地进行沟通和监控。

2) 做好计划。项目经理首先需要做出一个详细的、完整的项目计划，并在计划中详细地列清楚每一件工作需要哪方面的哪些人力来共同执行。在计划中的每一个进度都需要进行确认才能继续。例如，外包商在完成系统分析后，需要把分析的结果让客户理解，好让企业能够确认外包商对整个系统的理解和分析与企业本身对项目的需求和分析达成一致，这样才能让外包商进行其后的模块设计。不然设计出来的模块组合便有可能与企业的需求不太一样，存在质量和最后上的差异。这些差异也将会引发企业将来在系统维护、更新、增加功能模块、升级、集成等各方面的严重问题。

3) 避免延误。要避免项目发生延误，计划中要预留足够的时间来进行确认工作。由于双方工作地点的缘故，原本只需一天的确认会议便可能耗费两天或三天的时间来完成。议程中所达到的共识也可能需要时间来让外包商做出适当的修改才能让企业正式确认。也只能在正式确认后才能够进一步继续接下来的工作。如果没有预留足够的时间用于协商，当一个项目经过七八个确认会议之后，也许已经延误了一个月的时间。

## 9.4 校务通系统案例分析

校务通信息系统作为项目的参与双方都存在合同计划。

- 1) 甲方：在项目的立项阶段，通过评估项目，决定这个项目不自行开发（不自造），需要委托公司完成（外购外包），为此需要规划合同事项，见第2章的案例分析。其合同类型是固定价格合同类型。
- 2) 乙方：作为项目的开发方，通过项目的范围分析，确定项目范围中的“网上考试”、“资源库系统”、“教师备课系统”等模块需要外购外包，所以，需要一个合同计划。见下面的校务通系统的外包招标书，其合同类型也是固定价格合同类型。

招 标 文 件
招标编号：XUT-080006
项目名称：校务通部分软件采购
北京 XXX 技术有限公司
二〇〇三年四月

## 目录

第一章	投标邀请	.....
第二章	投标人须知及前附表	.....
第三章	合同专用条款	.....
第四章	合同通用条款	.....
第五章	合同格式	.....
第六章	货物技术规格、参数与要求	.....
第七章	附件（投标文件格式）	.....
第八章	评标标准	.....

## 第一章 投标邀请

日 期：2003 年 4 月 30 日

招标编号：XWT-080006

北京 XXX 技术有限公司对下列软件产品和服务进行国内公开招标，现邀请合格的投标人参加投标。

招标货物名称、数量、交货时间及交货地点

包号	货物（软件）名称	货物需求	交货期	交货地点
01	网上考试系统	见第六章	见第七章	北京 XXX 技术有限公司
02	教学资源库系统			
03	教师备课系统			

投标人可对上述软件的每一个包进行完整投标。

投标文件递交地点：开标日期前递交的投标文件送至北京 XXX 技术有限公司，开标日期当天递交的投标文件直接送达开标地点；届时请参加投标的单位派代表出席开标仪式。

开标地点：北京市中关村 11 号 北京 XXX 技术有限公司办公楼 6 楼会议室。

地 址：北京市中关村 11 号 北京 XXX 技术有限公司办公楼 313 室

联系人：韩一 丁二 邮 编：100081

电 话：010-88255520 传 真：010-68255888

电子信箱：wsks@sina.com.cn

账户名称：北京 XXX 技术有限公司

开 户 行：建行中关村分行

账 号：1100101888888261068

其他详细内容暂略。

## 9.5 小结

本章讲述合同、合同类型，当企业存在软件外购、软件外包等需求时，就可能存在软件合同计划，合同可以保证项目的有效实施。

## 9.6 练习题

### 一、选择题

- 某项目是 CPFF 的合同类型，甲方项目经理刚刚得知乙方增加了成本，项目经理首先应该确定是否（ ）。
  - 有足够的储备处理这个变更，增加风险意识

- B. 其他的乙方可以按照原来的成本完成项目
  - C. 其他的项目能够节约预算
  - D. 任务是关键的项目任务
2. 可以在（ ）阶段采用 Make-or-buy 决策分析。
- A. 乙方选择
  - B. 立项（采购计划编制）
  - C. 合同管理
  - D. 成本计划
3. 合同激励可以使（ ）。
- A. 乙方节约成本
  - B. 甲方节约成本
  - C. 增加乙方成本
  - D. 合同双方的目标和利益得到协调
4. 甲乙双方之间存在的法律合同关系称为（ ）。
- A. 合同条款
  - B. 合约
  - C. 合同当事人
  - D. 其他
5. 哪种合同类型，乙方承担的风险最大？（ ）
- A. 成本加成本百分比
  - B. 成本加固定费
  - C. 成本加奖金
  - D. 固定价格
6. 下面哪项与计划签署合同没有关系？（ ）
- A. 设计模板
  - B. 协议附件
  - C. 招标文件
  - D. 评估标准
7. 合同一旦签署了就具有法律约束力，除非（ ）。
- A. 一方不愿意履行义务
  - B. 合同违反了法律
  - C. 一方宣布合同无效
  - D. 一方没有能力负担财务结果

## 二、判断题

- 1. 如果一个项目的合同类型是固定价格（FFP），合同价格是 100 万，实际花费 160 万，则项目结算金额为 160 万。（ ）
- 2. 成本加奖金 CPIF(Cost Plus Incentive Fee) 合同具有激励机制。（ ）
- 3. 采购是为了执行项目而从项目团队外部采购或者获取产品、服务或者结果的过程。（ ）

# 第 10 章

## ■ 软件配置管理计划

软件项目进行过程中面临的一个主要问题是持续不断的变化，变化是多方面的，例如版本的升级，不同阶段的产品变化。而配置管理是有效管理变化的重要手段。本章进入本书路线图的配置管理计划，见图 10-1。

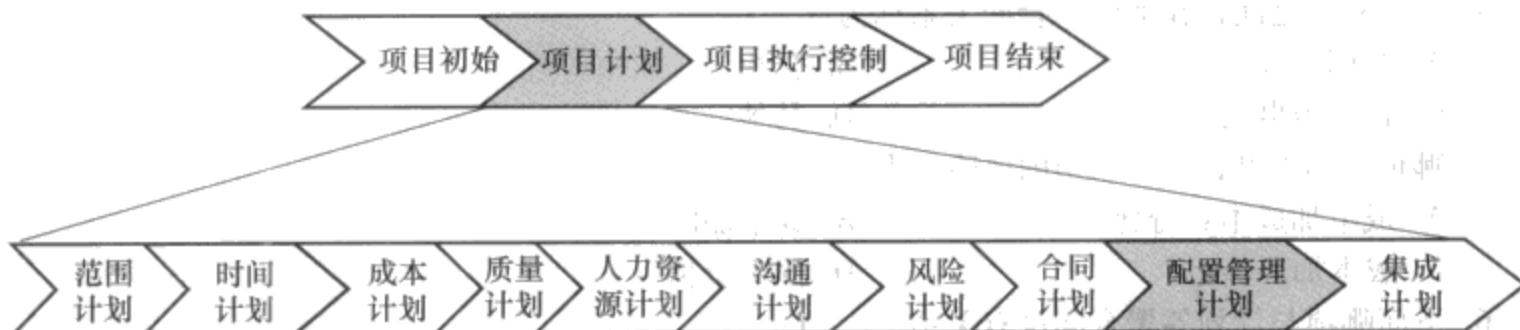


图 10-1 路线图——项目计划-配置管理计划

### 10.1 概述

软件项目的开发和实施往往都是在“变化”中进行的。可以毫不夸张地说，软件项目的变化是持续的、永恒的。需求会变，技术会变，系统架构会变，代码会变，甚至连环境都会变。有效的项目管理能够控制变化，以最有效的手段应对变化，不断命中移动的目标，无效的项目管理则被变化所控制。如何在受控的方式下引入变更、监控变更的执行、检验变更的结果、最终确认变更，并使变更具有追溯性，这一系列问题直接影响项目的成败，而配置管理可以应对这一系列问题。

配置管理的概念源于美国空军，为了规范设备的设计与制造，美国空军 1962 年制定并发布了第一个配置管理的标准《AFSCM375-1, CM During the Development & Acquisition Phases》。而软件配置管理概念的提出则在 20 世纪 60 年代末 70 年代初。当时加利福利亚大学圣巴巴拉分校的 Leon Presser 教授在承担美国海军的航空发动机研制合同时期，撰写了一篇名为《Change and Configuration Control》的论文，提出控制变更和配置的概念，这篇论文同时也是他在管理该项目的一个经验总结。

随着软件工程的发展，软件配置管理越来越成熟，从最初的仅仅实现版本控制，发展到现在的提供工作空间管理、并行开发支持、过程管理、权限控制、变更管理等一系列全面的管理能力，已经形成了一个完整的理论体系。同时在软件配置管理的工具方面，也出现了大批的产品，如：最著名的 ClearCase、开源产品 CVS、入门级工具 Microsoft VSS、新秀 Hansky Firefly 等。

在国外已经有 30 多年软件配置管理的历史，它随着软件产业的逐渐壮大，在国外成熟的软件企业中逐渐得到重视和普及。国内的软件配置管理也已经取得了迅速发展，并得到了软件公司的普遍认可。

#### 10.1.1 配置管理定义

软件配置管理，缩写为 SCM (Software Configuration Management)，是一套规范、高效的软件开发管理方法，同时也是提高软件质量的重要手段，它帮助开发团队对软件开发过程进行有效的变更控制，高效地开发高质量的软件。配置管理的使用取决于项目规模和复杂性以及风险水平。

软件配置管理是一套管理软件开发和维护以及其中各种中间软件产品的方法和规则，配置管理通过在特定的时刻选择软件配置，系统地控制对配置的修改，并在整个软件生命周期中维护配置的完整性和可追踪性。中间软件产品和用于创建中间软件产品的控制信息都应处于配置管理的控制下。

随着软件开发规模的不断扩大，一个项目的中间软件产品的数目也越来越多，中间软件产品之间的

关系也越来越复杂，对中间软件产品的管理也越来越困难，有效的软件配置管理则有助于解决这一问题。

软件配置管理贯穿于软件生存期的全过程，目的是用于建立和维护软件产品的完整性和可追溯性。

### 10.1.2 配置管理的目标

配置管理在系统周期中对一个系统中的配置项进行标识和定义的过程，这个过程是通过控制某个配置项及其后续变更，通过记录并报告配置项的状态以及变更要求，证明配置项的完整性和正确性实现的。配置管理相当于软件开发的位置管理，它回答了下面的问题：

- 我（他）是谁？（Who am I?）
- 为什么我（他）在这里？（Why am I here?）
- 为什么我（他）是某某？（Why am I who I am?）
- 我（他）属于哪里？（Where do I belong?）

软件配置管理的基本目标包括：

- 软件配置管理的各项工作是有计划进行的。
- 被选择的项目产品得到识别，控制并且可以被相关人员获取。
- 已识别出的项目产品的更改得到控制。
- 使相关组别和个人及时了解软件基线的状态和内容。

### 10.1.3 配置管理在软件开发中的作用

软件配置管理在软件项目管理中有着重要的地位，软件配置管理工作是以整个软件流程的改进为目标，是为软件项目管理和软件工程的其他领域打好基础，以便于稳步推进整个软件企业的能力成熟度。软件配置管理的主要思想和具体内容在于版本控制。版本控制是软件配置管理的基本要求，是指对软件开发过程中各种程序代码、配置文件及说明文档等文件变化的管理。版本控制最主要的功能就是追踪文件的变更。它将什么时候、什么人更改了文件的什么内容等信息忠实地记录下来。每一次文件的改变，文件的版本号都将增加，比如 V1.0、V1.1、V2.1 等。它可以保证任何时刻恢复任何一个配置项的任何一个版本。版本控制还记录了每个配置项的发展历史，这样就保证了版本之间的可追踪性，也为查找错误提供了帮助。除了记录版本变更外，版本控制的另一个重要功能是并行开发。软件开发往往是多人协同作战，版本控制可以有效地解决版本的同步以及不同开发者之间的开发通信问题，提高协同开发的效率。许多人将软件的版本控制和软件配置管理等同起来，这是非常错误的观念。版本控制虽然在软件配置管理中占据非常重要的地位，但这并不是它的全部，对开发者工作空间的管理等都是软件配置管理不可分割、不可或缺的部分。而且，简单地使用版本控制，并不能解决开发管理中的深层问题。软件配置管理给开发者带来的好处是显而易见的，但对于项目管理者来说，他所关心的角度与开发者是不一样的，他更关注项目的进展情况，这不是简单的版本控制能够解决的。项目管理者从管理者的角度去运用软件配置管理中的各种记录数据，将有巨大的收获。从这些记录数据中，我们可以了解到谁在什么时候改了些什么、为什么改；我们可以了解到开发项目进展得如何、完成了多少工作量；我们可以了解到开发工程师的资源是否充分使用、工作是否平衡等。

现在人们逐渐认识到，软件配置管理是软件项目管理中的一种非常有效和现实的技术，它能非常有效地适应软件开发需求。配置管理对软件产品质量和软件开发过程的顺利进行和可靠性有着重的意义。图 10-2 说明了配置管理在软件开发过程中的作用，从图

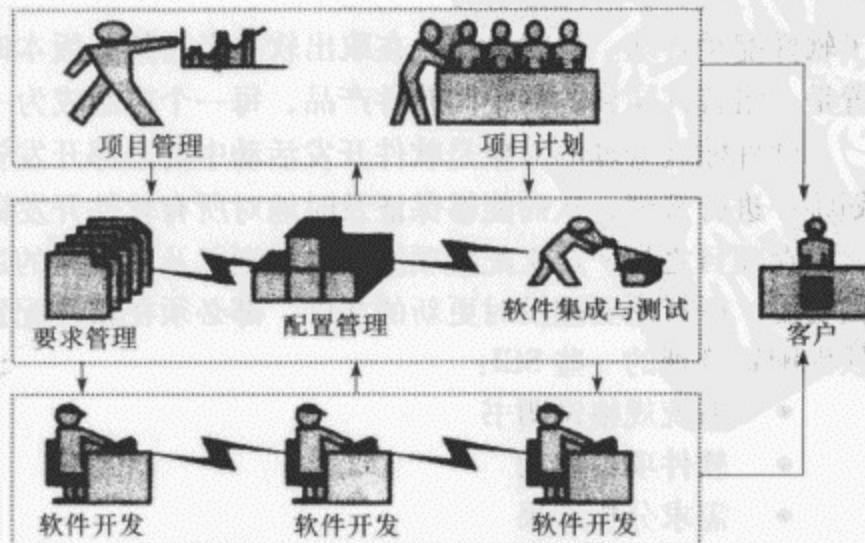


图 10-2 配置管理在软件项目中的作用

中可以看出配置管理相当于软件开发生产线中的仓库和调度。

合理的实施配置管理，软件产品的质量会得到了提高、开发团队能够工作在一个有助于提高整体工作效率的配置管理平台上。如果没有很好的进行配置管理，将会影响成本、进度和产品的规格，没有变更管理，项目就会无限放大。有效的配置管理可以不断命中移动的目标。配置管理是对工作成果的一种有效保护。

软件配置管理是软件开发环境管理部分的核心，有些管理功能（比如过程管理）在最初并不属于软件配置管理，但随着软件配置管理的不断发展，也逐渐成为了软件配置管理的一部分。

## 10.2 配置管理的相关概念

配置管理是软件项目能顺利进行的基础。一个软件项目开发过程中，会有大量的“中间产品”产生，典型的如代码、技术文档、产品文档、管理文档、数据、脚本、执行文件、安装文件、配置文件，甚至一些参数等，这些中间成果都是项目的产品。而且，不断变化的软件项目还会使这些产品产生多个不同的版本，可以想像，一旦配置管理失控，项目组成员就会陷入配置项的“泥潭”。很显然，制定配置管理计划、建立配置管理系统、确定配置管理的流程和规程、严格按照配置管理流程来处理所有配置项，是确保配置管理顺利实现的方法和必要的手段。

软件配置管理包括标识在给定时间点上软件的配置（即选定的软件工作产品及其描述），系统地控制对配置的更改并维护在整个软件生存周期中配置的完整性和可追溯性。置于软件配置管理之下的工作产品包括交付给顾客的软件产品（例如软件需求文档和代码），以及与这些软件产品等同的产品项或生成这些软件产品所要求的产品项（例如编译程序），在配置管理过程中需要建立一个软件基线库，当软件基线形成时就将它们纳入该库。通过软件配置管理的变更控制和配置审计功能，系统地控制基线的更改和那些利用软件基线库构造成的软件产品的发行。

在介绍配置管理过程之前，先介绍一下配置管理中的几个重要概念：配置项、基线以及软件配置控制委员会。

### 10.2.1 配置项

软件配置项也称 SCI (Software Configuration Item) 是项目定义其受控于软件配置管理的项。一个软件配置项是一个特定的、可文档化的工作产品集，这些工作产品是生存期中产生或者使用的。Pressman 对于 SCI 给出了一个比较简单的定义：“软件过程的输出信息可以分为三个主要类别：1) 计算机程序（源代码和可执行程序）。2) 描述计算机程序的文档（针对技术开发者和用户）。3) 数据（包含在程序内部或外部）。这些项包含了所有在软件过程中产生的信息，总称为软件配置项。”

每个项目的配置项也许会不同。所有源代码，包括构成文件，及其他所有制作软件产品所需的文件，都可能在软件配置管理控制之下。软件产品某一特定版本的源代码及其相关的工具，都可能受控于软件配置管理。也就是说，在取出软件产品某一版本时，同时也可以取出与此版本相关的工具。配置是一组有共同目的的中间软件产品，每一个产品成为一个配置项。

软件配置管理的对象是软件开发活动中的全部开发资产。所有这一切都应作为配置项纳入管理计划统一进行管理，从而能够保证及时地对所有软件开发资源进行维护和集成。

在项目之初，定义配置项的命名规则以及配置项的逻辑组织结构，在项目进行当中，以什么规则做变更。所有需要被及时更新的文件，都必须在软件配置管理控制之下。例如，下面的列表可以作为软件配置管理的一些 SCI：

- 系统规格说明书
- 软件项目规划
- 需求分析结果
  - 软件需求规格说明书
  - 系统原型

- 设计规格说明书
  - 数据设计描述
  - 总体结构设计描述
  - 模块设计描述
  - 界面设计描述
  - 对象描述（若采用面向对象技术）
- 源代码清单
- 测试规格说明书
  - 测试计划和过程
  - 测试用例与实验结果
- 操作和安装手册
- 可执行程序
  - 每个模块的可执行代码
  - 链接到一起的代码
- 用户手册
- 维护文档

除此之外，有时把软件工具和中间产生的文件也列入配置管理的范畴，即把软件开发中选用的编辑器、编译器和其他一些 CASE 工具固定地作为软件配置的一部分，当配置项发生变化时，应该考虑这些工具是否与之适应和匹配。

配置项也有不同的版本，这里类似地将面向对象的类和实例类比成配置项和配置项的版本。配置项可以看成是面向对象的类，版本可以看成是类的实例。图 10-3 表示的需求规格配置项，需求规格的不同版本类似于需求规格配置项的实例，配置项的不同版本是从最原始的配置项（类似于配置项类）演变出的不同情况，尽管每个都是不同的，但是它们具有相关性。

由此可见，配置项的识别是配置管理活动的基础，也是制定配置管理计划的重要内容。所有配置项都应按照相关规定统一编号，按照相应的标准生成。在引入软件配置管理工具进行管理后，这些配置项都应以一定的目录结构保存在配置库中。

### 10.2.2 基线

软件的开发过程是一个不断变化着的过程，由于各种原因，可能需要变动需求、预算、进度和设计方案等，尽管这些变动请求中绝大部分是合理的，但在不同的时机作不同的变动，难易程度和造成的影响差别甚大。为了有效地控制变动和在不严重阻碍合理变化的情况下控制变化，软件配置管理引入了“基线”（base line）这一概念。

基线是一个或者多个配置项的集合，它们的内容和状态已经通过技术的复审，并在生存期的某一阶段被接受了。对配置项复审的目标是验证它们被接受之前的正确性和完整性，一旦配置项经过复审，并正式成为一个初始基线，那么该基线就可以作为项目生存期开发活动的起始点。

IEEE 对基线的定义是这样的：“已经正式通过复审和批准的某规约或产品，它因此可作为进一步开发的基础，并且只能通过正式的变化控制过程改变。”所以，根据这个定义，我们在软件的开发流程中把所有需加以控制的配置项分为基线配置项和非基线配置项两类。例如：基线配置项可能包括所有的设计文档和源程序等；非基线配置项可能包括项目的各类计划和报告等。

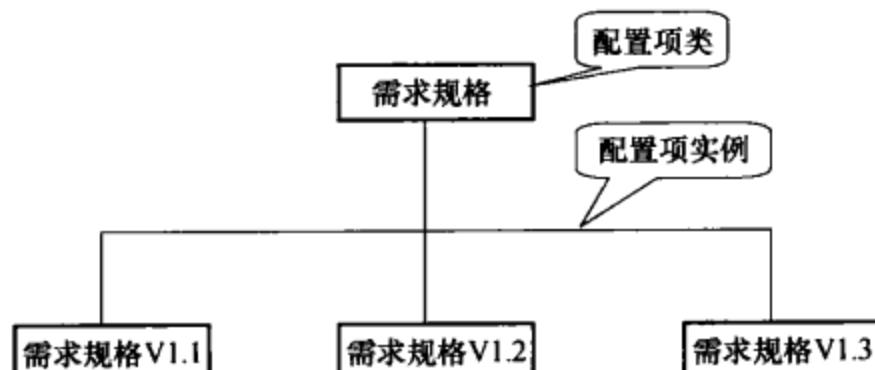


图 10-3 配置项类及实例

基线代表了软件开发过程的各个里程碑，它标志开发过程中一个阶段的结束。对于已成为基线的配置项，虽然可以修改，但必须按照一个特殊的、正式的过程进行评估，确认每一处修改。相反，对于未成为基线的配置项，可以进行非正式修改。在开发过程中，我们在不同阶段要建立各种基线。所以说，基线是具有里程碑意义的一个配置。

虽然基线可在任何级别上定义，但一般最常用的软件基线如图 10-4 所示。基线提供了软件生存期中各个开发阶段的一个特定点，其作用是把开发阶段工作的划分更加明确化，使本来连续的工作在这些点上断开，以便于检查与肯定阶段成果。在交付项中确定一个一致的子集，作为软件配置基线，这些版本一般不是同一时间产生的，但具有在开发的某一特定步骤上相互一致的性质，例如系统的一致、状态的一致。基线可以作为一个检查点，正式发行的系统必须是经过控制的基线产品。

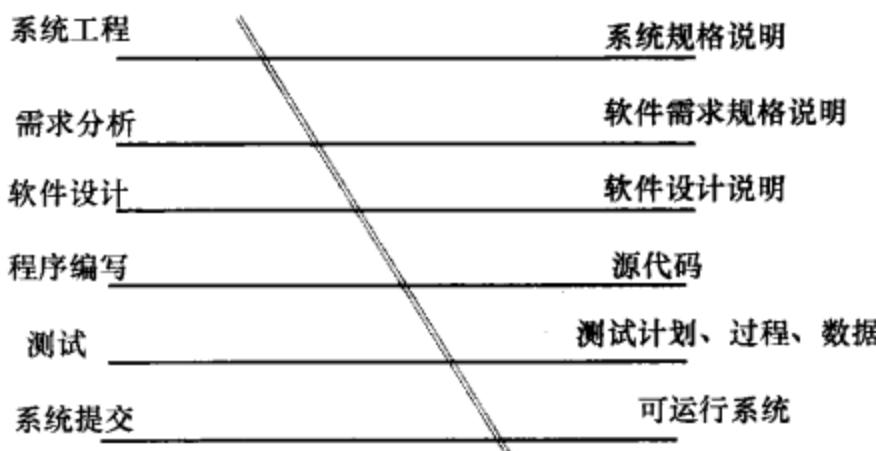


图 10-4 基线图示

### 10.2.3 配置控制委员会

配置管理的目标之一是有序、及时和正确地处理对软件配置项的变更，而实现这一目标的基本机制是通过配置控制委员会的有效管理。配置控制委员会（Software Configuration Control Board, SCCB）可以是一个人也可以是一个小组，基本是由项目经理及其相关人员组成的。对于一个新的变更请求，所执行的第一个动作是依据配置项和基线，将相关的配置项分配给适当的 SCCB，SCCB 根据技术的、逻辑的、策略的、经济的和组织的角度，以及基线的层次等，对变更的影响进行评估，SCCB 将一个变更的期望与它对项目进度、预算的影响进行比较，SCCB 的一个目标是保持一种全局观点，评估基线的变更对项目的影响，并决定是否变更。SCCB 承担变更控制的所有责任，具体责任如下：

- 评估变更。
- 批准变更申请。
- 在生存期内规范变更申请流程。
- 对变更进行反馈。
- 与项目管理层沟通。

一个项目可能存在多个 SCCB，它们可能有不同的权利责任。即使一个项目只有一个 SCCB，但是一个组织也可能存在多个 SCCB，不同的项目具有不同的 SCCB 定义。

## 10.3 配置管理过程

软件配置管理可以唯一地标识每个软件项的版本，控制由两个或多个独立工作的人员同时对一个给定软件项的更新，按要求在一个或多个位置对复杂产品的更新进行协调，标识并跟踪所有的措施和更改，这些措施和更改是由于更改请求或问题引起的。

配置管理主要包括配置项标识、变更控制、配置项状态统计和配置项审计等活动。配置项标识是识别产品的结构、产品的构件及其类型，为其分配唯一的标识符，并以某种形式提供对它们的存取，同时找出需要跟踪管理的项目中间产品，使其处于配置管理的控制之下，并维护它们之间的关系。变更控制是记录变化的有关信息，控制软件产品的发布和在整个软件生存期中对软件产品的修改。有效控制变更，可以保证软件产品的质量，例如，它将解决哪些修改会在该产品的最新版本中实现的问题。配置项状态统计是记录并报告配置项和修改请求的状态，并收集关于产品构件的重要统计信息。例如，它将解决修改这个错误会影响多少个文件的问题，以便报告整个软件变化的过程。配置项审计是利用配置项记录验证软件达到的预期结果，确认产品的完整性并维护构件间的一致性，

即确保产品是一个严格定义的构件集合。例如，它将解决目前发布的产品所用的文件的版本是否正确的问题。

配置管理基本是围绕着这四个活动进行的，为了更好地管理这四个活动，定义配置管理的基本过程如下：

- 1) 配置项标识、跟踪
- 2) 配置管理环境建立
- 3) 基线变更管理
- 4) 配置审计
- 5) 配置状态统计
- 6) 配置规划

### 10.3.1 配置项标识、跟踪

一个项目要生成很多的过程文件，并经历不同的阶段和版本。标识、跟踪配置项过程用于将软件项目中需要进行控制的部分拆分成配置项（SCI），建立相互间的对应关系，进行系统的跟踪和版本控制，以确保项目过程中的产品与需求相一致，最终可根据要求将配置项组合生成适用于不同应用环境的正确的软件产品版本。配置项应该被唯一地标识，同时应该定义软件配置项的表达约定，一个项目可能有一种或多种的配置项标识定义，例如文档类的、代码类的、工具类的等，或者统一一个规范定义。下面给出几个配置项标识的实例：

配置项的标识约定例1：

项目名称\_ 所属阶段\_ 产品名称\_ 版本标识

其中版本标识的约定如下：

- 1) 版本标识以“V”开头，版本号放后。
- 2) 版本号分3节：主版本号、次版本号和内部版本号。每小节以小数点（.）间隔。例如：School\_Design\_HLD\_V2.1.1 表示的配置项是名称为 School 的项目，在设计（Design）

阶段的总体设计（HLD）的 V2.1.1 版本

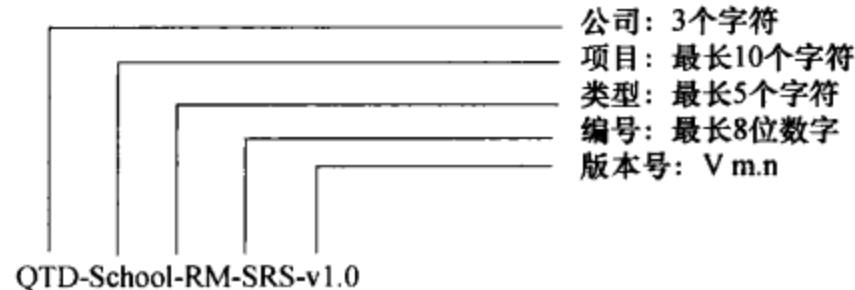


图 10-5 配置项标识说明

配置项的标识约定例2（如图 10-5 所示）。

通常，一个配置项与其他的配置项存在一定的关系，跟踪配置项之间的关系是很重要的。图 10-6 是需求规格配置项和系统测试用例配置项的跟踪关系。

版本控制是软件配置管理的核心功能。所有置于配置库中的元素都应自动予以版本的标识，并保证版本命名的唯一性。版本在生成过程中，自动依照设定的使用模型自动分支、演进。除了系统自动记录的版本信息以外，为了配合软件开发流程的各个阶段，我们还需要定义、收集一些元数据（metadata）来记录版本的辅助信息和规范开发流程，并为今后对软件过程的度量做好准备。当然如果选用的工具支持的话，这些辅助数据将能直接统计出过程数据，从而方便我们软件过程改进（Software Process Improvement, SPI）活动的进行。

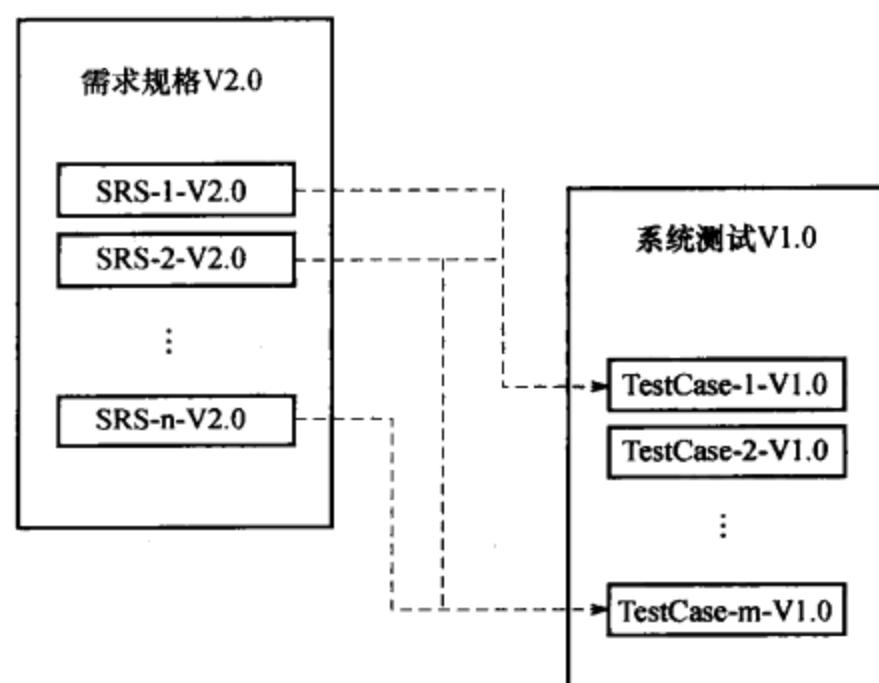


图 10-6 配置项跟踪

### 10.3.2 配置管理环境建立

在引入了软件配置管理工具之后，所有开发人员都会被要求把工作成果存放到由软件配置管理工具所管理的配置库中去，或是直接工作在软件配置管理工具提供的环境之下。所以为了让每个开发人员和各个开发团队能更好的分工合作，同时又互不干扰，对工作空间的管理和维护也成为了软件配置管理的一个重要的活动。

配置管理维护了配置项的发展史，在整个软件产品的生存期内，配置项的每次变更都会被配置管理系统忠实的记录下来，形成不同的版本。同时它还是并行开发得以实现的基础。版本控制的目的是按照一定的规则保存配置项的所有版本，避免发生版本丢失或混淆等现象，并且可以快速准确地查找到配置项的任何版本。

一般来说，比较理想的情况是把整个配置库视为一个统一的工作空间，然后再根据需要把它划分为个人（私有）、团队（集成）和全组（公共）这三类工作空间（分支），从而更好的支持将来可能出现的并行开发的需求。

每个开发人员按照任务的要求，在不同的开发阶段，工作在不同的工作空间上，例如：对于私有开发空间而言，开发人员根据任务分工获得对相应配置项的操作许可之后，他即在自己的私有开发分支上工作，他的所有工作成果体现为在该配置项的私有分支上的版本的推进，除该开发人员外，其他人员均无权操作该私有空间中的元素；而集成分支对应的是开发团队的公共空间，该开发团队拥有对该集成分支的读写权限，而其他成员只有只读权限；至于公共工作空间，则是用于统一存放各个开发团队的阶段性工作成果，它提供全组统一的标准版本。

当然，由于选用的软件配置管理工具的不同，在对于工作空间的配置和维护的实现上有比较大的差异。

配置管理环境是为了更好的进行软件配置管理的系统环境。其中最重要是建立配置管理库，简称配置库。

软件配置管理库是用来存储所有基线配置项及相关文件等内容的系统，是在软件产品的整个生存期中建立和维护软件产品完整性的主要手段。配置库存储包括配置项相应版本、修改请求、变化记录等内容，是所有配置项的集合和配置项修改状态记录的集合，是在软件产品的整个生存期中建立和维护软件产品完整性的主要手段。

从效果上来说，配置库是集中控制的文件库，并提供对库中所存储文件的版本控制。配置库中的文件是不会变的——它们不能被更改。任何更改被视为创建了一个新版本的文件。文件的所有配置管理信息和文件的内容都存储在配置库中。

配置管理库就是受控库，受控库的任何操作都要受到控制。如图 10-7 所示，从受控库导出（check out）的文件自动被锁定直到文件重新被导入（check in），一个版本号自动与新版本文件相关联。这样一来，用户可以随时根据特定的版本号来导出任何文件（缺省的是最新的版本）。对最新版本的修改的结果是产生一个新的，顺序递增的版本；而对更老版本的修改的结果是产生一个分支版本。配置库中不但存储了文件的不同版本，更改的理由，而且存储谁在什么时候替换了某个版本的文件等历史信息。请注意，有的配置管理库，对于每个不同版本文件，不是所有的代码都存储起来，而只是将不同版本间实际的差异存储起来，这称为增量。这种方法有利于节省空间和节省对最新文件版本的访问时间。另外，可以根据状态给文件加上标签，然后基于状态的值进行导出。它们同样也可以根据修订版本号，日期和作者进行导出操作。配置库捕捉配置管理信息并把不同版本的文件存储为不可修改的对象。

除了有效的版本管理之外，配置管理还能更好的解决并发控制的问题。并行开发中最常见的不同版本软件的错误（bug）修正问题，就可以通过版本控制中分支与合并的方法有效地解决。比如，软件已经发布了 1.0 版本，开发小组在为该软件添加新的功能，正在进行 2.0 版本的开发，如图 10-8 所示。而此时，如果 Release 1.0 中发现了错误，必须修正，我们就必须从 Release 1.0 中建立修正分支，进行

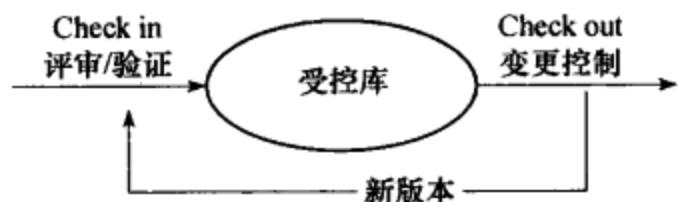


图 10-7 受控库环境

必要的修正后，发布修正版 Release 1.1，而这个版本的发布与 2.0 版本的开发没有直接关系。当 2.0 版本测试结束后，要与 1.0 版本中修正分支合并，从而发布 2.0 的版本。在这个并行开发过程中，创建分支和分支的合并起了非常重要的作用。

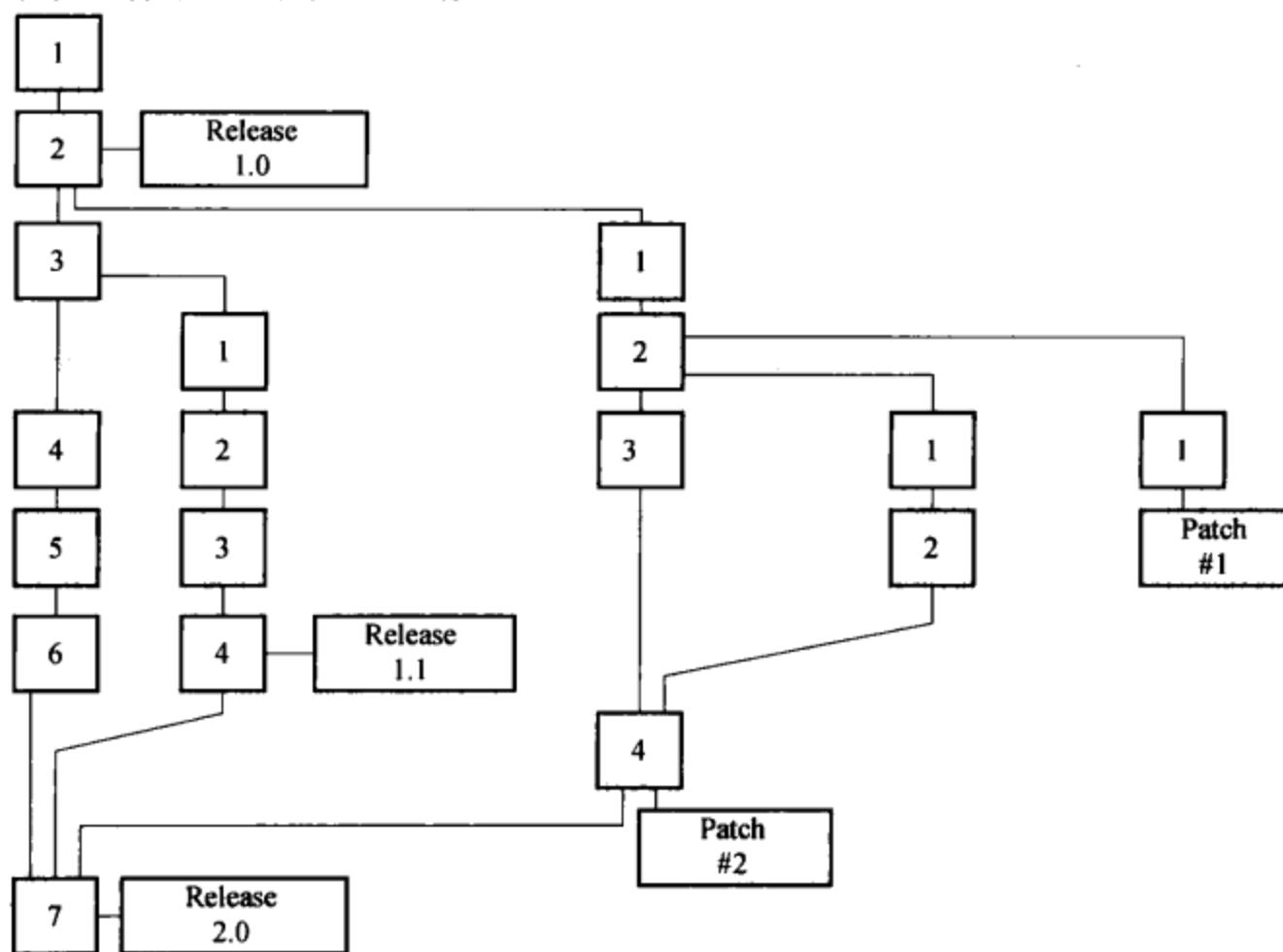


图 10-8 版本的分支结构

配置管理环境建立包括建立配置管理的硬件环境和软件环境，同时建立存储库的操作说明和操作权限。例如，图 10-9 是一个名为 Prj-B2B 项目的一个受控库和两个开发库（动态库）的目录结构：

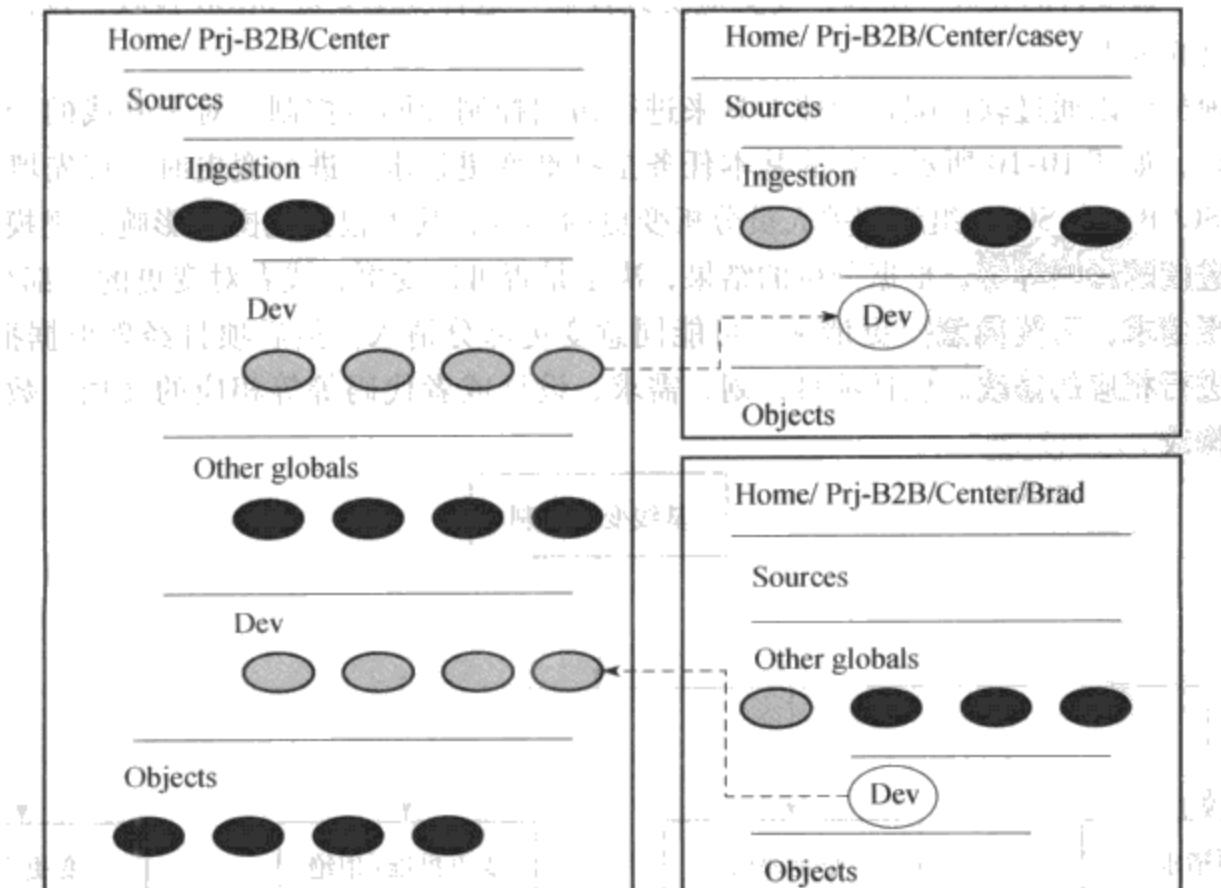


图 10-9 一个项目的受控库和开发库的目录结构

- 受控库按照系统架构设计的结构提供相应的子目录。
- 开发库（动态库）的结构映射了受控库的结构。
- 对于开发过程的各个子目录是按照架构设计的模块安排的，例如 Ingestion 是架构设计的一个模块。
- Prj-B2B 项目的受控库是/home/Prj-B2B/Center。其中包括 Sources 目录和 Object 目录，Sources 目录按照子系统的功能分为很多的子目录，例如：
 

```
/home/Prj-B2B/Center/Sources/ingestion
/home/Prj-B2B/Center/Sources/Other-globals
```

 Object 目录下保存了目标库，以便减少连接时间和执行程序的规模。
- 每一个开发人有一个动态的开发库（开发库），目录是/home/Prj-B2B/Center/username，结构如上所述，在受控库中建立对每个开发者的连接，开发者使用自己目录下的可写文件进行开发修改，同时将只读文件拷贝到自己的目录下以减少编译、连接的时间。

对于配置库中的各个基线控制项，应该根据其基线的位置和状态来设置相应的访问权限。一般来说，对于基线版本之前的各个版本都应处于被锁定的状态，如需要对它们进行变更，则应按照变更控制的流程来进行操作。

### 10.3.3 基线变更管理

基线是和变更控制紧密相连的。也就是说在对各个 SCI 做出了识别，并且利用工具对它们进行了版本管理之后，如何保证它们在复杂多变的开发过程中真正的处于受控的状态，并在任何情况下都能迅速的恢复到任一历史状态就成为了软件配置管理的另一重要任务。因此，基线变更管理就是通过结合人的规程和自动化工具，以提供一个变化控制的机制。

在项目开发过程中，项目的基线（的配置项）发生变更几乎是不可避免的，变更的原因很多：人们可能犯错误，客户的需求变更，产品的环境发生变更，人们开发了新的技术等等。变更包括需求、设计、实施、测试等所有开发过程以及相关的文件。变更如果没有得到很好的控制，就会产生很多的麻烦，以至于导致项目的失败。所以，变更应受到控制，这种变更要经 SCCB 授权，按照程序进行控制并记录修改的过程。

配置管理员可以通过软件配置管理工具来进行访问控制和同步控制。对于基线的变更需要指定变更控制流程，如图 10-10 所示，它的基本任务是批准变更请求，进行变更时，首先填写变更请求表，提交给 SCCB，由 SCCB 组织相关人员分析变更的影响，其中包括范围的影响、规模的影响、成本的影响、进度的影响等等，根据分析的结果，决定是否可以变更，或者对变更的一部分提出意见，可能拒绝变更请求，可能同意变更请求，可能同意变更部分请求。然后项目经理根据批准的结果，指导项目组进行相应的修改，包括项目计划、需求、设计或者代码等等相应的文档、数据、程序或者环境等的修改。

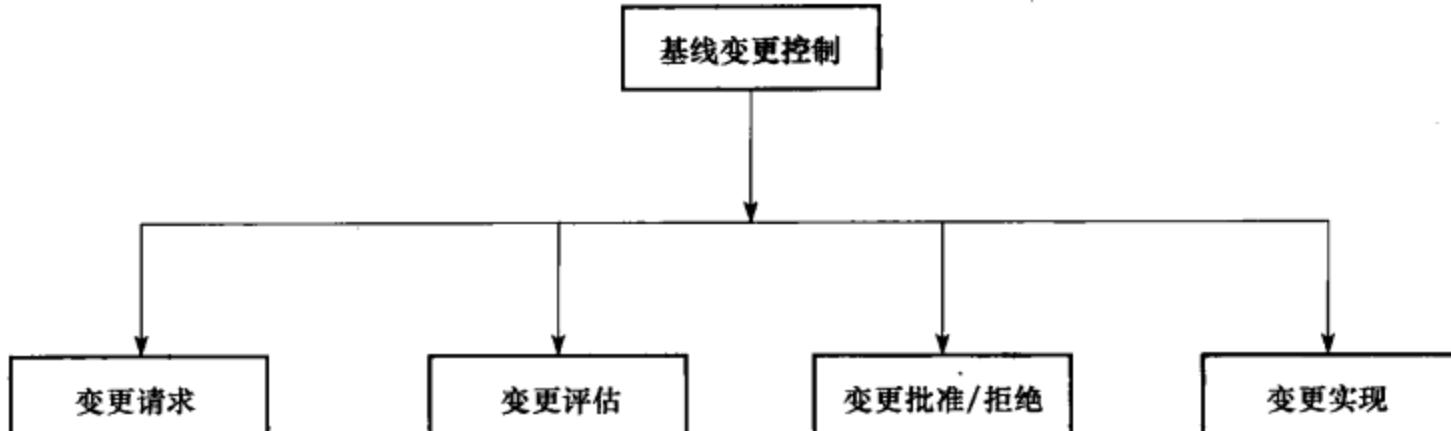


图 10-10 变更控制系统

有时我们称这样的流程中，为变更控制系统，之所以称为系统是因为在进行变更控制的时候，需要在综合运用各种系统，将分工、授权、控制有机的结合在一起，各司其职。一旦有问题，可以通过一定方式提出请求，一些小的变更可以自行决定，而一些大的纠正行为通过一系列的审批程序。所以，要注意内部结构和相互之间的关联，控制系统的流程必须是清晰、明确，否则会产生混乱。

### 1. 变更请求

变更请求是变更控制的起始点，变更请求很少来自配置管理活动本身，通常来自系统之外的事件触发。例如，需求变化、不符合项或者软件测试报告就是一些普遍的触发事件。他们可能组成大量的变更，对开发基线产生影响，并传播到开发基线中的配置项。变更请求需要准备一个软件变更请求的表单，如表10-1所示，这是一个正式的文档，变更申请者使用这一文档描述所标识的变更。

表10-1 项目变更申请

项目名称			
变更申请人		提交时间	
变更题目		紧急程度	
变更具体内容			

变更影响分析

表10-2 项目变更评估表

技术因素	技术因素	
运行	测试	
质量、可靠性	利益	
规模	紧急性	
选择余地	依赖性	
资源	接口影响	

进度影响

变更确认

处理结果	
签字	

预算影响

### 2. 变更评估

提交变更请求表单后，必须验证其完整性、正确性以及是否清晰，对变更申请进行评估。变更请求可能是由于提交人的错误理解而产生的，或者可能与现存的请求相重复，如果在检查中发现提出的变更是不完整的、无效的或者已经评估的，那么就应拒绝这一请求，并建立拒绝原因的文档，返回给提交变更请求的人，不论什么情况，都要保留该变更请求和相关的处理结果。一个有效的变更评估活动如图10-11所示。

变更评估过程的第一步，是对新的、有效的变更进行分类。任何分类模式的目的在于帮助执行配置管理计划给出的变更控制和配置过程中的后续步骤。在评估中可以回答表10-2中所列问题。

### 3. 变更的批准或者拒绝

批准或者拒绝软件变更请求中涉及的活动见图10-12。

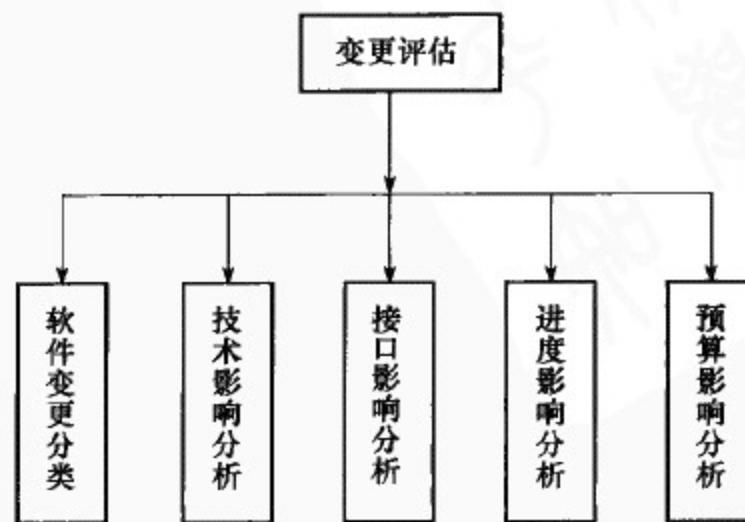


图10-11 变更请求的评估

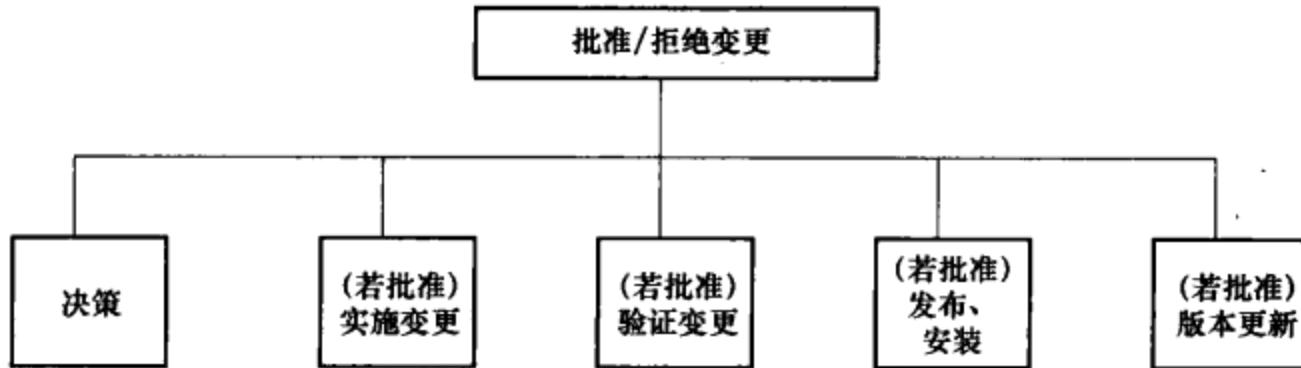


图 10-12 批准/拒绝变更的活动

根据变更评估的结果，SCCB 对变更请求做出决策。通常，决策包括：

- 直接实现变更。
- 挂起或者延迟变更。
- 拒绝变更。

对于最后一种情况，应该通知变更请求人，并且保存所有的相关记录，如果以后的事件证明拒绝变更是错误的，这些保存下来的记录是有用的。挂起或者延迟的变更常常是在 SCCB 评估分析之后，但是变更请求不在 SCCB 的控制范围之内的。

当变更被接受时，应该按照选择的进度实现变更，实现进度可以采用下面三种形式之一：

- 尽可能快的实现变更：期望的变更是修改开发基线中的一个配置项，只有解决了这个变更，其他的工作才能展开。
- 按照一个特定的日期实现变更：考虑项目内或者项目外的事件，确定合适的日期实现变更。
- 在另外的版本中实现，出于技术或者运行等原因，期望与另外的变更一起发布。

#### 4. 变更实现

实现已批准的软件变更请求中包含的活动如图 10-13 所示。项目人员使用 SCCB 给予的权限并遵循 SCCB 的指导，从受控库中取出基线的拷贝，并实现被批准的变更，对已经实现的变更实施验证。一旦 SCCB 认为正确实现并验证了一个变更，就可以将更新的基线放入配置库中，更新该基线的版本标识等。

在需求管理章节中谈到需求变更便是很重要的一个基线变更，软件需求变更表现在文档的需求变更和相关过程模型的变更。

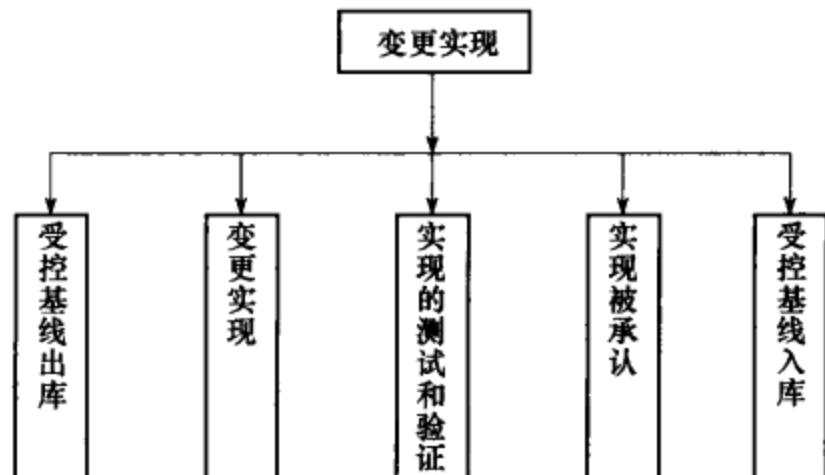


图 10-13 变更实现过程

#### 10.3.4 配置审计

配置审计的主要作用是作为变更控制的补充手段，来确保某一变更需求已被切实实现。在某些情况下，它被作为正式的技术复审的一部分。

配置审计包括两方面的内容：“配置管理活动审计”和“基线审计”。“配置管理活动审计”用于确保项目组成员的所有配置管理活动，遵循已批准的软件配置管理方针和规程，如导入/导出的频度、产品版本升级原则等。实施“基线审计”，要保证基线化软件工作产品的完整性和一致性，其目的是保证基线的配置项（SCI）正确地构造并正确地实现，并且满足其功能要求。基线的完整性可以下几个方面考虑：基线库是否包括所有计划纳入的配置项？基线库中配置项自身的内容是否完整？此外，对于代码，要根据代码清单检查是否所有源文件都已存在于基线库。同时，还要编译所有的源文件，检查是否可产生最终产品。一致性主要考察需求与设计以及设计与代码的一致关系，尤其在有变更发生时，要检查所有受影响的部分是否都做了相应的变更。审核发现的不符合项要进行记录，并跟踪直

到解决。

简单的配置管理活动审计是记录配置管理工具执行的所有命令，复杂的配置管理活动审计还包括记录每个配置项的状态变化。

当软件配置管理发布一个新版本时，可能需要审核一个构造记录，以确保这一构造中确实包含组件配置项的正确版本，或者复审变化历史数据库，以验证在新的发布中只有所期望的变更，验证配置系统是否保持了自身的完整性。通过基线审计可以发现系统中一直没有被处理的变化请求，或者发现那些不按照规程文档随意出现或者变更的软件项。

### 10.3.5 配置状态统计

由于软件配置管理覆盖了整个软件的开发过程，因此它是改进软件过程、提高过程能力成熟度的理想切入点。配置管理是贯穿整个项目生存期的，而且具有非常重要的作用，因此必须定期检测软件配置管理系统的运行情况，以及配置项本身的变更历史记录。检查配置管理系统以及内容，检测配置项变更历史的过程称为配置状态统计。这些过程的结果应以报告的形式给出。

配置系统状态统计的范围可以因项目而异。例如 IEEE 标准 828-1998 规定用于计算配置状态的最小数据集包括：

- 被批准的配置项。
- 配置项的所有请求的变化状态。
- 配置项所有被批准的变更实现状态。

配置状态报告是根据配置项操作数据库中的记录来向管理者报告软件开发活动的进展情况。这样的报告应该是定期进行，并尽量通过辅助工具自动生成，用数据库中的客观数据来真实的反映各配置项的情况。

配置状态报告应着重反映当前基线配置项的状态，以作为对开发进度报告的参照。同时也能根据开发人员对配置项的操作记录来对开发团队的工作关系作一定的分析。配置状态报告应该包括下列主要内容：

- 配置库结构和相关说明；
- 开发起始基线的构成；
- 当前基线位置及状态；
- 各基线配置项集成分支的情况；
- 各私有开发分支类型的分布情况；
- 关键元素的版本演进记录；
- 其他应予报告的事项。

此外，在评估一个配置系统状态以及系统所支持的产品状态时，经常需要以下信息：

- 变更请求的数量，当然可以按照类别进行分类，例如：需求变更、文档变更、设计变更、源码变更等。
- 变更请求的历史报告，包括从编写请求、请求复审、请求批准、请求实现、请求测试、请求接受等一系列活动所花费的时间和每个单项活动所花费的时间。
- 配置管理系统以及 SCCB 在运作中发生异常的次数等。

## 10.4 配置管理计划

软件配置管理计划过程就是确定软件配置管理的解决方案，软件配置管理的解决方案涉及面很广，将影响软件开发环境、软件过程模型、配置管理系统的使用者、软件产品的质量和用户的组织机构。

软件配置管理计划由配置管理者负责制订，它是软件配置管理规划过程的产品，并在整个软件项目开发过程中作为配置管理活动的依据进行使用和维护。首先由项目经理确定配置管理者，配置管理者通过参与项目规划过程，确定配置管理的策略，然后负责编写配置管理计划，配置管理计划是作为项目计划的一部分。

### 10.4.1 配置管理计划编制

配置管理的实施需要消耗一定的资源，在这方面一定要预先规划。具体来说，配置管理实施主要需要两方面的资源要素：一是人力资源，二是工具。

人力方面，因为配置管理是一个贯穿整个软件生存期的基础支持性活动，所以配置管理会涉及到团队中比较多的人员角色。比如，项目经理、配置管理员、配置控制委员会、开发人员、维护人员等。但是，工作在一个良好的配置管理平台上并不需要开发人员、测试人员等角色了解太多的配置管理知识，所以配置管理实施集中在配置管理者上。配置管理者是一个比较奇妙的角色，对于一个实施了配置管理、建立了配置管理工作平台的团队来说，他是非常重要的，整个开发团队的工作成果都在他的掌管之下，他负责管理和维护的配置管理系统。如果出现问题的话，轻则影响团队其他成员的工作效率，重则可能出现丢失工作成果、发布错误版本等严重的后果。

对于任何一个管理流程来说，保证该流程正常运转的前提条件就是要有明确的角色、职责和权限的定义。特别是在引入了软件配置管理的工具之后，比较理想的状态就是：组织内的所有人员按照不同的角色要求、根据系统赋予的权限来执行相应的动作。一般说，软件配置管理过程中主要涉及下列的角色和分工：

#### (1) 项目经理 (Project Manager, PM)

项目经理是整个软件开发活动的负责人，他根据软件配置控制委员会的建议批准配置管理的各项活动并控制它们的进程。其具体职责为以下几项：

- 制定和修改项目的组织结构和配置管理策略。
- 批准、发布配置管理计划。
- 决定项目起始基线和开发里程碑。
- 接受并审阅配置控制委员会的报告。

#### (2) 配置控制委员会 (Configuration Control Board, CCB)

负责指导和控制配置管理的各项具体活动的进行，为项目经理的决策提供建议。其具体职责为以下几项：

- 定制变更控制流程。
- 建立、更改基线的设置，审核变更申请。
- 根据配置管理员的报告决定相应的对策。

#### (3) 配置管理员 (Configuration Management Officer, CMO)

根据配置管理计划执行各项管理任务，定期向 CCB 提交报告，并列席 CCB 的例会。其具体职责为以下几项：

- 软件配置管理工具的日常管理与维护。
- 提交配置管理计划。
- 各配置项的管理与维护。
- 执行版本控制和变更控制方案。
- 完成配置审计并提交报告。
- 对开发人员进行相关的培训。
- 识别软件开发过程中存在的问题并拟就解决方案。

#### (4) 开发人员 (Developer)

开发人员的职责就是根据组织内确定的软件配置管理计划和相关规定，按照软件配置管理工具的使用模型来完成开发任务。

有人把配置管理称为软件开发的一种艺术，配置管理就是对软件开发过程中的产品进行标识、追踪、控制的过程，目的就是为了减少一些不可预料的错误，提高生产率。在实施配置管理的时候，一定要结合企业的实际情况，制定适合本企业适合本项目的配置管理方案。这里给出一些建议：

- 对于小的企业或者小的项目，可以通过制定配置管理的过程规则，可以不使用配置管理工具，实现版本管理的功能。当然如果条件允许，使用工具更好。
- 对于中小企业或者中小项目，可以通过制定过程规则，同时使用简单的版本管理工具，实现

部分配置管理功能。

- 对于大企业或者大项目或者异地开发模式，必须配备专门的配置管理人员，同时，需要制定配置管理严密的过程规则和配置管理工具，尽可能多的实现配置管理功能。

配置管理计划的形式可繁可简，完全根据项目的具体情况而定。下面给出一个配置管理计划的参考模板：

1. 引言
2. 软件配置管理 (SCM)
  - 2.1 SCM 组织
  - 2.2 SCM 责任
  - 2.3 SCM 与项目中其他机构的关系
3. 软件配置管理活动
  - 3.1 配置标识
    - 3.1.1 配置项的标识
    - 3.1.2 项目基线
    - 3.1.3 配置库
  - 3.2 配置控制程序
    - 3.2.1 变更基线的规程
    - 3.2.2 变更要求和批准变更的程序
    - 3.2.3 变更控制委员会 (SCCB)，描述并提供以下信息
      - 规章；
      - 组成人员；
      - 作用；
      - 批准机制。
    - 3.2.4 用于执行变更控制的工具
  - 3.3 配置状态报告
    - 3.3.1 项目媒体的存储、处理和发布
    - 3.3.2 需要报告的信息类型以及对于这类信息的控制
    - 3.3.3 需要编写的报告、各报告的相应读者以及写出各报告所需的信息
    - 3.3.4 软件版本处理，包括下述信息
      - 软件版本中的内容
      - 软件版本提供给谁、何时提供
      - 软件版本载体是何种媒体
      - 安装指导
    - 3.3.5 必要的变更管理状态统计
  - 3.4 配置审核
    - 3.4.1 何时审核及审核次数；每次审核提供下述信息：
      - 审核的是哪个基线；
      - 谁进行审核；
      - 审核对象；
      - 审核中配置管理者的任务是什么，其他机构的任务是什么；
      - 审核的正式程度如何。
    - 3.4.2 配置管理评审，每个评审提供下述信息：
      - 有待评审的材料；
      - 评审中配置管理者的责任，其他机构的责任。

#### 10.4.2 配置管理的其他作用

配置管理在项目管理的其他方面，也起着非常重要的作用。例如产品生成、过程管理、软件的复

用、小组协作开发等。

软件开发结束之后，需要向用户交付软件产品，软件产品必须从软件配置管理库中生成。它将解决最新发布的产品应由哪些版本的文件和工具来生成的问题。

过程管理是确保软件组织的规程、方针和软件周期得以正确贯彻执行。它将解决要交付给用户的产品是否经过测试和质量检查的问题。现在，很多配置管理工具都具有过程管理的功能。

软件复用是软件发展到一定阶段的产物。如果我们开发的软件产品是具有一定功能和性能的初始系统，但还不能完全满足所有用户的需求，为使最终的产品能满足用户的需求，我们必须认真研究用户的真正需求。经调查，了解到“用户1”代表了一些用户，这个用户群使用的计算机为“机型1”，所用的操作系统是“操作系统1”；而“用户2”所代表的用户群使用着“机型2”和“操作系统2”（参见图10-14）。就是说，不同用户有不同的工作环境，我们的软件产品必须考虑到这些差异，并且充分地使其满足各个用户的使用要求。为做到这一点，产品的设计可能做成这样的安排（如图10-15所示），两类产品分别针对两个用户群设计产品内部的模块（按上面的说法是配置项）。

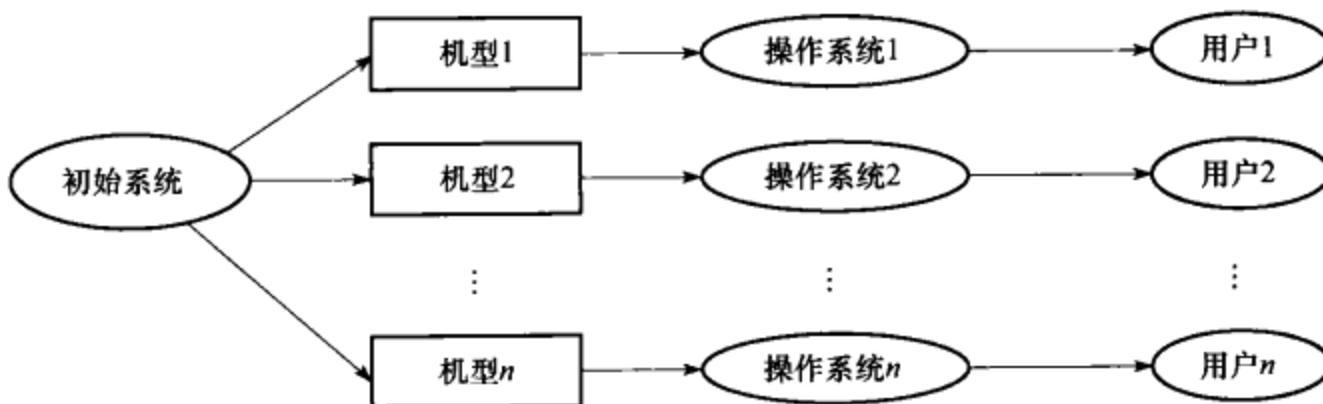


图 10-14 不同用户有自己的工作环境

其中，用户1包含的功能是A、B、C、D、E和F，而用户2包含的功能是A、B、C、D、E和G、H，两者的差别不仅表现在一个含有F，另一个含有G和H，而且即使两者的A在逻辑上是同一个内容，但在物理上仍然可能因两类用户需求的不同而有差异，例如，两个A分别以不同的媒体出现。

为实现这两种不同的软件配置，在实际工作中，我们完全可以将各个配置项分别开发出来，再根据需要，组合成针对不同用户使用要求的不同产品，正如图10-16所示。它是一个很好的复用的例子。

配置控制使用户能够通过对适当版本的选择来组成特定属性（配置）的软件系统，这种灵活的“组装”策略使得配置管理系统象搭积木似的使用已有的积木（版本）组装成各种各样、不同功能的模型。软件产品的每个版本都是一组配置项（源代码、文档、数据）的集合。配置控制就是要保证每个配置的完整性和精确性。

综上所述，配置管理是当今复杂软件项目得以实施的基础。通过有效的将现今复杂的系统开发过程以及产品纳入配置管理之下，使软件项目得以有效的、清晰的、可维护的和可控制的进行。也许在软件工程初期，可以通过手工维护所有软件产品及中间文档，但随着软件工程的发展，更主要的是随

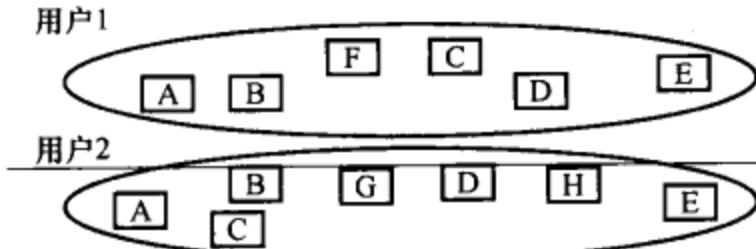


图 10-15 不同用户产品的不同配置

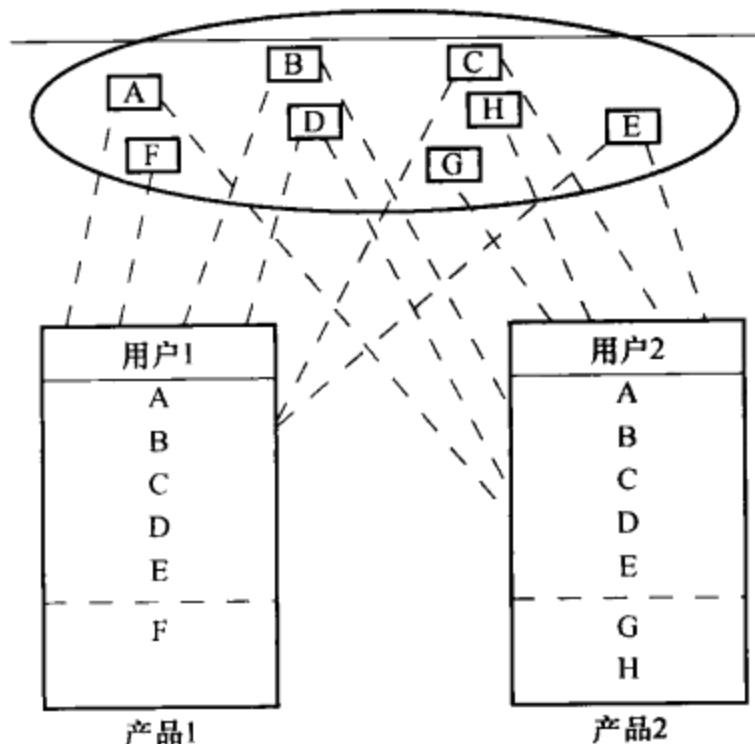


图 10-16 两个产品具有不同的配置

着软件系统复杂度的提高，可靠性的提高，必须要求有高效的配置管理同它相适应。

## 10.5 配置管理工具

配置管理包括三个主要的要素：人、规范、工具。首先，配置管理与项目的所有成员都有关系，项目中的每个成员都会产生工作结果，这个工作结果可能是文档，也可能是源程序等。规范是配置管理过程的实施程序。为了更好地实现软件项目中的配置管理，除了过程规范外，配置管理工具起到很好的作用。现代的配置管理工具提供了一些自动化的功能，从而大大方便了管理人员，减少了繁琐的人工劳动，但是必须为项目人员提供必要的有关配置管理系统知识的培训。

选择什么样的配置管理工具，一直是大家关注的热点问题。确实，与其他的一些软件工程活动不一样，配置管理工作更强调工具的支持；缺乏良好的配置管理工具的话，要做好配置管理的实施会非常困难。

选择工具就要考虑经费。市场上现有的商业配置管理工具，大多价格不菲。到底是选用开放源代码的自由软件、还是采购商业软件。一般来说，如果经费充裕的话，采购商业的配置管理工具会让实施过程更顺利一些，商业工具的操作界面通常更方便一些，实施过程中出现与工具相关的问题也可以找厂商解决。如果经费有限的话呢，就不妨采用自由软件，其实无论在稳定性还是在功能方面，自由软件也是一个不错的选择。

一个好的配置管理工具应该具备如下的功能：

- 并行开发支持：要求能够实现开发人员同时在同一个软件模块上工作，同时对同一个代码部分作不同的修改，即使是跨地域分布的开发团队也能互不干扰，协同工作，而又不失去控制。（对于这一点来说可能 CVS 比 VSS 做的更好，如果 VSS 不使用辅助工具 SOS (Source Off Site) 的话，那个公司或者是团队会把自己的 VSS 库共享到 Internet 上）。
- 履历管理：也就是修改的历史记录的可追踪性。能够明确地知道什么时候，谁作了什么，为什么这么做。从而达到管理和追踪开发过程中危害软件质量以及影响开发周期的缺陷和变化。
- 版本控制：能够简单、明确地取得软件开发期间的任何一个历史版本。
- 过程控制：能够贯彻、实施开发规范，包括访问权限控制、开发规则的实施等。
- 产品发布管理：软件开发过程中的一个关键活动是提取工件的相关版本，以形成软件系统的阶段版本或发布版本，我们一般将其称为稳定基线。一个稳定基线代表新开发活动的开始，而一系列定制良好的活动之后又会产生一个新的稳定基线。有效地利用此项功能，在项目开发过程中可以至始至终管理、跟踪工件版本间的关联。

配置管理工具可以提供必要的配置项管理，支持建立配置项的关系，并对这些关系进行维护、版本管理、变更控制、审计控制、配置项报告/查询管理等。当然，有的配置管理工具也提供了相关的其他功能，例如软件开发的支持、过程管理、人员功能等。

1975 年，加利福利亚大学的 Leon Presser 教授成立了一家名为 SoftTool 的公司，开发了自己的配置管理工具：CCC，这也是最早的配置管理工具之一。在软件配置管理工具发展史上，继 CCC 之后，最具有里程碑式的是两个自由软件：Marc Rochkind 的 SCCS (Source Code Control System) 和 Walter Tichy 的 RCS (Revision Control System)，它们对配置管理工具的发展做出了重大的贡献，直到现在绝大多数配置管理工具基本上都源于它们的设计思想和体系架构。下面介绍几种常见的配置管理软件。

### 10.5.1 Rational ClearCase

Rational 公司是全球最大的软件 CASE 工具提供商，现已被 IBM 收购。也许是受到其拳头产品、可视化建模第一工具 Rose 的影响，它开发的配置管理工具 ClearCase 也是深受用户的喜爱，是现在应用面最广的企业级、跨平台的配置管理工具之一，配置管理工具的高档产品，是软件业公认的功能最大、价格最贵的配置管理工具。

ClearCase 主要应用于复杂的并行开发、发布和维护。功能包括版本控制、工作空间管理、构造管理、过程控制。

### (1) 版本控制

ClearCase 不仅可以对文件、目录、链接进行版本控制，同时还提供了先进的版本分支和归本功能用于支持并行开发。另外，它还支持广泛的文件类型。

### (2) 工作空间管理

可以为开发人员提供私人存储区，同时可以实现成员之间的信息共享，从而为每一位开发人员提供一致、灵活、可重用的工作空间域。

### (3) Build 管理

对 ClearCase 控制的数据，既可以使用定制脚本，也可使用本机提供的 make 程序。

虽然 ClearCase 有很强大的功能，但是由于其不菲的价格，让很多的软件企业望而却步。而且需要一个专门的配置库管理员负责技术支持，还需要对开发人员进行较多的培训。

## 10.5.2 Hansky Firefly

做为 Hansky 公司软件开发管理套件中重要一员的 Firefly，可以轻松管理、维护整个企业的软件资产，包括程序代码和相关文档。Firefly 是一个功能完善、运行速度极快的软件配置管理系统，可以支持不同的操作系统和多种集成开发环境，因此它能在整个企业中的不同团队，不同项目中得以应用。

Firefly 基于真正的客户机/服务器体系结构，不依赖于任何特殊的网络文件系统，可以平滑地运行在不同的 LAN、WAN 环境中。它的安装配置过程简单易用，Firefly 可以自动、安全地保存代码的每一次变化内容，避免代码被无意中覆盖、修改。项目管理人员使用 Firefly 可以有效地组织开发力量进行并行开发和管理项目中各阶段点的各种资源，使得产品发布易于管理，并可以快速地回溯到任一历史版本。系统管理员使用 Firefly 的内置工具可以方便的进行存储库的备份和恢复，而不依赖于任何第三方工具。

## 10.5.3 CVS

CVS 是 Concurrent Versions System 的缩写，它是开放源代码软件世界的一个伟大杰作，它的基本思路是：在服务器上建立一个仓库，仓库里可以存放许多的文件，每个用户在使用仓库文件的时候，先将仓库的文件下载到本地工作空间，在本地进行修改，然后通过 CVS 的命令提交并且更新仓库的文件。由于其简单易用、功能强大、跨平台、支持并发版本控制、而且免费，它在全球中小型软件企业中得到了广泛使用。

其最大的遗憾就是缺少相应的技术支持，许多问题的解决需要自己寻找资料，甚至是读源代码。

## 10.5.4 Microsoft Visual Source Safe

Visual Source Safe，即 VSS，是微软公司为 Visual Studio 配套开发的一个小型的配置管理工具，准确来说，它仅能够称得上是一个小型的版本控制软件。VSS 的优点在于其与 Visual Studio 实现了无缝集成，使用简单。提供了创建目录、文件添加、文件比较、导入/导出、历史版本记录、修改控制、日志等基本功能。与 ClearCase 比起来功能比较简单，且由于其实惠的价格，方便的功能，是目前国内比较流行的。

但其缺点也是十分明显的，只支持 Windows 平台，不支持并行开发，通过 Check out- Modify- Check in 的管理方式，一个时间只允许一个人修改代码，而且速度慢、伸缩性差、不支持异地开发。

## 10.5.5 其他工具

当然，在项目配置管理的工具还有很多，例如 MERANT 公司的 PVCS、CCC Harvest 等等，PVCS 能够提供对软件配置管理的基本支持，通过使用其图形界面或类似 SCCS 的命令，能够基本满足小型项目开发的配置管理需求。PVCS 虽然功能上也基本能够满足需求，但是其性能表现一直较差，逐渐地被市场所冷落。

## 10.5.6 如何选择配置管理工具

随着软件开发规模的逐渐增大，越来越多的公司和团队意识到了软件配置管理的重要性，而相应

的软件配置管理工具也如雨后春笋一般，纷纷涌现，面对这些形形色色，各有千秋的配置管理工具，如何根据组织特点，开发团队需要，选择切合适用的工具呢？配置管理工具的选择所需考虑的因素大体包括以下几个因素：

1) 功能是否符合实际需求，是否符合团队特点。工具就是用来帮助您解决问题的，因此功能是否符合实际需求是最重要的判断因素。而大多数主流配置管理工具的基本功能都能够满足，因此主要需要判断以下几个因素：

- 并行开发支持。在团队协作开发过程中，有两种主要的模式：集体代码权和个体代码权。采用集体代码权模式进行开发时，一段代码可能同时会被多个开发人员同时修改；而采用个体代码权模式进行开发时，每一段代码都始终被一个开发人员独享，别人需要修改时也会通过该开发人员完成。而配置管理软件针对这一情况，也采用了不同的策略：Copy- Modify- Merge（拷贝、修改、合并）的并行开发模式、Check out- Modify- Check in（签出、修改、签入）的独占开发模式。
- 异地开发支持。如果你的开发团队分布在不同的开发地点，就需要对工具的异地开发功能进行仔细的评估了。大多数工具都提供基于 Web 的界面，用户可以通过浏览器执行配置管理的相关操作，而且有些工具就通过这样的方法来实现对异地开发的支持。

这种实现方法有太多的局限性，例如网络（Internet）连接带宽的限制、防火墙以及安全问题等。真正意义上的异地开发支持，是指在不同的开发地点建立各自的存储库，通过工具提供同步功能自动或手动同步。这样做的好处是与网络无关，即便各个开发地点之间没有实时连通的网络，也可以通过 E-Mail 附件等其他方式将同步包发给对方，实现手动的同步。

值得说明的是，在不同开发点建立各自存储库的方式，主要适用于两个或两个以上位于不同地点的开发团队协作开发的情况。如果仅是采用虚拟团队合作的方式，开发人员以个体的形式散落在不同地方，则更适合通过 Internet 直接操作远程的配置管理服务器。

- 跨平台开发支持。如果企业需要从事多个不同平台下的开发工作，就需要配置管理工具能够对跨平台开发提供支持，否则势必会给开发、测试、发布等各个环节带来不便，将使大量的时间被浪费于代码的手工上传、下载中。
- 与开发工具的集成性。配置管理工具与开发工具是编码过程中最常用到两种工具，因此它们之间的集成性直接影响到开发人员的便利性，如果无法良好集成，开发人员将不可避免地在配置管理工具与开发工具之间来回切换。

2) 性能是否满意。配置管理工具软件的一些性能指标对于最终的选择也有着至关重要的影响。

- 运行性能。如果开发团队规模不大的情况下，配置管理工具软件的性能不会造成很大影响，但如果项目规模比较大，团队成员逐渐增多的情况下，其运行性能就会带来很大的影响。
- 易用性。从用户界面、与开发工具的集成性角度来说，这几款主流的配置管理软件均有较好的设计，均有较好的易用性。
- 安全性。选择配置管理工具的安全包括存储的文件、目录等被访问情况，读写权限等的处理。例如 Firefly 服务器上的存储目录不用共享，对客户端不透明，客户端不可直接访问存储目录，使系统更安全可靠。而 ClearCase 采用 C/S 模式，需要共享服务器上的存储目录以供客户端访问，这将带来一定安全隐患。PVCS 基于文件系统共享，而且需要以“可写”的权限共享存储目录，存在较大的安全隐患。CVS 采用 C/S 模式，不需要共享服务器上的存储目录，安全性较好。VSS 基于文件系统共享实现对服务器的访问，需要共享存储目录，这将带来一定安全隐患。

3) 费用是否可以接受。Rational ClearCase、Hansky Firefly 两款均属于企业级配置管理工具软件。ClearCase 价格较贵，相比之下 Hansky Firefly 是一款不错的选择。PVCS 其价格大约是每客户端几百美元的水平，对于国内企业来说，性价比不太划算。VSS 是微软打包在 Visual Studio 开发工具包之中的，显然花费的精力不大，价格也比较便宜，可以做为个人、小项目团队版本控制之用。CVS 则是一款完全免费的开源软件，性能较之企业级配置管理工具差距不大，也是一种不错的选择。

4) 售后服务如何。产品售后服务也是需要考虑的一方面，例如 ClearCase 是大型商用软件，已被 IBM 公司收购，但国内市场拓展有限，因此服务支持会受到限制。Firefly 也是大型商用软件，已在中 国成立分公司，全面拓展市场之中，在北京设有支持中心。CVS 做为开源软件，无官方支持，需要用户自己查找资料解决技术问题，现在也出现专门为 CVS 做技术支持的公司。PVCS 在中国市场开拓有限，国内没有支持中心。VSS 做为微软的非核心产品，技术支持有限。在其网站上有提供一些常见问题，只有对正式购买的用户提供一定的技术支持。

## 10.6 校务通系统案例分析

### 校务通系统配置管理计划

1. 导言  
略。
2. 组织及职责
  - (1) 根据《项目计划》中的角色分配，确定配置管理者，SCCB 成员。
  - (2) 项目经理是 SCCB 的负责人。
  - (3) 配置管理的角色和职责见表 1。
3. 配置管理环境  
由于本项目属于中小型项目，工期也不是很长，而且大家对 SourceSafe 也比较熟悉，所以采用 SourceSafe 做为配置管理工具。
- 3.1 目录结构  
本项目配置库的目录结构见表 2。

表 1 配置管理角色职责表

角色	人员	职责、工作范围
配置管理者	岳好	(1) 制定《配置管理计划》 (2) 创建和维护配置库
SCCB 负责人	韩万江	(1) 审批《配置管理计划》 (2) 审批重大的变更
SCCB 成员	项目经理-韩万江， 质量保证人员-郭天奇， 配置管理者-岳好	审批某些配置项或基线的变更

表 2 配置库的目录结构

序号	内容		说明	路径
	TCM		技术合同管理	\$ \ prj-School \ TCM
	RM		需求管理	\$ \ prj-School \ RM
	SPP		软件项目规划	\$ \ prj-School \ SPP
	SPTO		软件项目跟踪与管理	\$ \ prj-School \ SPTO
	SCM		软件配置管理	\$ \ prj-School \ SCM
	SQA		软件质量保证	\$ \ prj-School \ SQA
	SPE	软件 产品 工程	设计	\$ \ prj-School \ SPE \ DESIGN
			源代码	\$ \ prj-School \ SPE \ SOURCECODE
			目标代码	\$ \ prj-School \ SPE \ BUILD
			测试	\$ \ prj-School \ SPE \ TEST
			发布	\$ \ prj-School \ SPE \ RELEASE

### 3.2 用户及权限

本项目配置库的用户及权限如表 3。

4. 配置管理活动
- 4.1 配置项标识
  - 4.1.1 命名规范
 

命名规范适用于过程文档、生存期中各阶段的计划、需求、设计、代码、测试、手册等文件。

本项目文件命名规范由五个字段组成，从左到右依次为：公司、项目、类型、编号和版本号，如图 1 所示。这些字段用一横线（-）分隔。

表 3 配置库的用户权限

类别	人员	权限说明
配置管理者	岳好	负责项目配置管理， 对库拥有所有权限
项目管理	韩万江	访问、读
质量保证人员	郭天奇	访问、读
开发人员	姜岳尊，孙泉	访问、读
高层管理		访问、读

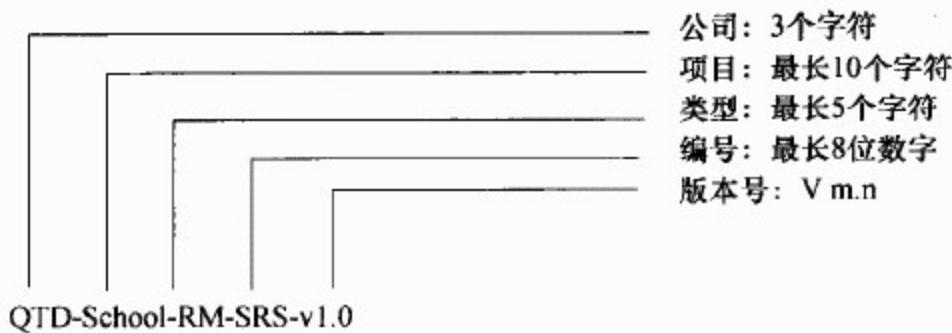


图1 文档命名规范

#### 4.1.2 主要配置项

本项目主要配置项如表4。

表4 配置项列表

类型	主要配置项	标识符	预计正式发表时间
技术合同	《合同》	QTD-School-TCM-Contract-V1.0	2003-4-11
	SOW	QTD-School-TCM-SOW-V1.0	2003-4-11
计划	《项目计划》	QTD-School-SPP-PP-V1.0	2003-4-11
	《质量保证计划》	QTD-School-SPP-SQA-V1.0	2003-4-11
	《配置管理计划》	QTD-School-SPP-SCM-V1.0	2003-4-11
需求	《需求规格说明书》	QTD-School-RM-SRS-V1.0	2003-4-18
	用户 DEMO	QTD-School-RM-Demo-V1.0	2003-4-18
设计	《总体设计说明书》	QTD-School-Design-HL-V1.0	2003-4-22
	《数据库设计》	QTD-School-Design-DB-V1.0	2003-4-22
	《详细设计说明书》	QTD-School-Design-LL-V1.0	2003-4-25
	《设计术语及规范》	QTD-School-Design-STD-V1.0	2003-4-22
编程	源程序	QTD-School-Code-ModuleName-V1.0	2003-6-2
	编码规则	QTD-School-Code-STD-V1.0	2003-4-22
测试	《测试计划》	QTD-School-Test-Plan-V1.0	2003-6-2
	《测试用例》	QTD-School-Test-Case-V1.0	2003-6-2
	《测试报告》	QTD-School-Test-Report-V1.0	2003-6-4
提交	运行产品	QTD-School-Product-Exe-V1.0	2003-6-5
	《验收报告》	QTD-School-Product-Repoort-V1.0	2003-6-6
	《用户手册》	QTD-School-Product-Manual-V1.0	2003-6-6

#### 4.1.3 项目基线

在SourceSafe中基线由LABEL标识，字母必须为大写。基线管理由项目执行负责人确认，SCCB授权，由配置管理员执行。本项目基线如表5所示。

#### 4.1.4 配置项的版本管理

配置项可能包含的分支从逻辑上可以划分成4个不同功能的分支，让它们分别对应4类工作空间。

- 主干分支
- 私有分支
- 小组分支
- 集成分支

上面定义的四类工作空间（分支）由项目执行负责人统一管理，根据各开发阶段的实际情况定制相应的版本选取规则，来保证开发活动的正常运作。在变更发生时，应及时做好基线的推进。

对配置项的版本管理在不同分支方面策略不同：

1) 主干分支。系统默认自动建立的物理分支——主干分支(/main)，BASELINE均以LABEL方式出现在主干分支上。

表5 基线发布计划

基线名称/ 标识符	基线所包含的主要配置项	预计建立时间
需求	《需求规格说明书》、 用户 DEMO	2003-4-18
总体设计	《总体设计说明书》、 《数据库设计》	2003-4-11
项目实现	软件源代码、 编码规则	2003-6-2
系统测试	《测试用例》、 《测试报告》	2003-6-4

2) 私有分支。如果多个开发工程师维护一个配置项时建议建立自己的私有分支。配置管理员对其基本不予管理，如个别私有空间上的版本树过于冗余，将对其冗余版本进行限制。

3) 小组分支。如果出现小组共同开发该配置项，该分支可视为项目组内部分组的私有空间，存放代码开发过程中的版本分支，由项目组内部控制。

4) 集成分支。集成测试时在主干分支的特定版本（由 LABEL 标识清晰）上建立集成分支，测试工作在集成分支上完成。

私有分支和小组分支均为可选，必要时建立。

#### 4.2 变更管理

变更管理的流程是：

1) 由请求者提交变更请求，变更控制委员会召开复审会议对变更请求进行复审，以确定该请求是否为有效请求。典型的变更请求管理有需求变更管理、缺陷追踪等。

2) 配置管理者收到基线修改请求后，在配置库中生成与此配置项相关的波及关系表。

3) 配置管理者将基线波及关系表提交给 SCCB，由 SCCB 确定是否需要修改，如果需要修改，SCCB 应根据波及关系表，确定需要修改的具体文件，并在波及分析表中标识出来。

4) 配置管理者按照出库程序从配置库中取出需要修改的文件。

5) 项目人员将修改后的文件提交给配置管理者。

6) 配置管理者将修改后的配置项按入库程序放入配置库。

7) 配置管理者按 SCCB 标识出的修改文件，由波及关系表生成基线变更记录表，并按入库程序放入配置库。

#### 4.3 配置状态统计

利用配置状态统计可以记录和跟踪配置项的改变。状态统计可用于评估项目风险，在开发过程中跟踪更改，并且提供统计数据以确保所有必须的更改被执行。为跟踪工作产品基线，配置管理者需收集下列信息：

- 基线类型
- 工作产品名称
- 配置项名称/标识符
- 版本号
- 更改日期/时间
- 更改请求列表
- 需要更改的配置项
- 当前状态
- 当前状态发生日期

项目组每周提交配置项清单及其当前版本。

配置管理人员每半个月提交变更请求的状态统计。

### 10.7 小结

配置管理可以有效的管理产品的完整性和可追溯性，而且可以控制软件的变更，保证软件项目的各项变更在配置管理系统下进行。一般配置管理过程包括：标识和跟踪配置项、建立配置管理环境、基线变更管理、配置审核、配置管理状态统计、报告管理、配置规划过程。

所有的配置管理活动都应在配置管理计划中进行合理的规划。配置管理计划可以根据项目的具体情况选择相应的配置管理过程。

### 10.8 练习题

#### 一、选择题

1. 在项目进行过程中，2 个项目成员使用不同版本的设计说明书，这时项目经理首先应该检查（ ）。
  - A. 信息管理系统
  - B. 配置管理系统
  - C. CPI
  - D. SPI

2. 变更控制主要关注的是（ ）。  
A. 阻止变更                           B. 标识变更，提出变更，管理变更  
C. 管理 SCCB                          D. 客户的想法
3. 为了更好的管理变更，需要定义项目基线，关于基线的描述，正确的是（ ）。  
A. 不可以变化的                      B. 可以变化，但是必须通过基线变更控制流程处理  
C. 所有的项目必须定义基线         D. 基线发生变更时，必须修改需求
4. 项目的基线发生变更应该经过（ ）授权执行的。  
A. 项目管理者                        B. 质量保证人员  
C. 配置管理人员                      D. SCCB
5. 变更控制系统必须包括下列所有的内容，除了（ ）。  
A. 文档说明                          B. 成功的谈判  
C. 跟踪系统                          D. 授权核准审批机构

## 二、判断题

1. 软件配置管理的目的是建立和维护整个生存期中软件项目产品的完整性和可追溯性。（ ）
2. 软件配置项是变更控制系统中的决策系统。（ ）
3. 统计被批准的配置项是一种配置审计。（ ）
4. 在进行配置管理过程中，一定要采用高档的配置管理工具。（ ）
5. 基线产品是不能修改的。（ ）

# 第11章

## ■ 项目集成计划

前几章讲述了很多计划，而每个计划都不是独立的，而是相互关联的，项目是一个集成的过程，包括从初始、计划、执行、监控到结束等过程。项目集成管理对项目的成功是至关重要的。作为项目管理必须有一个宏观的项目掌控，从大局出发的理念，当发生局部冲突的时候，项目经理必须作出一个最后的决定，形成项目集成计划。这章进入本书路线图的集成计划，如图11-1所示。

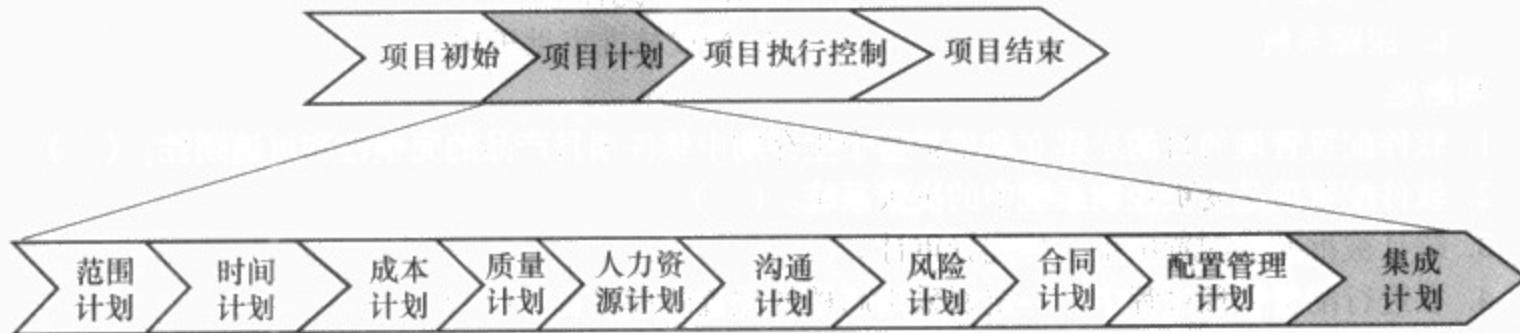


图11-1 路线图——项目计划—集成计划

### 11.1 软件项目集成管理

一个系统是一个整体，系统元素彼此联系、相互影响。小到微观粒子，大到宏观宇宙，都是一个相互影响、彼此联系的系统整体，都有其运行的内在规律。既然系统元素是相互影响彼此联系的，那么我们在软件项目管理中是否也可以用系统思考的方式来管理呢？项目集成管理的目标在于对项目中的不同组成元素进行正确高效的协调。它并不是所有项目组成元素的简单相加。项目集成管理就是在项目的整个周期内协调项目管理的各个知识域过程来保证项目的成功完成，项目经理的本职工作是对项目进行整合。为了成功地完成项目，项目管理者必须协调各个方面的人力、计划和工作。

软件开发项目必须识别项目的驱动、约束条件和自由程度。每个项目都需要平衡它的功能、人员、预算、进度和质量目标。范围、进度、成本、质量、风险等之间存在一定的关系，不能只考虑一个方面，应该综合起来考虑。压缩进度，可以导致成本提高，质量下降，也可能导致无法实现的进度风险；当然，如果减少成本，可能会影响进度，导致减少质量活动，降低了质量成本，最后可能发生成本不足的风险等。所以，应该建立一个关于进度、成本、质量、规模等的折中。

无论是进度计划、成本计划、质量计划还是风险计划等，所有计划的制定和管理都不是孤立的而是相互作用的，所有计划要以整体思想为指导，从全局角度出发，确保各项工作有机协调配合进行，消除管理的局部性，平衡各个目标的冲突。所以，项目计划应该是集成的计划，项目集成计划具有综合性、全局性、内外兼顾性。

项目集成管理是一项综合、统管全局的工作，它决定在什么时间，在哪些预期的潜在问题上集中资源，在问题变得严重前进行处理，协调项目干系人及各项工作使项目走向成功。

项目集成管理包括对计划的集成管理和项目跟踪控制的集成管理，它是保证项目各要素相互协调，在相互影响的项目目标和方案中做出权衡，以满足或者超出项目干系人的需求和期望。

在进行项目管理的时候，无论是哪个过程还是哪个阶段，都不应该只注重局部，应该将整体的管理思想融入到项目管理的方方面面，注重从项目的全局出发，从整体的角度分析问题、解决问题。例如：

1) 项目范围和产品目标的集成。作为项目经理，接受项目工作时，首先要考虑项目的工作范围。但是，工作范围是靠目标导向的，而目标就是提交让客户满意的项目产品。如果要压缩进度、降低成

本、提高质量的标准，都会直接影响到项目的工作范围；反之，项目的范围发生变化，也必然会影响项目的工期进度、成本、项目的质量，当然，项目的提交物也会发生相应的变化。

2) 在既定的工作范围内时间、成本、质量等目标的集成。需求定义了系统的范围与规模，资源决定了项目的投入（人、财、物），工期定义了项目的交付日期，质量定义了做出的系统好到什么程度，这四个要素之间是有制约平衡关系的。如果需求范围很大，要在较少的资源投入下，很短的工期内，以很高的质量要求来完成某个项目，那是不现实的，要么需要增加投资，要么工程延期；如果需求界定清楚了，资源固定了，对系统的质量要求很高，则可能需要延长工期。如图 11-2 所示，项目的范围目标、成本目标、进度目标、质量目标等各个目标是相互制约的；在诸多项目指标中，项目的进度和成本是关系最为密切的两个，几乎成了对立关系，进度的缩短常常要依靠增加成本实现，而成本的降低也常常以牺牲工期进度为代价。工期和成本与质量的关系也一样很密切，在一些项目中如果盲目缩短工期就会导致项目质量的下降，同样，质量出现问题、出现返工，也一样会延长工期；当然有的项目为了压缩成本，而减少一些必须的质量环节，会导致质量的下降。

关于时间、成本、质量三大目标管理的集成，既要依靠项目管理者和项目团队的经验和智慧，又需运用一些专业的工具。范围 (S)、质量 (Q)、进度 (T)、成本 (C) 为软件项目管理的最重要的四个要素，可以设定成本是范围、质量、进度的一个函数： $C = F(S, Q, T)$ ，其中：

- S 与 C 成一定的正比关系；
- Q 与 C 成一定的正比关系；
- T 与 C 成一定的反比关系。

对于一个项目经理，在项目预算确定的情况下，如何平衡范围、质量和进度成为了项目管理者的重要工作，只有处理好了这个问题，才能最大程度提高客户满意度。这四个要素中，客户最关注的是范围、质量、进度，在确定合同的情况下，客户一般不关心你的成本。而从项目经理的角度来说最关注的是成本、范围和进度。项目经理对质量的关注往往是在客户能忍受的底线。这么看来双方共同关注的是范围和进度。进度对双方来说几乎不能改变，牵涉到各方面的因素和来自多方的压力，需要尽可能地满足。范围是双方争论的焦点，客户希望花钱越少，功能越多，而你却希望合同额越高，功能越少。由此看来，双方都关注的就只剩下进度了。所以，项目经理无论如何都要处理好进度管理的问题，在这个问题上必须跟客户取得完全一致。

通过上面的论述，可以知道，一旦合同签订，四个因素中的两个，成本和进度几乎是确定的，那么在这种情况下如何提高客户满意度呢？

$$C = F(S, Q, T)$$

其中 C、T 是常量，S 与 C 成一定的正比关系，Q 与 C 成一定的正比关系。

在成本或者时间不充足的情况下，只能通过减少范围 S，或者降低质量 Q 来解决。到底是减少范围还是降低质量呢？这个需要根据不同的情况做出不同的选择。主要是根据客户的情况，若更重视范围就降低质量，若更重视质量就降低范围，若范围质量都重视，那就都有所降低，降到用户基本能忍受为止。

项目管理的目的就是通过对项目活动进行全过程、全方位的计划、组织、控制和协调，使项目在约定的时间内满足客户的需要，实现最终的项目目的，使项目取得成功。

## 11.2 项目集成计划

项目计划就像一张地图，它告诉开发人员如何从一个地方到达另外一个地方。编制计划需要做大量的工作：明确需求、任务分解、确定工作进度、分配资源等各方面的事情，初始计划需要经过细化、修改、再细化之后，才可以形成这张地图。



图 11-2 项目目标的相互制约

项目计划的合理程度与项目计划制定者从事类似规模和类似业务的项目的经验有直接关系，通过经验往往能够预见潜在的阻碍，这样要求项目计划制定者需要集众人之力来完善计划。

对于每个项目，如何合理的保证项目质量，正确处理质量与时间、成本之间的矛盾是项目管理的一个难点，这需要整合项目所有方面的内容，保证按时、低成本地实现预定的质量目标。

根据侧重点不同，项目可分为质量倾斜型、工期倾斜型及成本倾斜型体系。我们在编制项目计划时，一般而言是时间、成本、质量标准均已确定，在项目实施过程中就需从客观因素、具体情况出发，根据将要采取的行动和可能导致的后果进行综合分析研究，按切合实际的原则，使项目进展平衡有节奏地进行，以求达到预期目标。避免出现工期紧张或成本减少，导致质量降低的现象，而质量下降又往往造成返工等后果而导致延长工期和增加成本。

### 11.2.1 项目集成计划定义

项目集成计划是指，通过使用其他专项计划过程所生成的结果（项目的各种专项计划），运用整体和综合平衡的方法所制定出的，用于指导项目实施和管理的整体性、综合性、全局性、协调统一的整体计划文件。项目集成计划是一个批准的正式文件，用来跟踪控制项目的执行。随着项目的发展，可以不断的完善，它包括一些重要的基准计划（例如范围基准、进度基准、成本基准、质量基准等），基准计划是不能随便修改的，要经过相应的变更程序才可以修改。

项目计划集成的结果是创建一个内容充实、结构紧凑的文件，使它能够引导项目的实施和控制。这个过程经常重复几次。例如：最初的草案可能包括一般性的方法并没有时间期限，而最终计划则要反映具体的方法和有明确的时间期限。这个项目计划用于引导项目的实施，促进项目参与者之间的沟通，确定主要的管理问题，如内容、范围和时间等。为进一步提高测量和控制项目的水平提供一个标准。项目经理可以根据需要形成一个比较规范的项目计划文档。项目计划是项目规划过程的产品，记录了项目估算和计划的过程与结果。并在整个项目开发过程中作为项目管理的依据进行使用和维护。

集成计划将其他领域的子计划进行集成，其中项目范围计划、时间计划、成本计划三大核心子计划，是进行项目计划编制的基础文件，同时，项目的质量计划、沟通计划、人力资源计划、风险计划、合同计划等子计划也将是编制计划的原材料，但是集成计划并不是简单的堆砌，而是需要不断的进行反馈，以使各个子计划不断校正自己以便符合项目的总目标。

为了编写一个好的集成项目计划，项目经理应该通晓项目知识域的相关知识，与项目团队人员一起协作完成项目计划。项目计划的作用主要体现如下：

- 1) 指导项目实施。
- 2) 促进项目干系人之间的沟通。
- 3) 对有关内容、成本和时间安排等进行管理审查。
- 4) 为项目控制提供基准计划。

项目具有唯一性，项目管理计划同理也具有唯一性，与项目规模的大小也有关系，一个规模比较小的项目，项目计划包含的内容也比较少，对于规模比较大的项目，项目计划会复杂些，项目计划会更加详细一些的。项目经理应该根据项目的需要适当的选择裁减项目计划的内容。

### 11.2.2 集成计划的编写过程

在高技术行业，日新月异是主要特点，因此项目计划的制定需要在一定条件的限制和假设之下采用渐近明细的方式进行不断完善。一个好的项目计划的开发应该是渐近式的。由于软件开发的手工性、个体性特征，软件开发项目计划不可能是一个静态的计划，项目启动时，可以先制定一个颗粒度相对比较粗的项目计划，先确定项目高层活动和预期里程碑。粗颗粒度的项目计划需要不断地更新迭代，根据项目的大小和性质以及项目的进展情况进行迭代和调整。迭代和调整的周期也是根据项目的情况进行制订的。经过不断的计划制订、调整、修订等工作，项目计划从最初的粗粒度，变得非常详细。

这样的计划将一直延续到项目结束，延续到项目的成果出现。

一个完善的项目计划可以采用多层计划体系来逐步达到的，如图11-3所示。

最初项目计划不够精确和准确，不能直接拿来指导我们的日常工作，也不易跟踪。我们采用多层次计划机制将计划中的任务拆分成可跟踪的小的任务来执行。例如，可以将计划分成大、中、小三个层次，大计划由中计划构成，中计划由小计划构成，小计划可以说就是个人任务的规划。这样做也是为了使我们工作容易进行。

层次结构体现了工作流程，符合项目实际工作需要。在项目开始时，一般是很难具体到策划项目的细节工作，所以这时要求制作大计划，一般不包含细节内容，但列出了相对精细的中计划。之后中计划就开始被清晰地制定了出来，此时的中计划包含了粗糙的小计划。随着中计划的实施，小计划也就从粗糙变成了详细了。这个结构赋予任何成员一个自我施展的空间。当然计划的制定必须经过项目经理的认可。

制定计划的过程就是一个对项目逐渐了解掌握的过程，通过认真地制定计划，项目经理可以知道哪些要素是明确的，哪些要素是要逐渐明确的，通过渐近明细化不断完善项目计划。阶段计划中包含的工作汇报和下一阶段工作安排是掌握项目进度的依据，从阶段计划对照总体计划，才能一目了然地看出工作的进展情况。制定计划的过程，也是在进度、资源、范围之间寻求一种平衡的过程。制定计划的精髓不在于写出一份好看的文档，而在于运用您的智慧去应对各种问题和面临风险并尽可能做出前瞻性的思考。一旦计划被负责任地完成，他就可以给自己一个和管理层或客户交流与协商的基础，帮助你在项目过程中防范各种问题的出现，帮助你保证项目按时完成。

项目集成计划的编制步骤简单说明如下：

- 1) 项目信息收集、项目计划整体的综合性分析。项目计划整体的综合分析是编写项目计划时不可缺少的。项目在计划制定前，要对前期阶段所收集的信息、数据进行综合分析。了解总体目标等。还要对各要素相互依存的关系进行分析。在计划整体综合分析时要细致、实际、认真、科学地做好全方面的分析。我们在做项目计划整体综合分析时，要综合考虑到项目的工期、质量、成本三个要素。项目的成本与工期是相关的，项目的成本是随着工期的变化而变化的。工期直接影响着成本的增减，所以必须综合分析二者的关系和相互变化的影响。同时还要综合分析项目质量和成本的关系，我们对质量有相应的控制，这时成本也会随着控制的方法有所变动。所以质量的保障与否也直接影响着成本的高低。一个项目的成本不仅与工期有关，而且与项目质量直接相关。

项目经理组织项目团队成员通过分析项目相关文档、进一步与用户沟通等途径，在规定的时间内尽可能全面收集项目信息。项目信息收集要讲究充分的、有效率的沟通，并要达成共识。

- 2) 确定项目计划初步方案。项目经理组织前期加入的项目团队成员准备项目工作所需要的规范、工具、环境。如开发工具、源代码管理工具、配置环境、数据库环境等。只有对整体信息资料进行完全有效的实际分析后才可以制定项目计划初步方案。项目计划初步方案可以使用分析整体的方法。首先要对项目工期和成本两大要素进行规划，再将质量、资源等个要素集成一个综合统一的项目计划。项目计划初步方案是以技术和项目各种时间资源出发编制的。初步方案只是对下一步综合平衡和进一步优化。

- 3) 项目计划的综合平衡。在制定项目计划初步方案后，需运用综合平衡法对项目目标、任务、责任、进度、成本、质量等各种要素进行统一、协调。同时，最重要的是请项目的主要干系人参加，以各方要求和期望对项目计划的初步方案进行评价和调整，在综合平衡时要准确有效地作好对各要素的平衡。要由项目相关利益主体对项目初步方案进行全面的评审，以达到全部项目干系人的要求和满意。

项目计划书必须在相应阶段对项目目标、阶段目标和各项任务进行精确的定义，就是要在相应阶段进一步进行项目目标的细化工作。应当充分调查并掌握影响项目计划的一切内部和外部影响因素；

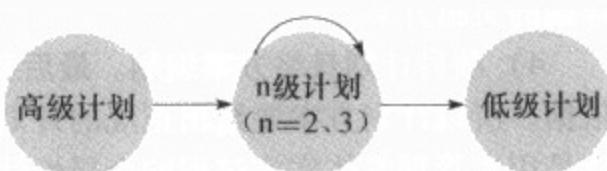


图11-3 渐近式计划

应当尽可能充分地分析项目任务分解结构，通过分析项目任务分解结构不仅获得项目的静态结构，而且通过逻辑分析，获得项目各工作任务之间动态的工作流程；应当将项目目标、任务进行分解，制定详细的实施方案。

4) 项目计划最终方案编制。最后我们就可以编制项目计划了，项目计划是项目组实施项目时的依据。所以计划方案要给出正规的计划格式，并要求项目的各方最高决策者予以批准。项目计划文件是用来管理整体实施过程的全局性计划文件。可以根据不同的项目编写详细程度不同的项目计划文件。

5) 软件项目计划书评审、批准。项目计划书评审、批准是为了使相关人员达成共识、减少不必要的错误，使项目计划更合理更有效。项目经理确保与所有人员就项目计划书中所列内容达成一致。这种一致性是要求所有项目团队成员对项目计划的内容进行承诺，无法承诺或者说是无法达成一致的，要么修改项目计划去适应某些项目团队成员，要么是由某些项目团队成员采取妥协措施，去适应项目计划的要求。

项目经理将已经达成一致的软件项目计划书提交项目高层分管领导或其授权人员进行审批，审批完成时间不能超过预先约定的时间。

批准后的软件项目计划书作为项目活动开展的依据和本企业进行项目控制和检查的依据，并在必要时根据项目进展情况实施计划变更。

### 11.2.3 编写项目集成计划书

集成的项目计划书是重要性的文件，所以项目的计划文件应该包括项目的批准与特许情况描述与说明、项目整体管理方法、项目任务范围综述、WBS 描述和说明、项目的成本和进度描述和说明、每项工作的工期、成本、绩效度量基准的描述和说明、项目重要里程碑和目标日期描述和说明、项目团队的关键成员和所需人员的描述和说明、项目的主要风险、限制、前提与预期的描述和说明、项目专项计划方面的描述和说明、存在问题和尚需决策方面的描述和说明。

对于集成的项目计划书，企业可以根据自己的需要选择合适的标准，并根据情况做适当的裁减。项目经理负责组织编写计划书，它包括计划书主体和以附件形式存在的其他相关计划，如配置管理计划等。项目计划内容基本如下：

1) 确定项目概貌。合同项目以合同和招投标文件为依据，非合同项目以可行性研究报告或项目前期调研成果为依据，明确项目范围和约束条件，并以同样的依据，明确项目的交付成果。进一步明确项目的工作范围和项目参与各方的责任。

2) 确定项目团队。确定项目团队的组织结构和与项目开发相关的职能机构，包括管理、开发、测试、QA、评审、验收等。确定项目团队人员及分工。与相关人员协商，确定项目团队人员构成。如内部不能满足人员需求，则提出人员支援申请。

3) 明确项目团队内、外的协作沟通。明确与用户单位的沟通方法。明确最终用户、直接用户及其所在本企业/部门名称和联系电话。客户更多的参与是项目成功的重要推动力量，加强在开发过程中与用户方项目经理或配合人员的主动沟通，将有助加强客户等项目的参与程度。建议采用周报或月报的方式通告项目的进展情况和下一阶段计划，通告需要客户协调或了解的问题。

当项目团队需要与外部单位协作开发时，应明确与协作单位的沟通方式。确定协作单位的名称、负责人姓名、承担的工作内容以及实施人的姓名、联系电话。

明确与项目团队内部的沟通方式，应该组织项目团队定期召开周例会，项目团队采用统一的交流系统建立项目团队的交流空间。

4) 规划开发环境和规范。说明系统开发所采用的各种工具、开发环境、测试环境等。列出项目开发要遵守的开发技术规范和行业标准规范。对于本企业还没有规范的开发技术，项目经理应组织人员制订出在本项目中将遵守的规则。

5) 项目范围说明。项目范围说明阐述这个项目的背景，形成项目的基本框架，使项目所有者或项

目管理者能够系统地、逻辑地分析项目关键问题及项目形成中的相互作用要素，使项目干系人在项目开始实施前或项目相关文档编写以前，能够就项目的基本内容和结构达成一致；项目范围说明应当形成项目成果核对清单，作为项目评估的依据，在项目终止以后或项目最终报告完成以前进行评估，以此作为评价项目成败的依据；范围说明还可以作为项目整个生命周期监控和考核项目实施情况的基础，和项目其他相关计划的基础。

6) 编制项目进度计划。进度计划是说明项目中各项工作的开展顺序、开始时间、完成时间及相互依赖衔接关系的计划。通过进度计划的编制，使项目实施形成一个有机的整体。进度计划是进度控制和管理的依据。

在进度控制计划中，要确定应该监督哪些工作、何时进行监督、监督负责人是谁、用什么样的方法收集和处理项目进度信息、怎样按时检查工作进展和采取什么调整措施，并把这些控制工作所需的时间和人员、技术、物资资源等列入项目总计划中。

根据本企业规定和项目实际情况，确定项目的工作流程，编制项目的工作计划，各阶段的工作时间安排要包括完成阶段文档成果、文档成果提交评审及进行修改的时间，各阶段结束的标志是阶段成果发布。在计划中要求明确以下内容：

- 确定项目的应交付成果。这里的项目的应交付成果不仅是指项目的最终产品，也包括项目的中间产品。例如通常情况下软件开发项目的项目产品可以是：需求规格说明书、概要设计说明书、详细设计说明书、数据库设计说明书、项目阶段计划、项目阶段报告、程序维护说明书、测试计划、测试报告、程序代码与程序文件、程序安装文件、用户手册、验收报告、项目总结报告等。
- 任务分解：从项目目标开始，从上到下，层层分解，确定实现项目目标必须做的各项工作，并画出完整的工作分解结构图。
- 在资源独立的假设前提下确定各个任务之间的相互依赖关系，以确定各个任务开始和结束时间的先后顺序；获得项目各工作任务之间动态的工作流程。显示项目各阶段的时间分配情况，例如甘特图，确定主要里程碑、阶段成果。
- 确定每个任务所需的时间，即根据经验或应用相关方法确定任务需要的时间；确定每个任务所需的人力资源要求，如需要什么技术、技能、知识、经验、熟练程度等等。
- 确定项目团队成员可以支配的时间，即每个项目成员具体花在项目中的确切时间；确定每个项目团队成员的角色构成、职责、相互关系、沟通方式。
- 确定管理工作，管理工作是贯穿项目生命周期的，如项目管理、项目会议、编写阶段报告。项目团队成员之间的沟通时间、项目团队成员和其他项目干系人之间的沟通时间也比较容易被忽视，而沟通时间也是比较不容易固定地量化和日程化。但这些工作在计划中都应当充分地被考虑进去。
- 根据以上结果编制项目总体进度计划，总体进度计划应当体现任务名称、责任人、开始时间、结束时间、应提交的可检查的工作成果。
- 考虑项目的费用预算、可能的风险分析及其对策、需要公司内部或客户或其他方面协调或支持的事宜。

7) 项目成本计划。项目成本计划包括资源计划、费用估算、费用预算等。资源计划就是决定在项目中的每一项工作中用什么样的资源（人、材料、设备、信息、资金等等），在各个阶段使用多少资源。

8) 项目质量计划。质量计划针对具体项目，安排质量监控人员及相关资源、规定使用哪些制度、规范、程序、标准。项目质量计划应当包括与保证与控制项目质量有关的所有活动。质量计划的目的是确保项目的质量目标都能达到。

9) 项目沟通计划。沟通计划就是制定项目过程中项目干系人之间信息交流的内容、人员范围、沟通方式、沟通时间或频率等沟通要求的约定。

10) 风险计划。风险计划是为了降低项目风险的损害而分析风险、制定风险应对策略方案的过程，包括识别风险、评估风险、编制风险应对策略方案等过程。分析项目过程中可能出现的风险以及相应

的风险对策。

11) 项目合同计划。项目开发过程中需要识别哪些项目需求可应通过从本企业外部采购产品或设备来得到满足。如果是软件项目决定外包，应当同时制定对外包的进度监控和质量控制的合同计划。

12) 配置管理计划。由于项目计划无法保证一开始就预测得非常准确，在项目进行过程中也不能保证准确有力的控制，导致项目计划与项目实际情况不符的情况经常发生，所以必须有效处理项目的变更。配置管理计划就是确定项目的配置项和基线，制定基线变更控制的步骤、程序，维护基线的完整性，向项目干系人提供配置项的准确状态和当前配置数据。

配置管理员根据计划书编制《项目配置管理计划》。以项目工作计划书中的阶段成果为依据，根据配置管理计划规范编制配置管理计划，项目经理审批配置管理计划，并对配置管理计划的有效性负责。

13) 制定其他辅助工作计划。根据项目需要，编制如培训计划、招聘计划、支持工作、验收计划等等。

项目计划文件是一种正式的、并获得批准的项目计划管理文件，是用来管理项目和控制整个项目实施全过程的综合性、全局性的计划文件，它根据项目的不同，详细程度也不同。

### 11.3 校务通系统案例分析

#### 校务通系统项目集成计划

##### 1. 导言

略。

##### 2. 概述

《校务通系统》是对学校教务和教学活动进行综合管理的平台系统，是基于 Internet 环境的综合信息系统，满足学校管理层、教师、学生、家长等日常工作、学习、管理、咨询等工作。目的是共享学校各种资源、提高学校的工作效率、规范学校的工作流程、便利校内外的交流。系统具有标准化、分布式存储和检索、易用、易维护、开放等特点。

##### 3. 项目任务范围

《校务通系统》项目需完成的任务总的分为两类：通用功能和学校日常业务管理功能。其中通用功能包括电子课表、会议通知和公告、日程安排、个人日记、通讯录、教师答疑、家庭作业等。学校日常业务管理功能包括招生管理、学生日常管理、教务管理、教师备课系统、资源库系统、网上考试功能、聊天室、论坛等。图 1 是项目任务的范围图示。

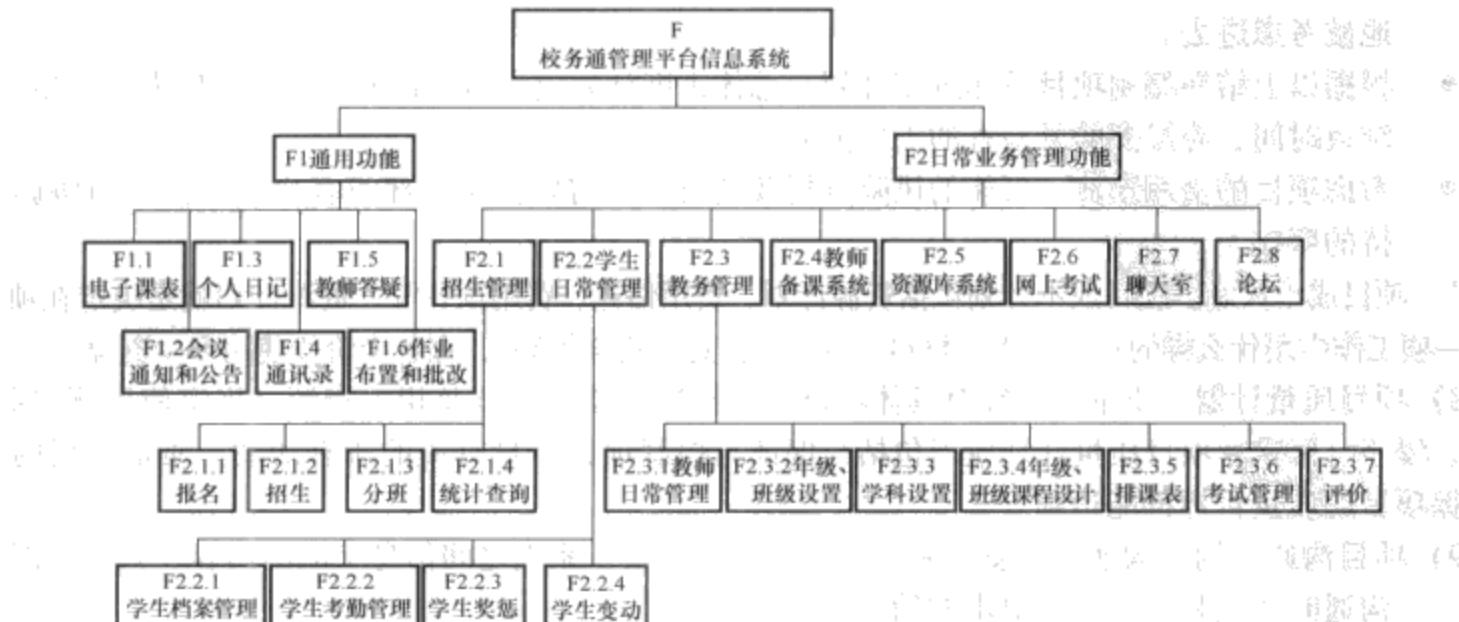


图 1 任务范围

#### 4. 项目目标

目前电化教育已经越来越普及，各地的学校纷纷建设自己的校园网，但是好多学校在投巨资建设校园网之后，未能高效利用校园网的资源。校务通管理系统提供了有效利用校园网，实现学校管理的电子化。本项目的产品可以达到以下目标。

- 提高生产效率，减少返工。
- 节省开支。
- 业务过程的流水线化。
- 先前人工劳动的自动化。
- 符合相关标准和规则。
- 与目前的应用产品相比较，提高了可用性或减少了失效程度。

另外，通过项目进一步验证和完善公司的质量体系，同时锻炼开发队伍的协同精神。

#### 5. 项目实施策略

实施策略是确定如何实施项目，以达到项目目标的策略。根据校务通项目特点和企业的战略要求，采取如下策略：

- 项目管理策略

- 1) 项目管理过程遵循公司质量体系中关于项目管理过程规范。
- 2) 根据项目计划中的评审点进行跟踪和管理，并根据结果对项目计划进行适当的调整。
- 3) 评审采用定期评审、阶段评审和事件评审相结合的方式。
- 4) 按周发布项目简报，通报项目进展情况及其他相关情况。

- 软件开发策略

- 1) 采用OO技术逐步构造系统。
- 2) 产品按阶段提交。
- 3) 开发实施过程采用公司的复用技术，同时遵循公司质量体系中关于项目实施过程规范。

- 质量保证策略

- 1) 质量管理过程遵循公司质量体系中关于项目质量管理过程规范。
- 2) 加强对项目参与人员的质量保证概念的培训。
- 3) 加强对过程的控制，重点确定该项目中需控制的过程。
- 4) 加强对产品规范的审计，重点确定该项目中需审计的产品。
- 5) 实施完整的软件配置管理。

#### 6. 项目组织结构

由于该项目在实施过程中需要涉及不同组织的各方面人员，而各组织之间的利益、任务和职责也不尽相同，因此明确定义项目组织结构和各自职责可保证项目的顺利进行。该项目的组织结构图如图2所示：

其中：

市场部

- 负责与用户的协调工作。
- 负责项目相关的商务活动。
- 负责用户需求的接口。
- 配合项目经理的资源协调活动。
- 负责产品的验收活动。
- 负责系统的维护活动。

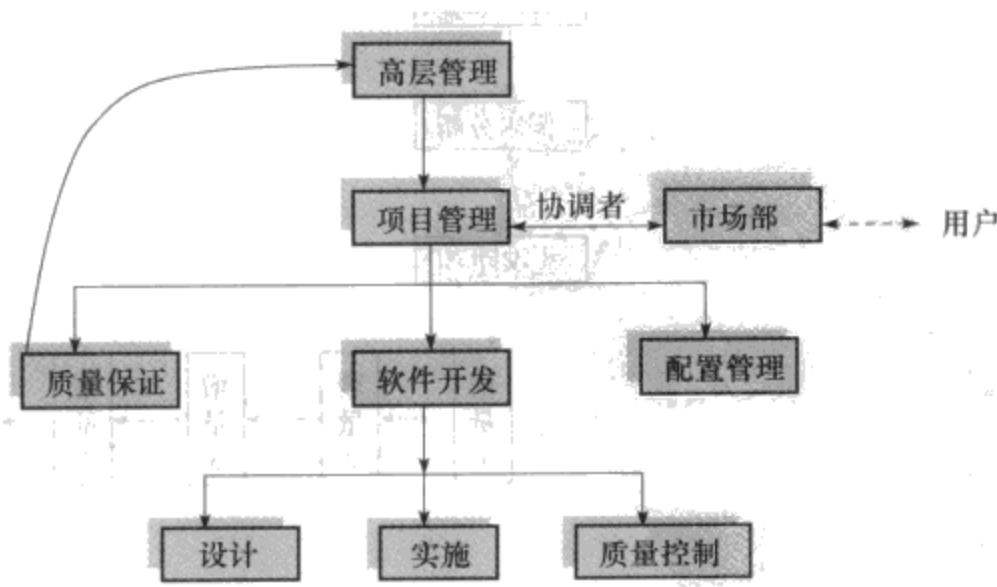


图2 项目的组织结构

### 项目管理

- 负责项目的组织和规划。
- 负责项目计划制定和维护。
- 负责项目的跟踪和管理。
- 负责资源的分配和协调活动。
- 负责各组织和计划之间的协调活动。
- 负责与市场部的协调活动。

### 软件开发

- 负责项目的软件开发，包括设计、编码、单元测试和集成测试。
- 负责产品质量控制的工作。
- 负责配合质量保证的活动，如系统测试、文档编制等。
- 配合产品验收的相关活动。

### 质量保证

- 负责项目过程和产品规范的制定。
- 负责项目过程的质量保证活动。
- 过程评审。
- 产品审计。

### 配置管理

- 负责项目的配置管理活动。
- 负责软件产品的提交。

### 用户

- 确保相关责任的实施。
- 参与项目的组织和规划。
- 负责产品的验收工作。

表1 为角色映射表。

表1 角色映射表

	项目管理组	软件开发组	质量保证组	配置管理组
负责人	韩万江	姜岳尊	郭天奇	岳好
组成(人)	1	2	1	1

## 7. 项目生存期

根据该项目的特点并结合公司已有的软件生存期模型定义，本项目生存期采用增量模型如图3。

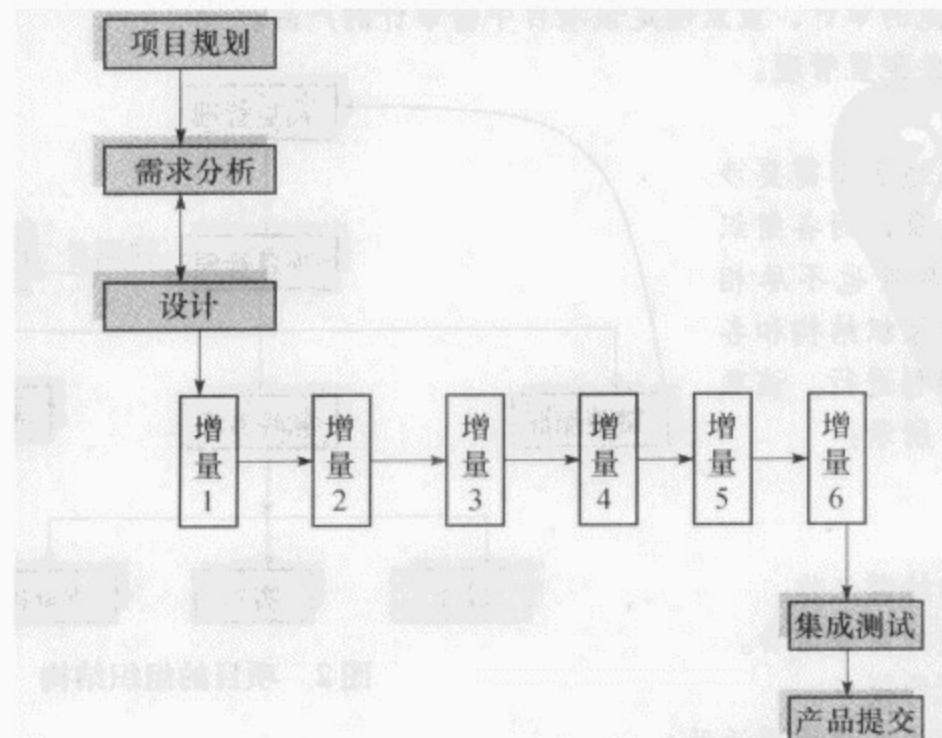


图3 生存期模型

生存期中的各阶段定义如下：

#### 项目规划阶段

阶段目标：根据合同和初步的需求分析确定项目的规模、时间计划和资源需求。

输入：合同文本、SOW

过程：项目规划，计划确认

输出：项目计划

#### 需求分析阶段

阶段目标：确定客户的需求

输入：项目计划，SOW

过程：需求获取，需求分析，需求控制

输出：原型系统，需求规格

#### 设计阶段

阶段目标：总体系统结构设计

输入：原型系统，需求规格

过程：总体设计

输出：系统设计说明书，数据库结构定义

#### 增量1 实现

阶段目标：实现系统的通用功能

输入：系统设计说明书、数据库结构定义

过程：详细设计，编码，代码走查，代码评审，单元测试

输出：详细设计说明书，源代码，可运行版本-1

#### 增量2 实现

阶段目标：实现系统的招生管理功能

输入：系统设计说明书、数据库结构定义

过程：详细设计，编码，代码走查，代码评审，单元测试

输出：详细设计说明书，源代码，可运行版本-2

#### 增量3 实现

阶段目标：实现系统的学生成绩管理功能

输入：系统设计说明书、数据库结构定义

过程：详细设计，编码，代码走查，代码评审，单元测试

输出：详细设计说明书，源代码，可运行版本-3

#### 增量4 实现

阶段目标：实现系统的教务管理功能

输入：系统设计说明书、数据库结构定义

过程：详细设计，编码，代码走查，代码评审，单元测试

输出：详细设计说明书，源代码，可运行版本-4

#### 增量5 实现

阶段目标：实现系统的教师辅助功能

输入：系统设计说明书、数据库结构定义

过程：详细设计，编码，代码走查，代码评审，单元测试

输出：详细设计说明书，源代码，可运行版本-5

#### 增量6 实现

阶段目标：实现系统的聊天室/论坛功能

输入：系统设计说明书、数据库结构定义

过程：详细设计，编码，代码走查，代码评审，单元测试

输出：详细设计说明书，源代码，可运行版本-6

#### 集成测试

阶段目标：通过集成环境下的软件测试

输入：测试计划、测试案例

过程：集成测试，系统测试

输出：系统软件包，测试报告，产品说明书

#### 产品提交

阶段目标：产品可投入使用

输入：系统软件包

过程：产品提交

输出：验收报告

### 8. 时间计划

项目进度计划甘特图如图4所示（详见进度计划mfp）。

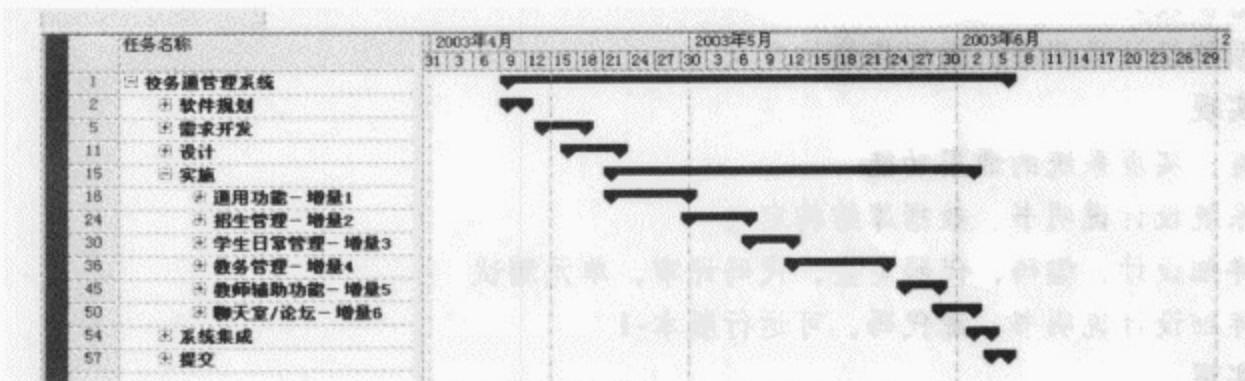


图4 进度计划

### 9. 项目成本估算

项目估算目的是为了确定项目所需的人力、时间以及项目完成过程中耗费的人力、物力、财力资源。图5是项目估算和预算的结果（详见估算计划）。

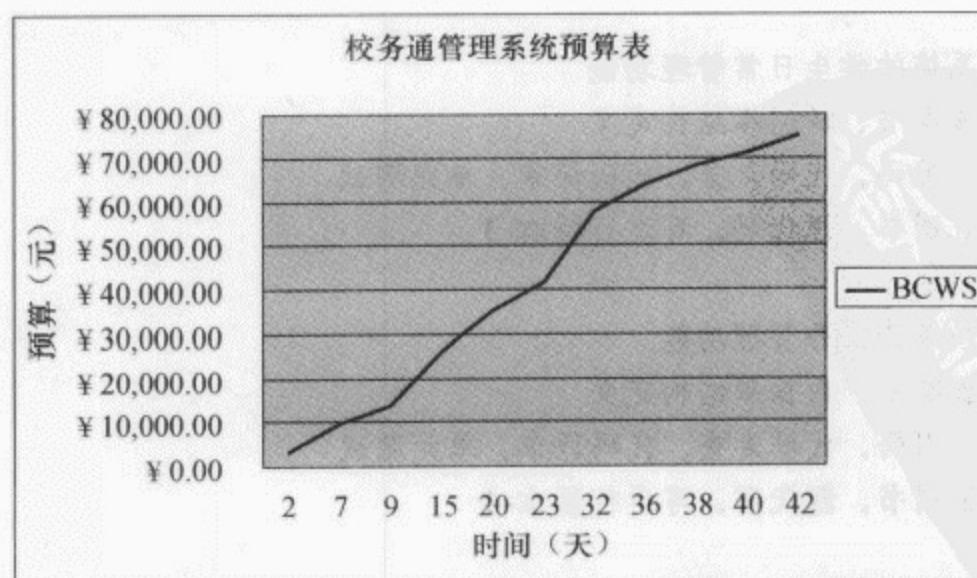


图5 成本计划

### 10. 质量管理计划

质量管理计划详见质量管理计划专题。

### 11. 配置管理计划

配置管理计划详见配置管理计划专题。

## 12. 项目风险计划

风险是指在项目进行过程中可能发生的事件，这些事件将会对项目按预期时间、资源和预算完成产生重大影响。风险分析的目标是识别这些事件，设法避免这些事件的发生并制定一旦这些事件发生后的处理措施。表2是本项目风险计划清单表。

表2 项目风险计划

排序	输入	风险事件	可能性	影响	风险值	风险应对措施
1	客户的SOW	需求不明确，增加需求，导致需求蔓延	70%	50%	35%	1. 采取加班的方法 2. 修改计划去掉一些任务 3. 与客户商量延长一些时间
2	合同	进度要求紧，合同金额有限	30%	50%	15%	可以请一些实习的学生做辅助工作，一来成本不高，二来可以加快进度
3	历史项目信息	开发人员对测试工作不重视	30%	40%	12%	1. 强制性要求每段代码保留测试单元，由SQA检查 2. 加入专门的测试人员
4	WBS	供货商、外商的质量问题	20%	50%	10%	多选择几个可以作为备份的外商和供应商
5	历史项目信息	开发人员的流动	15%	60%	9%	1. 注意项目团队的沟通，及时了解开发人员的动态 2. 控制好项目过程中的文档 3. 从其他的项目组借调人员 4. 从外部招聘有过此类开发经验的人员

## 13. 度量计划

详见度量计划专题。

## 14. 项目沟通与评审

项目评审的主要目的是根据项目计划对项目的执行活动进行检查，及时发现问题，研究解决对策，纠正偏差，保证项目的顺利实施。项目交流计划分为如下几类：

- 每天17:00的沟通交流
- 定期评审
- 阶段评审
- 事件评审

各类交流评审安排见表3。

表3 项目管理交流计划

评审类别	评审周期	评审要点	相关人员
日例会 定期评审 (周例会)	每天17:00~17:30 每周五	1. 不限定主题和内容，随意交流 2. 共享经验，避免错误 3. 本周工作进度 4. 问题及对策 5. 资源协调 6. 下周工作安排	项目组所有人 项目经理 开发经理 质量经理 配置管理员
阶段评审	阶段结束	1. 本阶段计划执行情况 2. 质量评审结果 3. 产品审计结果 4. 下阶段计划修正	项目主管 项目经理 开发经理 质量经理 配置管理员 市场人员
事件评审	当事件可能影响计划的执行	1. 事件性质和影响范围 2. 事件处理方案的讨论 3. 修改计划的评审时间	项目经理 开发经理 质量经理

## 11.4 小结

项目集成管理是为了实现项目目标，确保项目范围内的各项工作能够顺利协调的配合进行，消除项目管理中的局部性，平衡项目各个目标之间的冲突，保证项目过程各阶段的正确实施，所开展的以整体思想为指导，从全局出发，以项目总体利益最大化为目标，以统一协调各方面管理为内容进行全面管理的过程。它具有综合性、全局性和内外兼顾性的特征。本章是对前几章的综合说明，集成项目计划的完成是项目经理完成项目计划的标志。

## 11.5 练习题

### 一、选择题

1. 下列哪个不是项目管理计划的一部分？（ ）  
A. 里程碑图示      B. 进度      C. 数据库设计      D. 风险清单
2. 项目管理过程中的进度目标、成本目标、质量目标、范围目标等各个目标之间是（ ）。  
A. 相互独立      B. 相互关联和制约的      C. 进度目标最重要      D. 没有关系的

### 二、判断题

1. 进度和成本是关系最为密切的两个目的，几乎成了对立关系，进度的压缩一定要依靠增加成本实现，而成本的降低也一定要牺牲工期进度为代价的。（ ）
2. 项目无论大小，在编写项目集成计划的时候需要按照统一的标准模板进行编写。（ ）
3. 为了编写一个好的集成项目计划，项目经理应该通晓项目知识域的相关知识，与项目团队人员一起协作完成项目计划。（ ）
4. 为了节约成本，可以适当减低项目过程中的质量标准。（ ）
5. 项目管理过程是一个集成的过程，范围计划、进度计划，成本计划，质量计划，风险计划是相互联系的。（ ）
6. 所有的项目集成计划都按照国家的标准文档编写。（ ）
7. 一个好的项目计划的开发应该是渐近式的。（ ）
8. 项目的范围发生变化，也会影响项目的工期进度、成本、项目的质量。（ ）

## 第三篇

# 软件项目执行控制

完成项目计划后，需要按照项目计划执行项目，项目执行过程中，需要对项目的执行状况进行控制和管理，以保证项目按照预定的计划执行。进入本书路线图的第三阶段“软件项目执行控制”（如下图所示）。



# 第12章

## ■ 项目执行控制过程

项目计划和项目执行可以视为互相渗透、不可分割的活动。制定计划的主要职能就是用来指导项目实施工作，制定一个好的项目计划并在执行阶段很好地遵循它。

项目执行控制阶段主要是按照项目计划执行项目并监控项目性能，整合人力与其他资源，以实现项目计划。在项目执行控制过程中，项目经理要面临很多的问题，范围、进度、成本、质量、风险等等，所以，项目管理者需要制定一个可行的管理过程来保证项目执行。这个阶段包括集成变更管理、范围管理、时间管理、成本管理、质量管理、团队管理、沟通管理、风险管理以及合同管理等。

### 12.1 项目的集成变更管理

软件项目是一个系统，各个系统元素都不是彼此独立的，而是相互影响、互为联系、相辅相成的，所以，项目的执行控制过程是一个集成的管理过程。软件项目管理中涉及客户、客户需求、项目成员、资金、时间、技术、工具、场所等多种资源，只有确定各种资源的相互关系和影响，才能方便的利用既有的项目资源寻求一条达到项目目标（客户需求）的最小代价（cost）路径。

#### 12.1.1 集成管理

项目集成管理是项目组织对项目最具全面性的一项工作，它是以全局出发对整体项目中各专项计划进行平衡协调，以满足项目相关人员的利益要求和期望。进行项目执行控制的基本步骤如下：

1) 建立计划标准：即建立项目正确完成应该达到的目标，它是确保项目能够按照项目计划进行实施的具体执行任务的说明书，是合理进行过程控制的依据。

2) 观察项目的性能：建立项目监控和报告体系，确定为控制项目必要的数据。在项目计划实施过程中，为了便于管理和控制执行情况，必须做好项目计划实施记录，掌握好项目的实际进展情况。记录还可以为项目实施中的检查、分析、协调、控制、计划修订和总结等提供原始资料。

3) 测量和分析结果：将项目的实际结果与计划进行比较，掌握计划实施情况、协调各项工作、采取有效措施解决实施中出现的各种矛盾、调配资源以克服实施工作的薄弱环节、努力实现项目实施的动态平衡，从而保证项目计划目标的实现。

4) 采取必要措施：如果实际的结果同计划有误差时，采取必要的纠正措施，必要时修改项目计划。可以选用项目管理软件来协助项目执行过程，例如将项目工作进展在办公室张贴，标识参与者、可交付成果，这样对项目团队可以起到很好的激励效果。

5) 做好计划修订工作，控制反馈：计划不可能一成不变，当项目的内部和外部条件发生较大变化，项目计划就要根据项目实际情况进行更改，以控制计划的实时有效性。如果修正计划，应该通知有关人员和部门。

项目集成管理过程具体见图 12-1。首先设立基准计划，它是经过多方认可的计划。项目开始之后就会按照项目计划执行。在计划的执行过程中，不断从数据采集的信息库中获取各种数据信息，其中包括项目的性能信息，以及相关的变更信息等，并与计划进行比较，一方面观察项目是否按进度进行、成本如何、是否超支、质量是否达标等。另一方面是判断原来制定的项目计划是否合理。如果有偏差，应该标识偏差，同时，对偏差进行分析，预测对将来的影响，并提出纠正措施，同时评审纠正措施的方案，必要时修改项目计划，修改项目计划时，应该按照变更控制系统的过程执行。将修改后的项目计划通知相关人员。

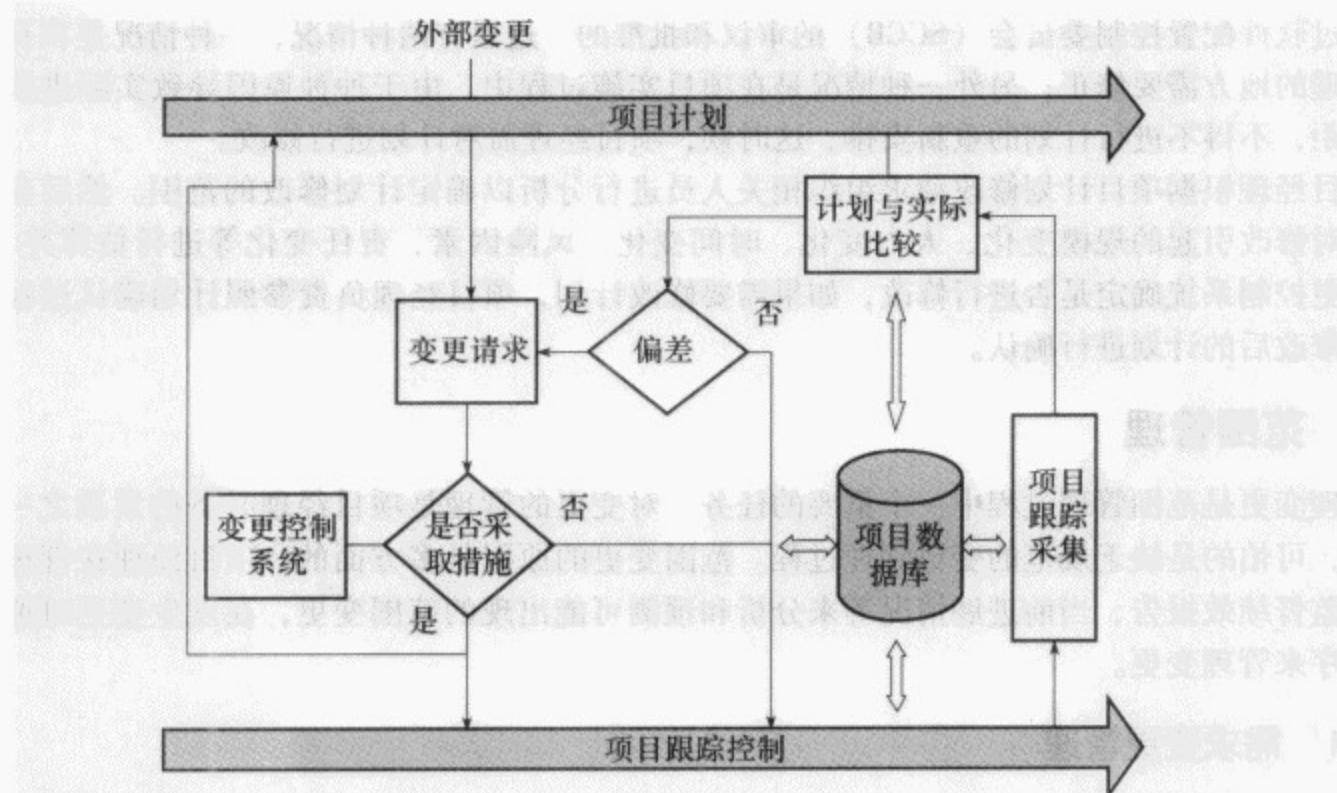


图 12-1 项目执行控制过程

项目控制过程的目的是保证项目目标的达成。进度、成本、质量控制是重要的监控内容。这个阶段需要对项目进行过程中的产品和活动进行审计和改进，项目经理需要保证项目沟通，组织和激励大家。项目管理者需要不断收集项目信息、度量分析项目，并发布项目运行状况。同时也需要分析趋势，以便决定改善的步骤。而变更是项目执行过程中不可以避免的，所以，需要管理好变更控制系统，执行批准的纠正措施，批准预防措施，批准缺陷的修复措施。必要时需要采购外包等合同管理。

只应急处理问题而不是提前观察各种征兆是项目控制中常见的问题，为了更好进行项目控制，需要遵循一定的原则才能很好的做好计划管理工作。项目是一个整体，要坚持系统管理的原则；项目计划是与各个专项计划紧密联系的，这就需要提高项目管理的透明度，管理要制度化，令行禁止，统一调配，合理进行适度授权，以便于提高项目团队的主动性、创造性。

为了控制好项目，必须做好以下事情：1) 充分了解项目当前的状态；2) 依据所期望的状态、当前的状态和目标做出一些决策。其中项目经理的分析能力、经验和悟性都决定了做出决策的质量，当然决策主要受到所了解信息的限制，如果掌握的项目状态数据不够准确或者不够完整，就不会做出最好的决策。

## 12.1.2 集成变更管理

项目控制过程中，变更管理对监控项目进展，防范风险起到重要作用，同时对发生的变化做出反应。

变更控制的目的就是为了防止配置项被随意修改而导致混乱。变更控制是在整个软件生命周期中对变化的控制和跟踪。变更控制记录每次配置项变化的相关信息，如变化的原因、变化的内容、变化的影响范围等等。变更包括项目执行过程中的各种变更，例如需求的变更、进度的变更、成本的变更、质量的变更、风险的变更、人员的变更等等。变更发生后要综合考虑，权衡关系。一个变更可能导致其他的变更。

变更控制是通过对变更请求（Change Request, CR）进行分类、追踪和管理的过程来实现的。在项目变动时要通知协调各方面项目干系人。

一个有序的变更控制过程，要确保合理地考虑每个变更所产生的影响，该过程确保只有被批准的变更才能予以实现，并放入相应的基线中，而且确保所有被批准的变更均实现，与进度和成本控制系统一起，它提供了技术活动和项目进展的可见性。变更控制的对象主要指配置库中的各基线配置项。

另外，在项目管理过程中计划的变更也是一个非常重要的基线变更。

计划不是“固定化”，常有人道，“计划赶不上变化”，但“要跟上变化”，就需要对计划进行调整。如果项目评审发现项目进展不符合计划且必须调整时，有必要进行项目计划的修改。这个调整一

定是经过软件配置控制委员会（SCCB）的审议和批准的。这里有两种情况，一种情况是项目计划制定有不合理的地方需要修正；另外一种情况是在项目实施过程中，由于种种原因导致实际进展与计划产生了差距，不得不进行计划的重新安排。这时候，项目经理需对计划进行修改。

项目经理根据项目计划修改请求组织相关人员进行分析以确定计划修改的范围。然后参照项目规划过程对修改引起的规模变化、人力变化、时间变化、风险因素、责任变化等进行估算并记录结果，通过变更控制系统确定是否进行修改，如果需要修改计划，项目经理负责参照计划确认过程与相关各方对于修改后的计划进行确认。

## 12.2 范围管理

管理变更是范围管理过程中一个重要的任务。对变更的管理是项目经理必备的素质之一。变更并不可怕，可怕的是缺乏规范的变更管理过程。范围变更的原因是多方面的。项目经理在管理过程中必须通过监督绩效报告、当前进展情况等来分析和预测可能出现的范围变更，在发生变更时遵循规范的变更程序来管理变更。

### 12.2.1 需求变更管理

需求的变化问题是每个开发人员、每个项目经理都可能遇到的问题，也是很头痛的问题，一旦发生了需求变化，你不得不来修改你的设计、重写你的代码、修改你的测试用例、调整你的项目计划等等。需求的变化好比是万恶之源，为项目的正常进展带来不尽的麻烦，因此，必须管理好需求的变更，使需求在受控的状态下发生变化，而不是随意变化，需求管理就是要按照标准的流程来控制需求的变化。需求变化一般不是突发的革命性的变化，最常见的是项目需求的蔓延问题。

根据以往的历史经验，随着客户方对信息化建设的认识和自己业务水平的提高，他们会在不同的阶段和时期对项目的需求提出新的要求和需求变更。用户总是有新的需求要项目开发方来做，有些客户认为软件项目需求的改变可以很容易地被实现。就像用户在“漫天要价”，而开发方在“就地还钱”。这个现象的本质问题就是需求变更所引发的。项目经理应该将重点放在对项目的控制方面，需求一扩大，就如同河堤出现了缺口，会越来越大，失去控制。

需求的变更可以发生在任何的阶段，即使到项目后期也可能发生变更（例如测试阶段，用户根据测试的实际效果会提出一些变更要求），后期的变更，会对项目产生很负面的影响。Standish Group 从 1994 年到 1997 年的 CHAOS Reports 证实，导致项目失败的最重要的原因与需求有关。

需求变更主要是因为在需求确定阶段，用户往往不能确切地定义自己需要什么。用户常常以为自己清楚，但实际上他们提出的需求只是依据当前的工作所需，而采用的新设备、新技术通常会改变他们的工作方式；或者要开发的系统对用户来说也是个未知数，他们以前没有过相关的使用经验。随着开发工作的不断进展，系统开始展现功能的雏形，用户对系统的了解也逐步深入。于是，他们可能会想到各种新的功能和特色，或对以前提出的要求进行改动。他们了解得越多，新的要求也就越多，需求变更因此不可避免地一次又一次出现。

项目经理一定要坚持一个最基本的原则：一般不要轻易答应这样的要求——你给我们做个什么。今天一个，明天肯定会有第二个，后天会有第三个……只要你答应一次，就会一个一个接踵而至，永远没有结束的时候，项目肯定会失去控制，也不可能最终验收。为了把需求控制在一定的范围，要避免与一般业务人员交谈，树立顾问的权威和信心，要以专家的姿态与客户接触。

管理需求变更应该处理好变更的请求，对需求的变更进行严格的控制，没有控制的变更会对项目的进度、成本、质量等产生严重的影响。说到变更，项目人员常常存在某种顾虑，其实变更并不可怕，可怕的是对变更束手无策，或者不采取任何的预防控制措施。对待变更的正确处理方法是，根据变更的输入，按照变更控制系统规定的审批程序执行，通过严格审查变更申请后，决定项目变更是否应该得到批准或者拒绝。

如果开发团队缺少明确的需求变更控制过程或采用的变更控制机制无效，抑或不按变更控制流程

来管理需求变更，那么很可能造成项目进度拖延、成本不足、人力紧缺，甚至导致整个项目失败。当然，即使按照需求变更控制流程进行管理，由于受进度、成本等因素的制约，软件质量还是会受到不同程度的影响。但实施严格的软件需求管理会最大限度地控制需求变更给软件质量造成负面影响，这也正是我们进行需求变更管理的目的所在。

需求变更管理的过程很大程度上就是用户与开发人员的交流过程。软件开发人员必须学会认真听取用户的要求、考虑和设想，并加以分析和整理。同时，软件开发人员应该向用户说明，进入设计阶段以后，再提出需求变更会给整个开发工作带来什么样的冲击和不良后果。有时开发任务较重，开发人员容易陷入开发工作中而忽略了与用户的随时沟通，因此需要一名专职的需求变更管理人员负责与用户及时交流。

对于需求变更控制可以采取商业化的需求管理工具，在数据库中存储不同类型的需求。这些工具提供了对每项需求的属性描述，状态跟踪等，并可以将需求与其他的相关工作产品建立关联关系。

总之，管理需求变更的主要控制方法是确定一个规范化的变更控制系统（流程），图 12-2 便是一个需求变更控制流程例子。

表 12-1 是需求变更的一个变更申请表格的实例，它是一个软件项目实施过程中，对需求提出的一个变更，这个变更发生在设计阶段，还没有进行具体的编程工作，按照变更控制系统的程序，首先提出需求变更请求，然后评估变更的影响，最后通过 SCCB 的表决，决定可以接受其中的一部分变更，另外一部分变更推到下一版本实现。

需求变更后需要重新修改 WBS。关于更多管理范围变更的内容可以参见配置管理章节的基线变更管理。需求是项目中的一个重要基线，所以需要变更管理也是一个重要的基线变更管理。

## 12.2.2 管理变更的策略

有一个实际的案例：某软件公司为某企业实施 ERP 系统。这个公司与两家公司争夺这个项目，在项目前期做出了巨大让步，没有对项目需求作出严格而系统的管理，在项目过程中也没有制定出切实可行的项目任务分解、成本估算、进度计划、配置计划等。项目计划、需求、实施内容一变再变，最终成本超过了整个项目款一百多万元，差点造成了公司解体。其实，这种情况对企业也不好，因为项目延期对企业的潜在影响也不小。而作为公司是绝对的输家，由于项目控制不力，不但没有在项目中赚到钱，而且还赔了 100 多万，资金紧张又造成了人员的流失，给企

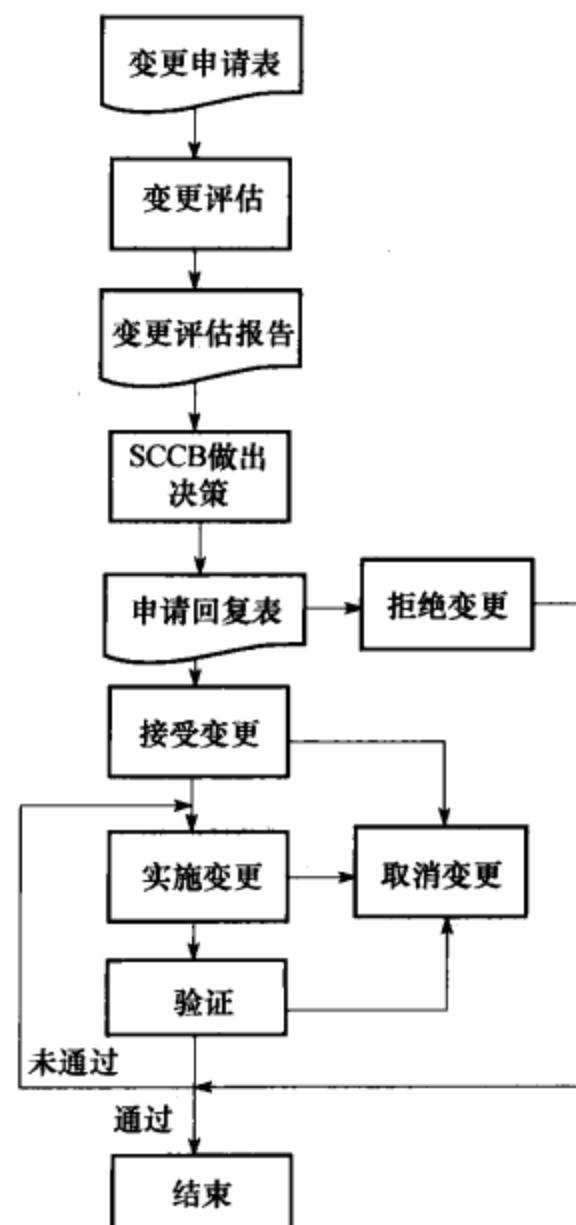


图 12-2 需求变更控制流程

表 12-1 需求变更提交单

软件基线产品修改提交单			
申请人	韩万江	申请日期	2007.10.12
项目名称	项目管理系统		
阶段名称	系统设计		
文件名称	RCR-PM-01.doc, RCR-PM-02.doc, 变更简述如下		
修改内容	1) 修改测试流程控制：将 2 个角色，3 个渠道流，改 为 3 个角色，4 个渠道流，详见 RCR-PM-01.doc 2) 增加开发人员技能信息库管理，详见 RCR- PM-02.doc		
验证意见 SCCB	同意 RCR-PM-01.doc 变更。RCR-PM-02.doc 的变更可 以推迟到下一个版本实施	验证人： 韩万江， 姜岳尊， 孙泉	验证日期 填表人 韩万江

业造成了巨大的伤害。这个项目失败的最主要原因是缺乏有效的需求变更控制系统。

在管理变更的时候应该采取一定的策略，例如：

1) 对照合同规定，发现有些变化是合同规定范围内的，在需求分析和设计阶段因疏忽造成的遗漏或者错误；有些变化是合同之外的，而这些变化又可以分成两种，一种会影响系统开发，另一种可以在系统开发之后再开发。针对这些分析，采用的策略是：合同范围之内的要求坚决修改；合同范围之外的、但影响系统开通的，也进行修改，但要通知客户；合同范围之外的可延后开发，要和客户商量并达成一致，在系统开通之后再进行开发。需求变更控制要遵循变更控制系统的规定。

需求变更给软件开发带来的影响有目共睹，所以在与用户签订合同时，可以增加一些相关条款，如限定用户提出需求变更的时间，规定何种情况的变更可以接受、拒绝接受或部分接受，还可以规定发生需求变更时必须执行变更控制流程。

2) 随着开发进展，有些用户会不断提出一些在项目组看来确实无法实现或工作量比较大、对项目进度有重大影响的需求。遇到这种情况，开发人员可以向用户说明，项目的启动是以最初的基本需求作为开发前提的，如果大量增加新的需求（虽然用户认为是细化需求，但实际上增加了工作量的新需求），会使项目不能按时完成。如果用户坚持实施新需求，可以建议用户将新需求按重要和紧迫程度划分档次，作为需求变更评估的一项依据。同时，还要注意控制新需求提出的频率。

3) 选用适当的开发模型 采用建立原型的开发模型比较适合需求不明确的开发项目。开发人员先根据用户对需求的说明建立一个系统原型，再与用户沟通。一般用户看到一些实际的东西后，对需求会有更为详细的解释，开发人员可根据用户的说明进一步完善系统原型。这个过程重复几次后，系统原型逐渐向最终的用户需求靠拢，从根本上减少需求变更的出现。目前业界较为流行的迭代式开发方法对工期紧迫的项目的需求变更控制很有成效。

4) 用户参与需求评审 作为需求的提出者，用户理所当然是最具权威的发言人之一。实际上，在需求评审过程中，用户往往能提出许多有价值的意见。同时，这也是由用户对需求进行最后确认的机会，可以有效减少需求变更的发生。

5) 对于客户的需求，我们要尽量地予以满足，但也不是一味地不顾技术实现上的困难而迁就客户的无理要求，在需求管理进行的同时，我们也不能忽略了成本问题。因为每一个功能的实现都需要我们花费时间去努力的。做需求管理的人员要和客户进行很好的沟通，在成本和需求之间找到平衡点。面对客户必须要保持涵养，客户可能是一点都不了解软件，所以就要求管理人员对客户的需求进行必要的补充说明（基本上应该站在为减少成本，提高质量的立场上）。

### 12.2.3 范围核实

范围核实和验证是贯穿项目生存期始终的过程。完成项目范围可能会有交付物的生成，这些交付物必须获得客户或者相关项目干系人的认可，当然，客户对项目的正式接收一般是项目收尾阶段最重要的任务。在项目终止时，范围核实过程应该记录项目完成的水平和程度。

## 12.3 进度、成本管理

项目计划中进度、成本是项目跟踪控制的主要内容。项目的进度和成本是关系最为密切的两个元素，几乎成了对立关系，进度的缩短常常要依靠增加成本来实现，而成本的降低也常常以牺牲工期进度为代价的。

### 12.3.1 进度管理

进度管理是一动态过程。有的软件项目需要一年，甚至需要几年的时间。一方面，在这样长的时间里，项目环境在不断变化；另一方面，实施进度和计划进度会发生偏差。因此在进度控制中要根据进度目标和实际进度，不断调整进度计划，并采取一些必要的控制措施，排除影响进度的障碍，确保进度目标的实现。

项目经常延期是很多项目的特点，因此项目实施中的过程检查重点是“进度检查”。在实际工作中，很多项目都是启动一段时间后就开始不停地加班，使整个团队处于疲惫状态，导致工作效率低下，最后把项目计划丢在一边。对于这种情况，比较好的做法是不断地检查项目计划与实际进度是否存在偏差，如果存在偏差则找出问题的根源，然后消除引起问题的原因。例如，可以调整进度安排或者增加人力投入，这样就避免了问题不断放大。

很多人可能会以为，进度管理是项目管理者的事情，而与团队的其他开发人员毫无关系，其实，开发人员和所有过程都应该是有关联的。进度管理有两点，一点是总体进度，另一点就是个人进度了，而项目进度却是建立在个人进度的基础上的。

总体的进度应该由项目经理来控制和调整，而个人的进度却是软件开发人员个人的责任和职责所在。开发人员很多时候都是以自己出发点进行思索，他们之间可能存在着相互间的沟通意识，但也只是个别人员之间的交流，并不能从根本上把握全体进度，也无法对进度作必要的分析和调整。项目经理需要从项目整体上控制进度，决定相应的对策。

项目经理必须在进度计划制定过程中动员所有项目干系人参与计划制定。制定现实的进度计划并在项目生命周期内留出一些应急储备是很重要的。

建议在软件开发的过程中，不管项目的大小我们都应该抽出时间来写一份个人的进度报告，而在个人的进度报告上面应该有清晰的个人进度，存在问题，准备如何解决等描述。个人进度管理也是软件开发人员的自我管理，是进度控制的最重要组成部分，个人进度的状况好坏直接影响到团队全体的进度推进状况。

进度跟踪中发现进度偏差的原因有很多，主要体现在：

1) 任务本身的估算问题。进度出现偏差首先要考虑工作量的估算是否合理，是否考虑了工作中存在的技术难点，是否考虑了项目成员自身的技能，是否考虑了其他应该考虑的风险。任务计划下达给项目成员时候应该获取承诺，但很多时候获取承诺是无用的，是否可以承诺或者是否能完成承诺跟项目成员的个人经验和技能有太大的关系。

如果项目任务中存在着技术攻关或技术难点需要解决，对于这种任务往往是很难估计工作量的，而且一旦在技术问题上被卡住往往对项目进度产生致命的影响。唯一的方式就是把无法预测和不透明的东西转换为透明，在项目开始之前就应该进行风险分析和应对，提前进行技术问题的预研，开发原型，积累相关的知识和经验。

准确的估算依赖于专家的经验，但专家的经验同样是依赖于历史项目和历史数据。估算问题的根源还在于对项目成员技能和生产率水平没有较清楚的认识，一个软件任务的完成往往存在着很大的个人生产率差异和进度差异。

2) 任务本身的粒度问题。对于一个小软件项目，出现几天的偏差容易得到纠正。而如果出现几周的偏差则很难再进行纠正。任务本身的粒度和工作量直接和偏差的大小相关。当任务本身的粒度太大的时候是不适宜进行跟踪的，任务本身是否会偏差不在取决于跟踪者，而是执行者对于大粒度的任务是否有很好的细分和自我控制能力。

如果我们的进度偏差最多允许一周，则需要把任务粒度细化到周，按周进行跟踪。如果对于任务最多允许偏差1~2天，则需要把任务粒度细化到天，按天来进行跟踪。细粒度的跟踪目的就是要消除不确定性因素和风险，尽可能早的发现任务中的问题，这样才有可能有时间来解决问题和纠正偏差。

有些人误认为在开发进度滞后的情况下，可以聘请更多的程序员加入到开发团队中，通过增加人力资源来赶上进度。因为尽管其个人能力很高，但是为了使其与大家一起协同工作，开发团队不得不分出人手对其进行与项目有关的技术/业务培训，更重要的（也是难度最大的）是还要引导其融入团队。这可能需要花费开发团队许多时间和精力，很有可能使项目进度更慢。

对于大粒度的项目任务，成员反馈的完成任务百分比有时是不可靠和主观的数据。项目成员的自我监控能力对进度是否偏差起到重要的影响，在这种情况下，任务是否能够按期完成对项目经理是不可控的，因此项目经理必须对成员有充分的了解和信任。

3) 项目经理对业务和技术领域的不熟悉。对于软件项目经理，对业务和技术领域的熟悉是必须具

备的能力。有了这些能力才可能和项目组一起开发出比较好的 WBS 分解和任务工作量的估算，有了这些能力才可能实现细粒度任务的划分，并定义清晰的出入口准则。

举个例子：项目经理给项目成员安排了一个任务，要求本周完成，到了周末项目成员反馈无法完成需要延期 2 天，项目经理就确认延期两天并调整后续任务。到了下周二，项目成员又反馈出现了新问题，有个细节没有考虑到还需要延期三天，项目经理不得已又进行任务调整。这就是我们常看到的场景，整个任务和项目计划都变得不可控制了。项目成员有责任，但项目经理同样有责任，项目经理在第一次出现偏差时候就应该介入任务或问题本身，帮忙一次诊断和分析问题，挖掘问题延期根源，或者调整任务粒度，改进监控方式，而这些都需要项目经理具备一定的业务和技术能力，具备相关的经验积累及时做出指导。

在第一次出现进度偏差的时候，项目经理需要的就是及时介入问题，查找问题根源而不是简单的关注成员反馈的下一个可能完成的时间点。所以，进度出现小偏差就立即查找根源并控制，而不是出现大偏差的时候进行应急。

项目总体进度允许偏差确定了项目任务粒度划分和任务跟踪频度。很多进度问题是前期没有进行充足风险分析和提前应对。估算很重要，一份不切实际的进度再怎么跟踪都只可能延期或低质量。任务完成百分比不可靠，可靠方法是任务细分并定义严格的出入口准则。第一次延期就应该介入问题，查找根源而不是乐观期待下一个可能完成点。

作为项目经理不仅仅要把握全体的进度，还要把握住开发的成本，如果开发的成本超出了，那开发的项目就不能盈利，而不能盈利的开发也就意味着失败。

### 12.3.2 成本管理

项目成本管理是为了确保项目在预算内按时、保质、经济高效地完成项目目标而展开的一种必要管理活动，必须要加强项目实际成本的控制，一旦项目成本失控，就很难在预算内完成项目，不良的项目成本控制常常使项目处于超预算的危险境地。在实际实施过程中，预算超估算，决算超预算现象还是屡见不鲜的，因此，进行成本管理是必需的。项目成本控制，就是在项目实施过程中按照预先制定的目标计划，通过适当的技术和管理手段，使项目成本自始至终置于有效的控制之下。在整个过程中，为了做好成本的控制，应该就每一项任务进行严格的成本核算，确保一切开支都控制在计划成本内，并尽可能的降低成本和消耗。

成本控制是落实成本计划的实施，保证成本计划在过程中得到全面、即时和正确的执行。需要进行成本核算，即根据完成项目实际发生的各种费用来计算，要动态地对项目的计划成本和实际成本、直接成本和间接成本进行比较和分析，找出偏差并进行相应的处理。在计划成本之内。最终实现高质量、确保成本目标的实现，就是要从影响成本的因素着手，制定相应的组织措施、技术措施和经济措施。

项目上的反反复复，开发人员加班费的支出，不仅加大了开发的费用，同时也给员工带来了身体、精神上的双重疲惫，直接导致个人抵抗力、免疫力的下降，更可能会造成员工身体上的隐藏疾病。同时也带动着相应的管理费用也随之增加了。这些都会使得我们软件的成本增加。

软件项目中的人力成本是总成本的主要部分，项目实际的人力成本决定了赢利的水平。在实际工作中可能会发现：预算时项目的利润很高，但最后核算总体利润时却赔本。这是因为，应用开发项目的人力成本很难估算准确，很多项目为了质量和进度要求，执行中都会不断追加人力，最后使用的人力资源大大超出了预算。因此软件公司必须核算项目人力成本以控制项目的人力资源投入。具体要做的就是：在做项目预算时就应该明确需要的人力资源总数，执行中要记录实际使用的人力资源，项目结束时核算一个项目到底是赚了还是赔了。特别是一些利润水平低风险又大的项目，可能只要多投入一个人月项目就赔了，因此项目过程中要动态监控人力投入情况并与预算进行比较，一旦发现超出预算就应即时处理。

项目进度、成本的跟踪控制过程是根据跟踪采集的进度、成本、资源等数据，并与原来的基准计划比较，对项目的进展情况进行分析，以保证项目在可以控制的进度、成本、资源内完成。常用的项目性能分析方法有图解控制法，挣值分析（已获取价值）法等。

### 12.3.3 图解控制法

图解控制方法是利用表示进度的甘特图，表示成本的累计费用曲线图和表示资源的资源载荷图对

项目的性能进行分析的过程。

从甘特图可以看出计划中各项任务的开始时间、结束时间，通过甘特图可以看出计划进度和实际进度的比较结果。

累计费用(S)曲线是项目累计成本图，将项目各个阶段的费用进行累计，得到平滑的、递增的计划成本和实际支出的曲线。累计费用(S)曲线对于监视费用偏差是很有用的，计划成本曲线和实际支出曲线之间的高度差表示成本偏差，理想情况下，两条曲线应当很相似。

资源载荷图，显示项目生存期的资源消耗，项目早期处于启动，资源使用的少，到了中期，项目全面展开，资源大量被使用，而在结束阶段，资源消耗再次减少。利用资源分配表的总数很容易构造资源载荷图，资源载荷图围住的面积代表某段工作时间的资源消耗。例如如果实际资源载荷图面积大于计划资源载荷图面积，说明我们已经比计划投入了更多的人工。反之，说明我们已经比计划消耗的人工日少，如果相近说明资源偏差很小。

图12-3是一个项目的甘特图、累计费用曲线和资源载荷图的图示说明：从甘特图中可以看到，项目按照计划开始，但是每项任务所用时间都比计划长，从费用曲线上看，实际费用比计划少了一些，但是不要以为是成本节约了，因为从资源载荷图可以看出是因为没有使用足够的资源导致实际费用没有计划的多。所以，可以说进度推迟的原因是缺乏足够的资源（人力资源和设备资源）。

再看图示12-4，从进度甘特图看，项目进度按照计划进行，如果不仔细分析，认为项目进展得很好，然而从费用曲线和资源载荷图可以知道保持进度是以大量费用为代价的，费用高的原因是资源超出计划很多。

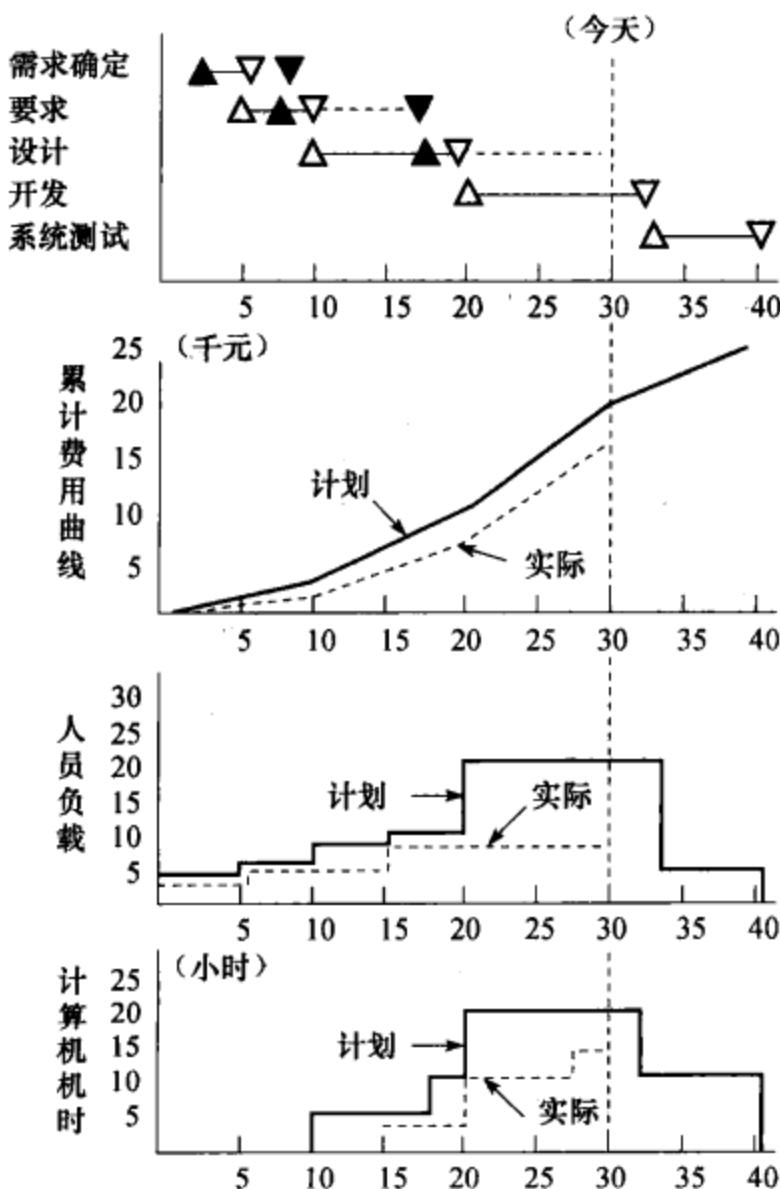


图12-3 项目图解控制图例1

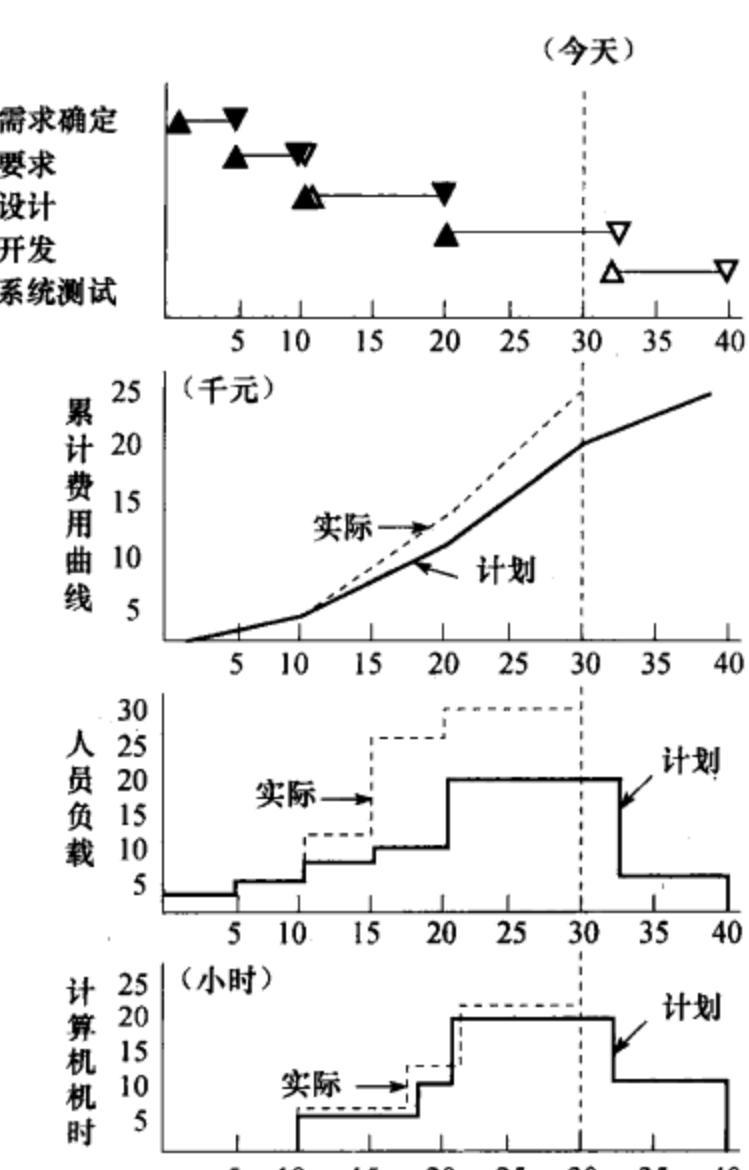


图12-4 项目图解控制图例2

用图解控制方法进行项目分析的时候应该对甘特图、累计费用曲线图和资源载荷图共同进行监控项目。这个方法可以给项目经理提供项目进展的直接信息。

图解控制方法的优点是可以一目了然的确定项目状况，采用这种易于理解的方法，既可以向上级管理层，也可以向项目人员报告项目的状况。它的最大缺点是只能提供视觉印象，但本身并不能提供其他重要的量化信息。例如，相对于完成的工作量预算支出的速度，每项工作的预算和进度中所占的份额，或者完成的工作百分比等。而挣值分析（也称为已获取价值分析）可以提供这方面的功能，而且还可以提供更多量化的信息，无论是大型项目还是小型项目，都可以采用这个挣值分析法。

### 12.3.4 挣值分析法

挣值分析也称为已获取价值分析，是对项目实施的进度、成本状态进行绩效评估的有效方法，是计算实际花在一个项目上的工作量，以及预计该项目所需成本和完成该项目的日期的一种方法。该方法依赖于被称为“已获取价值”的一种主要测量。

挣值分析法是利用成本会计的概念评估项目进展情况的一种方法。传统的项目性能统计是将实际的项目数据与计划数据进行比较，计算差值，判断差值的情况。但是，实际的执行情况可能不是这样简单，如果实际完成的任务量超过了计划的任务量，那么实际的花费可能会大于计划的成本，但不能说成本超出了。所以，应该计算实际完成结果的价值，这样就引出了已获取价值的概念，即到目前为止项目实际完成的价值。有了“已获取价值”就可以避免只用实际数据与计划数据进行简单的减法而产生的不一致性。挣值分析（已获取价值分析）方法的模型如图 12-5 所示。

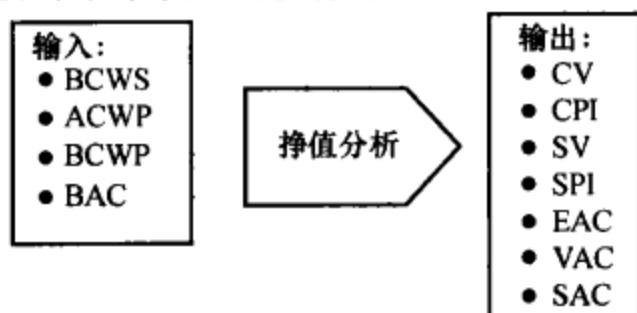


图 12-5 挣值分析模型

其中的输入：

- BCWS(Budgeted Cost of Work Scheduled) 计划完成工作的预算成本：是到目前为止的总预算成本。它表示“到目前为止原来计划成本是多少？”或者说“到该日期为止本应该完成的工作是多少？”，它是根据项目计划计算出来的。
- ACWP(Actual Cost of Work Performed) 已完成工作的实际成本：是到目前为止所完成工作的实际成本，它说明了“到该日期为止实际花了多少钱”，可以由项目组统计。
- BCWP(Budgeted Cost of Work Performed) 已完成工作的预算成本，又称已获取价值：是到目前为止已经完成的工作的原来预算成本，它表示了“到该日期为止完成了多少工作？”
- BAC (Budgeted At Completion) 工作完成的预算成本：是项目计划中的成本估算结果。是项目完成的预计总成本。

结合图 12-6 说明挣值分析的原理，截止到当前日期，图中  $BCWS = 10$  (万)， $ACWP = 11$  (万)， $BCWP = 12$  (万)，也就是说计划成本是 10(万)，为此已经付出的实际成本是 11(万)，而完成任务的价值（已获取价值）是 12(万)。费用差异为  $CV = BCWP - ACWP = 12 - 11 = 1$  (万)，表示成本比计划有节余；进度差异为  $SV = BCWP - BCWS = 12 - 10 = 2$  (万)，表示实际进度比计划快，多完成 2(万) 工作的任务。

如果项目一切正常的话， $ACWP$ 、 $BCWP$ 、 $BCWS$  应该重合或接近重合。

模型分析如下

- 1) 进度差异： $SV(Schedule Variance) = BCWP - BCWS$ 。  
若此值为零，表示按照进度进行；如果为负值，表示项目进度落后；如果为正值，表示进度超前。
- 2) 费用差异： $CV(Cost Variance) = BCWP - ACWP$ 。  
若此值为零，表示按照预算进行；如果为负值，表示项目超出预算；如果为正值，表示低于预算。

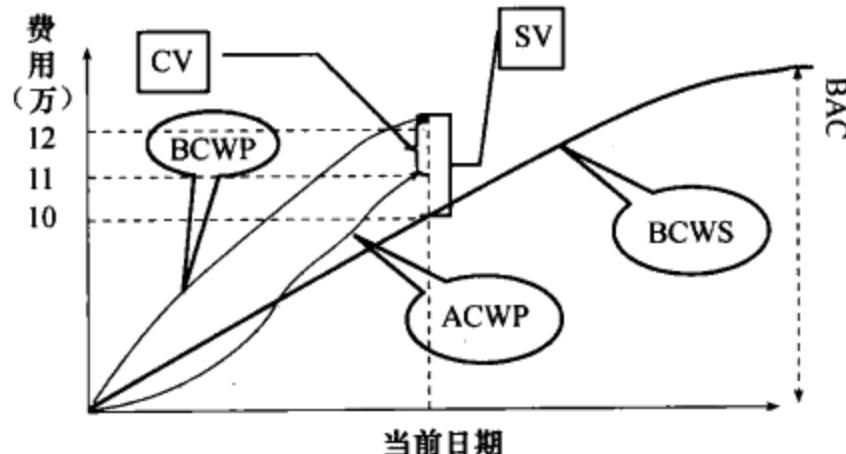


图 12-6 挣值分析原理

3) 进度效能指标: SPI(Schedule Performance Index) = BCWP/BCWS。

表示完成任务的百分比。若此值为1, 表示按照进度进行; 如果小于1, 表示项目进度落后, 如果大于1, 表示超进度进行。

4) 成本效能指标: CPI(Cost Performance Index) = BCWP/ACWP。

表示花钱的速度。若此值为1, 表示按照预算进行; 如果小于1, 表示项目超出预算; 如果大于1, 表示低于预算。研究表明: 项目进展到20%左右, CPI应该趋于稳定的, 如果这时CPI的值不理想的话, 应该采取措施, 否则这个值会一直持续下去的。

5) 项目完成的预测成本: EAC(Estimate At Completion) = BAC/CPI。

6) 项目完成的成本差异 VAC(Variance At Completion) = BAC - EAC。

7) 项目完成的预测时间: SAC(Schedule At Completion) = 计划完成时间/SPI

对于这些基本参数值, 可以按照一定的时间段来计算(比如说一周)。这个方法的难点在于计算BCWP, 但是如何计算已获取价值是值得研究的问题, 可以归结为两种计算方法: 方法一是自下而上, 这种方法比较的费时费力, 需要专门的人员及时连续的计算开发出来的产品的价值; 方法二是公式计算方法, 如果没有比较简单又实用的公式, 这个方法也存在问题。目前, BCWP公式计算方法通常可以采用一些规则计算, 主要是50/50规则, 0/100规则, 或者其他的经验加权法等等。最常用的方法是采用50/50规则, 50/50规则是当一项工作已经开始, 但是没有完成时, 我们就假定已经实现一半的价值, 当这个工作全部完成的时候才实现全部的价值; 0/100规则是当一项工作没有完成时, 不产生任何价值, 直到完成时才实现全部的价值。例如如果一个任务的成本是100元, 这个任务没有开始前价值为0。当采用50/50规则时, 如果这项任务开始了但没有完成, 我们就假设已经实现了50元的价值, 不管是完成了1%, 还是完成99%, 都认为实现了50元的价值, 直到这项任务全部完成, 我们才说它实现了100元的价值。如果用0/100规则, 直到项目全部完成, 我们才认为任务实现了100元价值, 除此之外, 这个任务没有任何价值。

图12-7可以说明50/50规则的计算原理, 按照此规则可以计算: 截止到今天A、B、C任务已经完成, 即实现了300美元工作量的价值, 而D任务开始了但是没有完成, 所以它实现了50美元工作的价值。所以, 截止到今天项目总共已经实现了预算的400美元工作的350美元的价值, 即已经完成的工作量价值是350美元, 称为“已获取价值”, 即 $BCWP = 350$ 美元, 而 $BCWS = 400$ 美元, 那么, $SPI = 350/400 = 87.5\%$ : 表明项目已经实现了87.5%。假设实际的花费是700美元, 实现了350美元的工作量价值, 说明, 实际每支出1美元实现50美分的价值, 即 $CPI = 350/700 = 50\%$ 。如果项目总预算是 $BAC = 1000$ 美元, 按照目前的花钱速度, 这个项目最终的费用将是 $EAC = 1000/0.5 = 2000$ 美元。所以这种方法可以告诉我们一种计算已完成工作百分比的方法, 还可以帮助我们分析成本支出的速度。使用50/50规则的前提是任务分解的足够细。

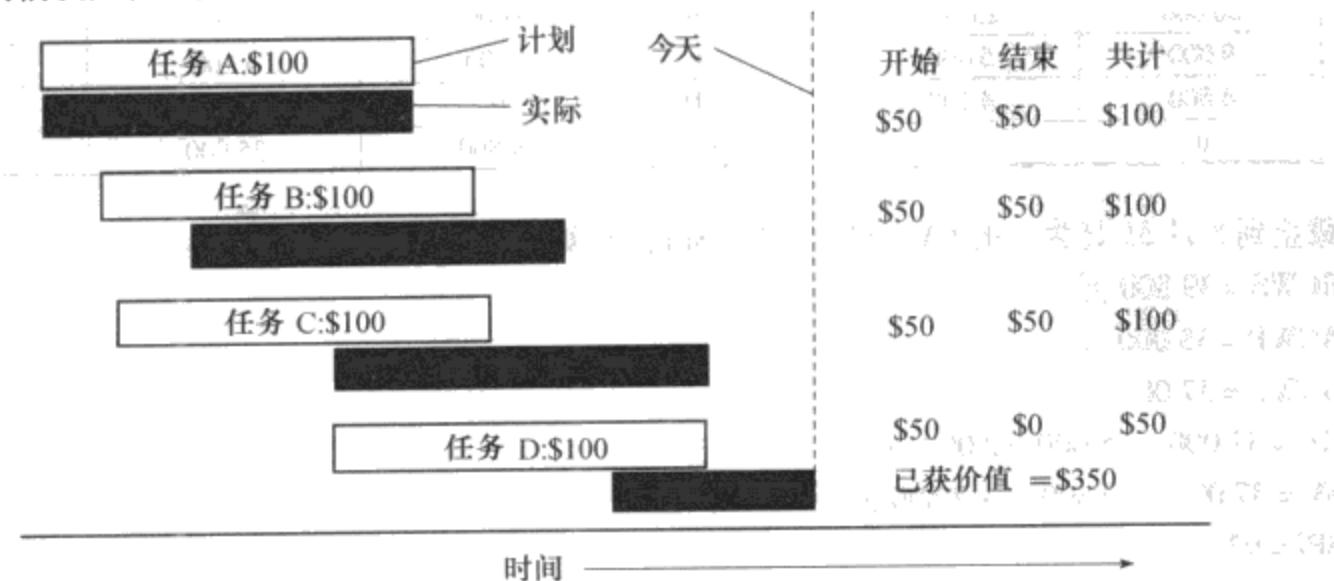


图12-7 50/50 规则

当然，我们可以采用保守 0/100 规则，例如一个 100 元价值的任务，无论是完成任务的百分之几，都认为是没有实现任何价值，直到全部完成才实现 100 元的价值。如果对任务非常了解的话，也可以采用经验加权的方法进行已获取价值的计算。

根据挣值分析原理继续分析图示 12-7，按计划今天应该完成 400 美元的任务，可是目前只完成了其中的 350 美元的任务，说明进度差异是 -50 美元，它表明进度落后，还差 50 美元的任务没有完成；而实际完成 350 美元的任务花了 700 美元钱，所以，成本差异是 -350 美元，说明超出预算 350 美元。进度效能指标是 87.5%，成本效能指标是 50%，也就是说花钱的速度是预算的 2 倍。

表 12-2 中四个任务的数据是截止到目前为止的性能数据：可以看到任务 1 按照计划执行中，任务 2 一定有问题，进度落后、成本超出预算，任务 3 和任务 4 可能有问题，也可能没有问题，所以，分析偏差的时候一定要分析 CV、SV 都是负值的情况，因为这种情况一定出现了问题了。如果 CV、SV 有正有负需要进一步分析 SPI、CPI 等参数以决定是否有问题。

表 12-2 性能分析结果

任务	BCWS (元)	ACWP (元)	BCWP (元)	CV (元)	SV (元)
1	100	100	100	0	0
2	100	100	70	-30	-30
3	100	120	110	-10	10
4	100	70	80	10	-20

**【例 1】** 你被指定负责一个软件项目，这个项目由 4 部分组成（A、B、C、D），项目总预算为 53000 元，其中 A 任务为 26000 元，B 任务为 12000 元，C 任务为 10000 元，D 任务为 5000 元。截止到 8 月 31 日，A 任务已经全部完成，B 任务过半，C 任务刚开始，D 任务还没有开始，表 12-3 是截止到 8 月 31 日的计划成本和实际花费，采用 50/50 规则计算截止到 8 月 31 日为止的 CV、SV、CPI、SPI？

计算 CV、SV、CPI、SPI 的关键是正确计算 BCWP，由于采用 50/50 规则：A 任务已经全部完成，所以其 BCWP 为 26000 元；B 任务过半，C 任务刚开始：根据 50/50 规则：不管完成多少，只要是任务开始，但是没有完成，我们认为实现了 50% 的预算价值，所以，B 任务的 BCWP = 6000 元，C 任务 BCWP = 5000 元，D 任务还没有开始，则 D 任务 BCWP = 0（见表 12-4）。

表 12-3 截止到 8 月 31 日计划  
成本和实际花费

任务	BCWS (计划费用：元)	ACWP (实际花费：元)
A	26 000	25 500
B	9 000	5 400
C	4 800	4 100
D	0	0

表 12-4 计算 BCWS、ACWP、BCWP

任务	BCWS (计划费用：元)	ACWP (实际花费：元)	BCWP (已获取价值：元)
A	26 000	25 500	26 000
B	9 000	5 400	6 000
C	4 800	4 100	5 000
D	0	0	0
总计	39 800	35 000	37 000

截止到 8 月 31 日为止的 CV、SV、CPI、SPI 的计算结果如下（见表 12-4）：

$$BCWS = 39800 \text{ 元}$$

$$ACWP = 35000 \text{ 元}$$

$$BCWP = 37000 \text{ 元}$$

$$CV = 37000 - 35000 = 2000 \text{ 元}$$

$$SV = 37000 - 39800 = -2800 \text{ 元}$$

$$SPI = 93\%$$

$$CPI = 106\%$$

SPI 小于 1 说明截止到 8 月 31 日没有完成计划的工作量，即进度落后一些；但是 CPI 大于 1，即截

止到8月31日费用节省了，完成工作量的价值大于实际花费的价值。

注：不可以根据表12-4中BCWS的值来计算BCWP，表12-4中BCWS的值是截止到8/31的预算值，而不是A、B、C、D任务本身的预算值，采用50/50规则计算BCWP时，应该根据任务总预算来计算已获取价值。

**【例2】** 表12-5是一个小型软件开发项目的实施阶段的计划，根据项目管理的要求，每周末计算本周应该完成的计划工作量和在本周末已经完成的任务。一个好计划应该是渐进完善的，随着项目的展开，计划也会随之细化和完善的计划。同时按照经验加权规则（经验百分比）计算已获取价值，如表12-7所示。表12-7的已获取价值每周计算一次，BCWP随着项目进展而每周有所增加。任何一周，可以从表12-6得到BCWS，从表12-7得到BCWP。ACWP是已经付出的实际工作量（单位是人天）。将实际周数乘上5（每周工作5天），如果开发人员是全职工作，再乘上人数，即为实际工作量（单位是人天）。根据人员成本参数可以计算相应的货币价值。例如500元/人天，则3个工作日的价值是1500元。

现在，我们分析一下截止到第三周的项目性能情况。由于项目人员不是全职在这个项目中，所以需要计算实际的工作时间，如果实际花在这个项目只有9人天（工时），则截止到第三周的性能数据如下：

- $BAC = 31$  (人天)
- $BCWS = 7$  (人天)
- $BCWP = 6.5$  (人天)
- $ACWP = 9$  (人天)
- $SV = BCWP - BCWS = -0.5$  (人天)，进度落后0.5人天的工作量
- $SPI = BCWP/BCWS = 92.8\%$ ，以计划进度的92.8%效能工作
- $CV = BCWP - ACWP = -2.5$  (人天)，超出预算2.5人天（如果500元/人天，则超出预算1250元）
- $CPI = BCWP/ACWP = 72.2\%$ ，以超预算27.8%的状态在工作
- $EAC = BAC/CPI = 43$  (人天)，因为预算是 $BAC = 31$ 人天，按照目前的工作性能 $CPI = 72.2\%$ 计算EAC是43人天
- $VAC = BAC - EAC = -12$  (人天)，超出预算12人天的工作量（如果500元/人天，则超出预算6000元）
- $SAC = 10/SPI = 10.8$  (周)，因为计划的完成时间是10周，按照目前的工作进度效能估算完工时间为10.8周。

表12-5 项目实施阶段的计划

任务	计划工作量(人天)	估计完成的周数	负责人
规划	3	1	章一
需求规格	2	2	王二
软件设计	10	5	章一，李三
测试计划	3	6	章一
编码	5	7	王二
单元测试	3	8	章一
集成测试	2	9	王二
Beta 测试	3	10	李三
总计	31		

表12-6 细化的项目计划

周	任务	累计计划工作量(人天)	BCWS(人天)
1	规划	3	3
2	需求规格	5	5
3	软件设计	7	7
4		11	11
5		15	15
6	测试计划	18	18
7	编码	23	23
8	单元测试	26	26
9	集成测试	28	28
10	Beta 测试	31	31

表12-7 截止到第三周的已获取价值(BCWP)

任务	任务工作量(人天)	完成百分比	已获取价值BCWP(人天)
规划	3	100	3
需求规格	2	50	1
软件设计	10	25	2.5
测试计划	3	0	0
编码	5	0	0
单元测试	3	0	0
集成测试	2	0	0
Beta 测试	3	0	0
总计	31		6.5

从上面的数字可知，这个项目将推迟 0.8 周（4 工作日）左右，超出预算 27.8%，完成预算比较困难。这时可以研究一下“为什么花费了比计划多的工作量？”，“是否因为不是全职工作影响了工作效率？”，“还是对任务缺乏了解？”，抑或是“应该做了更多其他的任务，没有统计上来，可能以后任务会完成的很快，而且没有想象的那样糟糕”，还是“计划做的不够科学”。经过分析找出原因，然后解决，如果问题解决了，也无法按照计划进行，就有必要更正计划以适应项目今后的情况。如果是因为计划做的不够科学，就必须修正计划。

### 12.3.5 偏差管理

项目经理在做项目计划的时候已经对项目进行了预算，而且经常是按照预算内完成、低于预算完成或者超预算完成来评价项目的成败。超预算会给项目经理以及公司带来很严重的后果，一个根据合同得到经费的项目，经费超支可能会导致经济损失。看到预算的重要性，就不难理解为什么很多的软件企业都很重视这方面的管理。项目管理上面临的一个尴尬的现实是常常受到超支的威胁，控制预算就是一个非常重要的管理过程，对于超出控制容许偏差范围的，就要引起重视，应该调查偏差发生的原因。

进度问题，更是项目经理关注的问题，进度冲突常常是项目管理过程中最主要的冲突，进度落后是危机项目成功的一项非常重要的因素。所以，一般在安排进度的时候，很多的项目经理对团队成员的要求是先紧后松，先让开发人员有一定的紧张感，然后在实施的过程中做适度的调节，以预防过度的紧张。记住一个经验教训：最会误导项目发展、伤害产品质量的事情就是过分重视进度，这不仅打击人员的士气，而且还会逼迫组员做出愚昧的决定。对上层管理或者客户的进度沟通，基本上是先松后紧，给项目留出一定的余地，然后在实施过程中，紧密控制。保证进度的偏差在可以控制的范围内。

一旦发现项目已经落后于进度或者超出预算，下一个问题是“如何做才能回到正常的轨道上来？”，为此，需要计算两个有待完成的效能指标 TCPI 和 TSPI。有待完成的效能指标是指为了达到预想的成本进度目标，必须在项目期间怎么做。若待完成的成本的效能指标 TCPI 大于 1，则必须做的比计划要好，才能达到目标，如 TCPI 小于 1，则表明做的比计划差一些也可以达到目标的。 $TCPI = (BAC - BCWP) / (Goal - ACWP)$ ，其中，Goal 是项目希望花费的数额，或者我们预期将花费的数目。仔细考察一下可以看出，分子是还有多少工作要做，分母是还有多少钱可以花费，也就是要保证将来不超出这些花费，必须以什么方式来工作。如果 Goal 就是 EAC，那么也就是说，如果不改变目前的效能，EAC 就是最终成本的正确估计值。

同理，可以给出待完成的进度的效能指标 TSPI 的计算公式。

图 12-7 的例题中， $CPI = 50\%$ ，严重偏离，那么要保证成本与计划的一样，即希望将来的最终成本为计划的 1000 元，则  $TCPI = (1000 - 350) / (1000 - 700) = 2.17$ ，即将来 1 元钱应该产生 2.17 元的工作量价值，才能保证成本控制在 1000 元，而不是 2000 元。

偏差如果超出一定的范围，可以根据变更控制系统提出变更申请，对进度或者成本等基准计划进行变更。

项目进度滞后的主要原因，往往是项目的范围、成本变化以及对项目风险分析不足，对项目要素的理解掌控能力不够造成的。例如：

1) 因为对项目的范围没有做明确透彻的分析和定义，致使项目在执行当中作了许多额外的工作。项目管理者对项目的范围未作深入细致分析，未和相关责任人做详细讨论，或未作明确说明和定义，就开始启动项目，而埋下隐患。

2) 因为对项目所涉及的资源、环境、工具等的成本分析不够完善准确，致使项目实施过程中遇到资源、环境、工具的限制，而不得不以时间作代价。

3) 对于项目的质量不够重视,或者说不具备质量管控的能力,导致项目执行过程中不断出现质量问题,活动安排顺序部分失控或者完全失控,项目进度管理计划形同虚设。项目进度失去控制。

4) 许多项目的风险分析并未引起项目管理者的足够重视,项目风险的管控能力其实更多的体现项目管理者的专业素质和从业经验。许多风险之所以不能被预先识别,或者给予足够重视主要原因是受制于项目管理者的个人能力。然而,给与项目进度影响最大的就是“风险”。因为项目管理者想不到的事情太多,因此项目实施过程中的意外问题接踵而至,不断需要应对这种所谓的“风险”。

5) 关于对项目进度影响的主要因素中还有一个不经常为人所重视的就是“项目组成员的职业素养”。在项目中真正专注于自身工作,对工作精益求精,对自己的质量、自身形象负责,应该是项目成员的职业素养。

为了更好地管理项目的进度,可以采用一些方法,例如:

1) 设立里程碑。任何一个项目都是由若干个相对独立的任务链组成的,只有在任何一条链都已经优化的基础上,才可能进行系统的优化。因此,保证每条任务链的效率是整个项目进度优化的前提和基础。通常,可以采用设置“里程碑事件”的方法来保证单独任务链的最优。

“里程碑事件”的目的就在于将一个过程性的任务用一个结论性的标志标出来的,从而使得任务拥有明确的起止点,这一系列的起止点就成为引导整个项目进展的“里程碑”。

在项目进度跟踪的过程中,给予里程碑事件足够的重视,往往可以起到事半功倍的效用,只要能保证里程碑事件的按时完成,整个项目的进度也就有了保障。

2) 关注薄弱环节,实现动态平衡。项目的进度管理并不是一个静态的过程,项目的实施与项目的计划也是互动的,在项目进度的管理过程中,需要不断调度、协调,保证项目的均衡发展,实现项目整体的动态平衡。

项目的关键路径始终是项目经理最为关心的,但随着项目的实施,关键路径可能会由于一些情形而发生变化,项目的拖延可能导致原来不在关键路径上的任务成为关键路径的必经之路。因此,团队成员需要随时关注项目进展,跟踪项目的最新计划,确保关键路径上任务的进度。

3) 明确每个成员的责任。对于项目中相对独立的关键任务组可采用专项承包的方式,设立子项目,再明白一点,就是定任务、定人员、定目标,进一步明确责任,确保关键任务的进度。

从普遍意义上说,应当根据项目的特点,建立项目组织的各种责任制度,将进度计划指标的完成情况与部门、单位和个人的利益分配结合起来,做到责权利一体化。

对项目的进度和成本进行监控的同时也应该对项目的质量、风险、人员等方面进行监控,只有它们的指标在计划的控制范围之内,项目的进度和成本控制才有意义。

## 12.4 质量管理

质量管理是管理者在对软件质量进行一系列度量之后做出的各种决策,促使软件产品符合标准。

质量的重要性已经在各个领域得到了广泛的认同。在软件项目的开发过程中必须及时跟踪项目的质量计划,对软件质量的特性进行跟踪度量,以测定软件是否达到要求的质量标准。通过质量跟踪的结果来判断项目执行过程的质量情况,决定产品是否可以接受,还是需要返工或者放弃产品。如果发现开发过程存在有待改善的部分,应该对过程进行调整。

在软件项目质量管理中,质量管理总是围绕着质量保证过程和质量控制过程两方面。这两个过程相互作用,在实际应用中还可能会发生交叉。质量保证是在项目过程中实施的有计划、有系统的活动,确保项目满足相关的标准。质量控制指采取适当的方法监控项目结果,确保结果符合质量标准,还包括跟踪缺陷的排除情况。质量管理不是一次性的事件,而是持续不断的过程。

### 12.4.1 质量保证

质量保证活动的一个重要内容就是报告对软件产品或软件过程评估的结果,并提出改进建议。质量保证通常由质量保证部门或有类似名称的组织单位提供。这种保证可以向项目管理小组和组织提供

(内部质量保证)，或者向客户和其他没有介入项目工作的人员提供(外部质量保证)。

质量保证(SQA)的三个要点是：

- 在项目进展过程中，定期对项目各方面的表现进行评价。
- 通过评价来推测项目最后是否能够达到相关的质量指标。
- 通过质量评价来帮助项目相关的人建立对项目质量的信心。

质量保证是为了确保对项目进行客观公正的审核和评价，在软件开发过程中，质量保证的主要任务是对项目执行过程和项目产品进行检查，验证它们与项目采用的过程和标准的一致性，这两项任务是项目审计活动。项目审计(audit)是对过程或者产品的一次结构化的独立评估，将审核的主体与为该主体以前建立的一组规程和标准进行比较，目的是确保真正的遵循了这一个过程，产生了合适的项目产品和精确反映实际项目的报告，项目审计可以事先规划，也可以是临时决定的。

所以质量保证的主要活动是项目产品审计和项目执行过程审计。

项目产品审计过程是根据质量保证计划对项目过程中的工作产品进行质量审查的过程。质量保证管理者依据相关的产品标准从使用者的角度编写产品审计要素；然后根据产品审计要素对提交的产品进行审计，同时记录不符合项，将不符合项与项目相关人员进行确认；质量保证管理者根据确认结果编写产品审计报告；同时向项目管理者及相关人员提交产品审计报告。例如可以对需求文档、设计文档、源代码、测试报告等产品进行产品审计。

项目执行过程审计(有时也称为质量审查)，是对项目质量管理活动的结构性复查，是对项目的执行过程进行检查，确保所有活动遵循规程进行。过程审计的目的是确定所得到的经验教训，从而提高组织对这个项目或其他项目的执行水平。例如质量保证人员可以审计软件开发中的需求过程、设计过程、编码过程、测试过程等，确认软件人员是否按照企业的过程体系执行这些过程，如果有问题，需要进行记录，得出过程审计报告。过程审计可以是有进度计划的或随机的，可以由训练有素的内部审计师进行，或者由第三方如质量体系注册代理人进行。一方面是作为项目进行中的观察员，监督审查质量体系的执行，另外一方面是项目质量状态的报告员，报告项目的质量现状和质量过程的状态。

质量保证是保证软件透明开发的主要环节，是项目的监视机构和上报机构。在项目开发的过程中几乎所有的部门都与质量保证小组有关。质量保证小组对项目经理提供项目进度与项目真正开发时的差异报告，提出差异原因和改进方法。独立的质量保证组是衡量软件开发活动优劣与否的尺度之一。这一独立性，使其享有一项关键权利——“越级上报”。当质量保证小组发现产品质量出现危机时，它有权向项目组的上级机构直接报告这一危机。这无疑对项目组起到相当的“威慑”作用，也可以看成是促使项目组重视软件开发质量的一种激励。这一形式使许多问题在组内得以解决，提高了软件开发的质量和效率。但是，作为质量保证人员应该清晰的认识工作的性质，采取妥当的方法。一定注意相应的方法，否则会出现不应有的麻烦的。在开始前，一定与项目经理及相关人员交流，耐心协调和指导项目组人员，确认符合相应的规范，发现问题之后，要及时和项目经理沟通，争取问题的合理的解决，不要轻易上报，会产生不和谐的因素。同时要尽量减少开发人员的附加工作量，为他们提供更多的标准参考或者相应的工具，以便方便执行，质量保证人员应该掌握广泛的知识和方法，才会取得相应的信任和威信，才会起到应有的作用，否则会进入不到项目管理的实质，只是做文章了。

具体如何执行各个过程，应该参照企业相应的质量体系的定义，以及项目计划针对项目而特制的过程定义等。例如下面是针对产品审计过程定义的一个实例：

参与角色

R1：质量经理

R2：质量保证人员

R3：待审产品的负责人

进入条件

E1：待审计产品提交

输入

I1：待审计产品的标准

I2：待审计产品

活动

A1：质量保证人员依据产品标准从使用者的角度编写产品审计要素；

A2：质量保证人员根据产品审计要素对产品进行审计，并记录不符合项，将不符合项与项目相关人员进行确认；

A3：质量保证人员根据确认结果编写产品审计报告；

A4：质量保证人员向项目管理者提交产品审计报告；

A5：质量保证人员将产品审计报告提交入库。

输出

O1：产品审计报告

完成标志

F1：产品审计报告入库

质量保证活动的一个重要输出是质量报告（SQA 报告），SQA 报告是对软件产品或软件过程评估的结果，并提出改进建议。表 1 是一个产品审计的质量报告实例，当然，实际项目中的格式可以根据企业的具体情况和具体项目而定。

表 1 产品审计报告

项目名称	XXX 检测系统	项目标识	QTD-HT0302-102			
审计人	郭天奇	审计对象	《功能测试报告》			
审计时间	2003-12-16	审计次数	1			
审计主题	从质量保证管理的角度审计测试报告					
审计项与结论						
审计要素	审计结果					
测试报告与产品标准的符合程度	与产品标准存在如下不符合项： 1. 封页的标识 2. 版本号 3. 目录 4. 第一章（不存在） 5. 第二章和第三章（内容与标准有一定出入）					
测试执行情况	本文的第一章“测试方法”应在测试设计中阐述，本文的第二章基本描述了测试执行情况，但题目应为“测试执行情况”					
测试情况总结	测试总结不存在					
结论（包括上次审计问题的解决方案）						
由于测试报告存在上述不符合项，建议修改测试报告，并进行再次审计。						
审核意见						

不符合项基本属实，审计有效！

审核人：韩万江

审核日期：2003-12-16

填写人：郭天奇

填写日期：2003 年 12 月 16 日

质量保证人员要定期进行基线配置产品审计，对配置管理活动进行审计。

## 12.4.2 质量控制

质量控制是对阶段性的成果进行测试、验证，可以为质量保证提供参考依据。质量控制的结果包括产品结果和管理结果（例如成本、进度的性能等）。质量控制包括：监控特定的项目成果，以判定它们是否符合有关的质量标准，并找出方法消除造成项目成果不令人满意的原因。它应当贯穿于项目

执行的全过程。质量控制通常由开发部门或类似质量控制部门名称的组织单位执行，当然并不都是如此。

质量控制的三个要点是：

- 检查控制对象是项目工作结果。
- 进行跟踪检查的依据是相关质量标准。
- 对于不满意的质量问题，需要进一步分析其产生的原因，并确定采取何种措施来消除这些问题。

质量控制有很多方法，例如技术评审、走查、测试、返工、控制图、趋势分析、抽样统计、缺陷追踪等。

#### (1) 技术评审

技术评审（Technical Review, TR）的目的是尽早发现工作成果中的缺陷，并帮助开发人员及时消除缺陷，从而有效地提高产品的质量。

技术评审的目标是：

- 软件产品是否符合其技术规范。
- 软件产品是否遵循项目可用的规定、标准、指导方针、计划和过程。
- 软件产品的变更是否被恰当的实现，以及变更的影响等。

技术评审的主体一般是产品开发中的一些设计产品，这些产品往往涉及多个小组和不同层次的技术。主要评审的对象有：软件需求规格、软件设计规格、代码、测试计划、用户手册、维护手册、系统开发规程、安装规程、产品发布说明等。

技术评审应该采取一定的流程，这在企业质量体系或者项目计划中应该有相应地规定，例如下面便是一个技术评审的建议流程：

- 1) 召开评审会议：一般应有3~5相关领域人员参加，会前每个参加者做好准备，评审会每次一般不超过2小时；
- 2) 在评审会上，由开发小组对提交的评审对象进行讲解；
- 3) 评审组可以对开发小组进行提问；提出建议和要求；也可以与开发小组展开讨论；
- 4) 会议结束时必须做出以下决策之一：
  - 接受该产品，不需做修改；
  - 由于错误严重，拒绝接受；
  - 暂时接受该产品，但需要对某一部分进行修改。开发小组还要将修改后的结果反馈至评审组。
- 5) 评审报告与记录：所提出的问题都要进行记录，在评审会结束前产生一个评审问题表，另外必须完成评审报告。

技术评审可以把一些软件缺陷消灭在代码开发之前，尤其是一些架构方面的缺陷。在项目实施中，为了节省时间应该优先对一些重要环节进行技术评审，这些环节主要有：项目计划、软件架构设计、数据库逻辑设计、系统概要设计等。如果时间和资源允许，可以考虑适当增加评审内容。表12-8是项目实施技术评审的一些评审项。

表12-8 项目实施中技术评审

评审内容	评审重点与意义	评审方式
项目计划	重点评审进度安排是否合理	整个团队相关核心人员共同进行讨论、确认
架构设计	架构决定了系统的技术选型、部署方式、系统支撑、并发用户数量等诸多方面，这些都是评审重点	邀请客户代表、领域专家进行较正式地评审
数据库设计	主要是数据库的逻辑设计，这些既影响到程序设计，也影响到未来数据库的性能表现	进行非正式评审，在数据库设计完成后，可以把结果发给相关技术人员，进行“头脑风暴”方式的评审
系统概要设计	重点是系统接口的设计接口设计的合理，可以大大节省时间，尽量避免很多返工	设计完成后，相关技术人员一起开会讨论

很多软件项目由于性能等诸多原因最后导致失败，实际上都是由于设计阶段技术评审做的不够。对等评审是一个特殊类型的技术评审，是由与工作产品开发人员具有同等背景和能力的人员对工

作产品进行的一种技术评审，目的是早期和有效的消除软件工作产品中的缺陷，并可对软件工作产品和其中可预防的缺陷有更好的理解。对等评审是提高生产率和产品质量的重要手段。

采用检查表（Checklist）的技术评审方法对软件前期的质量控制起到非常重要的作用。采用检查表对软件需求的验证是一种结构化的工具，检查是否所有的应该完成的工作点都按标准完成，检查所有应该执行的步骤是否都正确的执行了，所以它首先确认该做的工作，其次是落实是否完成。一个成熟度高的软件企业应该有很详细、很全面、可执行性很高的评审流程和各种交付物的评审检查表（Review Checklist）。

### （2）代码走查

代码走查是在代码编写阶段，开发人员检查自己代码的过程，代码走查是非常有效的方法，它可以检查到其他测试方法无法监测的错误，好多的逻辑错误是无法通过测试手段发现的，很多的项目证明这是一个很好的质量控制方法。例如，我们做过一个项目，由于是嵌入式系统，无法很方便进行实地验证，所以我们只好自己走查自己的代码，然后互相评审代码，经过几个回合的努力，我们发现了很多的错误，每次我们都很兴奋，最后，我们的产品实地运行的时候，质量非常的高，几乎没有什么错误发生。

代码走查可以看成是开发人员的个人质量行为，代码评审是更高一层的质量控制，也是一种技术评审，代码评审是由一组人通过阅读、讨论和争议对程序进行静态分析的过程。评审小组由几名开发人员组成。评审小组在充分阅读待审程序文本、控制流程图及有关要求、规范等文件基础上，召开代码评审会，程序员逐句讲解程序的逻辑，并展开热烈的讨论甚至争议，以揭示错误的关键所在。实践表明，程序员在讲解过程中能发现许多自己原来没有发现的错误，而讨论和争议则进一步促使了问题的暴露。例如，对某个局部性小问题修改方法的讨论，可能发现与之有牵连的甚至能涉及到模块的功能说明、模块间接口和系统总结构的大问题，导致对需求定义的重定义、重设计验证，大大改善了软件的质量。

### （3）测试

项目实施相关的全部质量管理工作中，软件测试的工作量最大。由于很多项目在实施中非常不规范，因此软件测试一定要把好关。软件测试应该重点做好测试用例设计。

测试用例设计是开发过程必不可少的。项目实施中设计测试用例应该根据进度安排，优先设计核心应用模块或核心业务相关的测试用例。

单元测试可以检验单个模块是否按其设计规格说明运行，它测试的是程序逻辑。一旦模块完成时，就可以进行单元测试。集成测试是测试系统各个部分的接口，以及在实际环境中运行的正确性，保证系统功能之间接口与需求的一致性，且满足异常条件下所要求的性能级别，单元测试之后可以进行集成测试。系统测试是检验系统作为一个整体是否按其需求规格说明的正确运行，验证系统整体的运行情况，在所有模块都测试完毕，或者集成测试完成之后，可以进行系统测试。功能测试是软件首先应该从功能上满足用户需求，因此功能测试是质量管理工作中的重中之重。功能测试在产品试运行前一定要开展好，否则将会发生“让用户来执行测试”的情况，后果非常严重。

性能测试是经常容易被忽略的测试。在实施项目过程中，应该充分考虑软件的性能，运行较慢的软件仍然不会为用户所接受。性能测试可以根据用户对软件的性能需求来开展，通常系统软件和银行、电信等特殊行业应用软件对性能要求较高，应该尽早进行，这样更易于早解决问题。

另外，压力测试是测试系统在特殊条件下的限制和性能，测试系统在大数据量、低资源条件下的健壮性、系统恢复能力等等，可以在集成测试或者系统测试结束之后进行。接收测试是在客户的参与下检验系统是否满足客户的所有需求，尤其在功能和使用方便性上。

此外，对于一些项目，如果实在没有测试人员，可以考虑让开发人员互相进行测试，这样也可以发现很多缺陷。

这里强调一下测试的目的在于证明软件的错误，不是证明软件的正确。一个好的测试用例在于能发现至今未发现的错误；一个成功的测试是发现了至今未发现的错误的测试。传统的开发模式中，对

软件质量的控制放在后期的测试，指希望通过测试提高产品的质量，这种方式不仅不能从根本上提高软件的质量，而且还增加了软件的成本。其实，质量控制在项目的早期就应该开始，而且是越早开始，越能保证软件的质量和降低软件的成本。在需求、设计以及代码编写阶段可以通过各种早期的评审来保证质量。

类似质量保证过程一样，质量控制的各个过程具体如何执行，也应该参照相应的质量体系的定义，以及项目计划中针对项目而特制的过程定义等，而且会采用一定的工具和具体的方法来实施，例如测试过程会采用很多的工具，例如：SiteManager，LoadRunner，ClearTestCase 等，还应该形成测试报告，报告的格式可以参照企业的具体情况而定，例如表 12-9 是一个测试案例报告。

表 12-9 测试案例编码：WebSite-Base-link-01

版本：V1.0

测试项目名称：网站	测试人员：	测试时间：
测试项目标题：对网站页面之间超连接的测试	郭天奇	2003/11/29

**测试内容：**

- 验证网站页面中没有失败的超连接
- 验证网站页面中图片能正确装入
- 验证网站页面中的超连接的连接页面与页面中指示（或图示）相符
- 验证 Pageflow（参见网站的 pageflow.doc）

**测试环境与系统配置：**

详见《网站测试设计》，本测试需依赖 SiteManager 测试工具

测试输入数据	无
--------	---

**测试次数：**应至少在 3 次不同的负载下测试，且每个测试过程至少做 3 次

**预期结果：**

SiteManager 扫描的图中没有错误的连接页面

LinkDoctor 生成的报告中，页面连接正确

（如包含错误的连接，则为单纯的页面点击不能激活的连接）

**测试过程：**

（由于网站页面太多，可以采用分区域测试超连接）

subTest 1：

1. 在 Client 端，运行 SiteManager，扫描“X-1 功能页面”
2. 查 SiteManager 的运行结果中是否有错误的连接页面
3. 用工具 linkDoctor 生成诊断报告

subTest 2：

1. 在 Client 端，运行 SiteManager，扫描“X-2 功能页面”
2. 查 SiteManager 的运行结果中是否有错误的连接页面
3. 用工具 linkDoctor 生成诊断报告

.....

subTest n：

1. 在 Client 端，运行 SiteManager，扫描“X-n”
2. 查 SiteManager 的运行结果中是否有错误的连接页面
3. 用工具 linkDoctor 生成诊断报告

subTest n + 1：

1. 在 Client 端，运行 SiteManager，扫描所有页面
2. 查 SiteManager 的运行结果中是否有错误的连接页面
3. 用工具 linkDoctor 生成诊断报告

**测试结果：**3

**测试结论：**正常

**实现限制：**

备注：运行的硬件环境不是最佳环境

**(4) 返工**

返工是将有缺陷的、不符合要求的产品变为符合要求和设计规格的产品的行为。返工也是质量控

制的一个重要的方法，将有缺陷的和不合格项改造为与需求和规格一致的。返工，尤其是预料之外的返工，在大多数应用领域中是导致项目延误的常见原因。项目小组应当尽一切努力减少返工。可以使用检查列表或者过程调整作为质量控制的手段。

#### (5) 控制图法

控制图法是一种图形的控制方法，它显示软件产品的质量随着时间变化的情况，在控制图法中标识出质量控制的偏差标准。图 12-8 是一个软件项目的缺陷控制图示，从图中可以看到缺陷还是在可控制的范围之内的。如果超出缺陷的控制范围，应该采取措施，例如对相应的产品进行返工，或者修改开发过程，必要的时候可能提出计划的变更。

如果在控制图中有连续的 7 个或者更多的点发生在平均线的同一个方向，如图 12-9 所示，尽管它们可能都处于受控范围内，但是已经说明存在了质量问题，这个时候也需要采取措施了，称为七点规则。

#### (6) 趋势分析

趋势分析指运用数字技巧，依据过去的结果预测将来的产品。趋势分析常用来监测质量绩效，例如有多少错误和缺陷已被指出、有多少仍未纠正，以及每个阶段有多少活动的完成有明显的变动。进行趋势分析可以对一些偏向于不合格的趋势及早进行控制。

Pareto 图分析就是一种趋势分析方法，它是源自于 Pareto 规则，Pareto 规则是一个很常用的项目管理法则：80% 的问题是由于 20% 的原因引起的，(80% 的财富掌握在 20% 人的手里)。如图 12-10 是客户投诉数据图，柱状图展示了各种类型的投诉，其中登录问题是最大的问题，然后是系统问题等，第一个问题占了 50% 以上，第一个、第二个问题累计占总问题的 80% 以上，所以企业要想减少投诉问题，首先需要解决前两个问题。

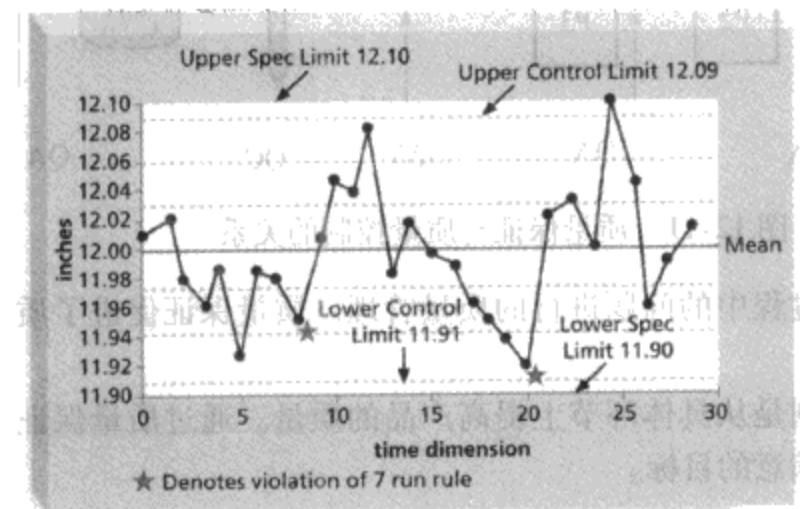


图 12-9 质量控制图与七点规则

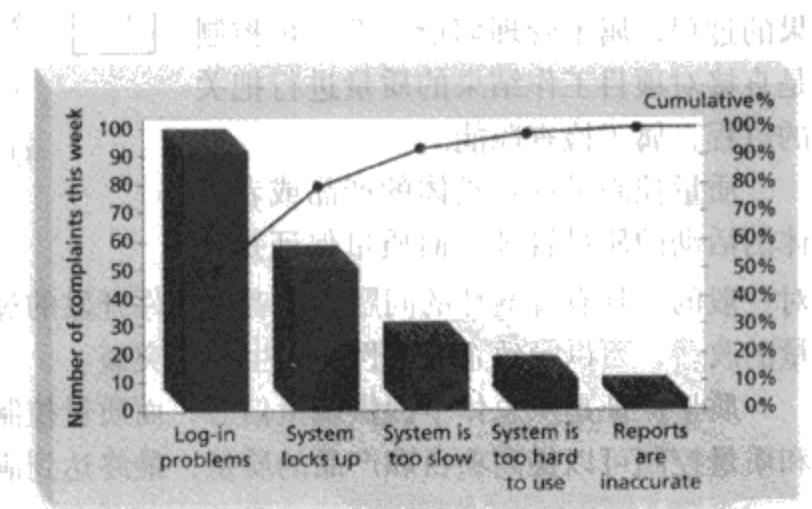


图 12-10 Pareto 图分析

#### (7) 抽样统计

抽样统计是根据一定的概率分布抽取部分产品进行检查。它是以小批量的抽样为基准进行检验，以确定大量或批量产品质量的最常用的方法。例如需要检查的代码有几万行，为了在有效的时间内检查代码的质量，可以选择其中的几段程序代码进行检查，从而找出普遍的问题所在。抽样比 100% 检查能够降低成本，但是也可能导致更高的失败成本。

#### (8) 缺陷跟踪

在软件项目开发和维护阶段，总会出现各种各样的缺陷。缺陷跟踪正是为了跟踪软件产品的所有问题。并且记录缺陷的原因，缺陷引入阶段，对系统的影响，状态，以及解决方案。而且，更加复杂的跟踪系统会跟踪到具体配置项的详细情况，以及开发人员所作的修改。缺陷追踪是从缺陷发现开始，

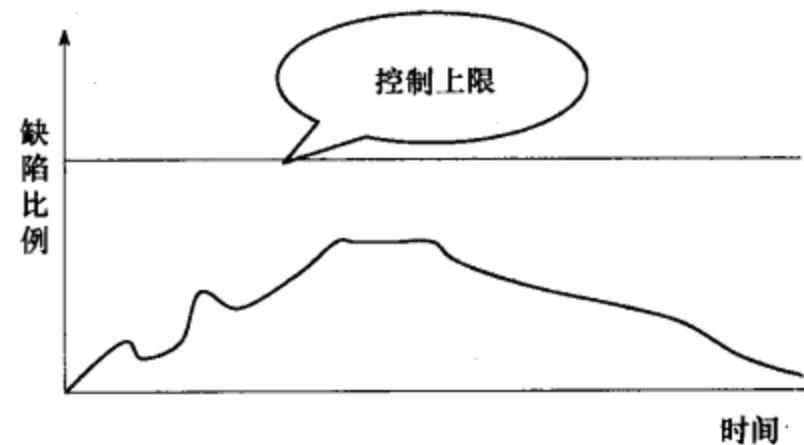


图 12-8 缺陷控制图

一直到缺陷改正为止的全过程的跟踪。

例如，在软件系统开发过程中，一个问题出现了，经过问题识别、修正并测试，这个缺陷可能被解决。但不幸的是，这种修改也可能会引入另外的缺陷。通过缺陷跟踪，我们可以在以后了解当时是谁因为什么更改了配置项。从而，理解配置项的变更历史。为软件项目的维护提供了方便。

缺陷跟踪不仅可以为软件项目质量监控提供详实的数据，也可以为质量改进提供参考。例如，通过统计某项目的缺陷数目和引入的阶段，可以改善相应阶段的质量和过程。从而达到有的放矢，为最终软件质量的提高做出贡献。

可以采用工具跟踪测试的结果，如表 12-10 就是一个缺陷跟踪工具中的表格形式。目前市场上存在很多缺陷跟踪的商用工具软件。

序号	时间	事件描述	错误类型	状态	处理结果	测试人	开发人
1							
2							
3							

对于质量控制中发现的不合格项，应及时分析原因，并进行适宜的处置，保证不合格项得到识别和有效的控制。对于已发现的不合格项或潜在不合格项，应制定相应的纠正措施或预防措施，以消除不合格或潜在不合格项的原因，防止不合格项的发生。纠正措施或预防措施制定后，应对质量计划进行相应的调整，保证项目的顺利实施。

一般说通过质量控制可以做出接受或者拒绝的决定。经检验后的工作结果或被接受，或被拒绝。被拒绝的工作成果可能需要返工。质量控制完成之后的报告应作为项目报告的组成部分。另外，通过质量控制的结果可以作为对不合理过程进行调整的依据。过程调整可以作为质量检测结果而随时进行的纠错和预防行为。

### 12.4.3 质量保证与质量控制的关系

质量保证的焦点在过程；而质量控制的焦点在产品推出前的质量把关。如图 12-11，质量保证是一种通过采取组织、程序、方法和资源等各种手段的保证来得到高质量的软件结果的过程，属于管理职能；而质量控制是直接对项目工作结果的质量进行把关的过程，属于检查职能。

质量控制是针对具体的产品或者具体的活动的质量管理；而质量保证是对一般的、具有普遍性的问题，或者说软件开发的过程中的问题进行的质量管理。质量保证促进了质量的改善，可以导致企业的性能产生一个突破。

质量保证是从总体上提供质量信心，而质量控制是从具体环节上提高产品的质量。通过质量保证和质量控制可以提高项目和产品的质量，最终达到满意的目标。

### 12.4.4 项目质量度量

软件质量度量主要有两类：预测型和验收型。预测型度量是利用定量或者定性的方法，对软件质量的评价价值进行估计，以便得到软件质量的比较精确的估算值，它是运用在软件开发过程中的。而验收型度量是在软件开发各个阶段的检查点，对软件的要求质量进行确认性检查的具体评价值，它相当于对预测型度量的一种确认，对开发过程中的预测进行评价。

质量度量方法主要有两种：第一种是尺度度量，它是一种定量度量，适用一些可以直接度量的特性，例如缺陷率等。表 12-11 便是一个项目的缺陷统计表。第二种度量是二元度量，是一种定性度量，它适用能间接度量的质量特性，例如使用性，灵活性等。一般可以制定检查表来实现质量度量，例如表 12-12 是对设计文档是否完备的检查表，对于用尺度度量可以填“值”一栏，用二元度量时，可以填“是/否”一栏。

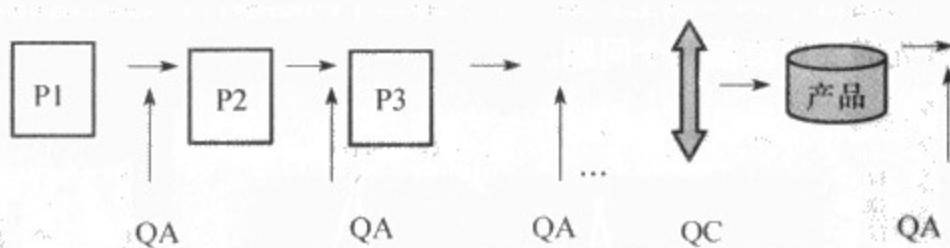


图 12-11 质量保证与质量控制的关系

表 12-11 缺陷统计表

日期	编号	类型	引入	排除	处理时间	处理错误	问题描述	归属模块
13 11月21日	6	20	20:10	20:30	20分钟	X	判断文件结束语句错误	产生程序代码源文件新的版本文件模块
14 11月21日	7	20	20:40	20:45	5分钟	X	少“)”号	产生程序代码源文件新的版本文件模块
15 11月21日	8	80	20:50	21:00	10分钟	X	读取系统时间函数错误	产生程序代码源文件新的版本文件模块
16 11月21日	9	20	21:05	21:10	5分钟	X	计数器未被赋初值	产生程序代码源文件新的版本文件模块
17 11月21日	10	80	21:20	21:30	10分钟	X	字符串转浮点数函数错误	产生程序代码源文件新的版本文件模块
18 11月22日	11	20	19:10	19:20	5分钟	X	判断文件结束语句错误	生成VER1.0版本模块
19 11月22日	12	20	19:30	19:35	5分钟	X	少“;”号	生成VER1.0版本模块
20 11月22日	13	20	19:30	19:35	5分钟	X	少“;”号	生成VER1.0版本模块
21 11月22日	14	20	19:30	19:35	5分钟	X	少“;”号	生成VER1.0版本模块
22 11月23日	15	80	19:40	20:00	20分钟	X	字符串转浮点数函数错误	生成VER1.0版本模块
23 11月22日	16	90	20:00	2:30	30分钟	X	对打开文件未做异常处理	产生程序代码源文件新的版本文件模块
24 11月23日	17	90	20:00	2:30	30分钟	X	对打开文件未做异常处理	产生程序代码源文件新的版本文件模块
25 11月22日	18	90	20:00	2:30	30分钟	X	对打开文件未做异常处理	产生程序代码源文件新的版本文件模块
26 11月23日	19	90	20:00	2:30	30分钟	X	对打开文件未做异常处理	产生程序代码源文件新的版本文件模块
27 11月22日	20	90	20:00	2:30	30分钟	X	对打开文件未做异常处理	生成VER1.0版本模块
28 11月23日	21	90	20:00	2:30	30分钟	X	对打开文件未做异常处理	生成VER1.0版本模块
29 11月24日	22	50	19:30	22:00	100分钟	X	文件名不能被函数调用	产生程序代码源文件新的版本文件模块
30 11月24日	23	50	19:30	22:00	100分钟	X	文件名不能被函数调用	产生程序代码源文件新的版本文件模块
31 11月24日	24	50	19:30	22:00	100分钟	X	文件名不能被函数调用	产生程序代码源文件新的版本文件模块
32 11月24日	25	50	19:30	22:00	100分钟	X	文件名不能被函数调用	生成VER1.0版本模块

表 12-12 评价设计文档是否完备的检查表

评价准则	度量	需求		设计		编码	
		是/否	值	是/否	值	是/否	值
设计文档的完备性	无二义性引用						
	所有数据引用都可以从一个外部源定义、计算和取得						
	所有定义的功能都被使用						
	所有使用的功能都被定义						
	对每一个判断点，所有的条件和处理都已经定义						
	所有被定义、被引用的调用序列的参数一致						

目前还不能精确地做到定量评价软件的质量，一般可以采取由若干位软件专家进行打分来评价。在评分的时候可以对每一阶段的质量指标列出检查表，同时列出质量指标应该达到的标准。有的检查表是针对子系统或者模块的，然后根据评分的结果，对照评估指标，检查所有的指标特性是否达到了要求的质量标准。如果某个质量特性不符合规定的标准，就应当分析这个质量特性，找出原因。

## 12.5 团队管理

一个系统内的元素是相互影响彼此联系的，仅有优秀的项目经理是不够的，巧妇难为无米之炊，要达到项目目标，离不开项目资源，项目经理的职责就是利用既有的项目资源达到项目的目标，满足客户的需求。人力资源是一种特殊的资源，人有主动性和情感。一个人有好恶、有情绪、有自尊。人是一种社会动物，受社会、经济、法律、家庭、生活等各方面活动的影响。“人性化管理”就是从人的精神需要出发，给以认同、尊重，以激发人最大的潜能，减少内耗，达到最大效能，参与并服务于公司行为和目标。所以，项目团队建设是必须。

我们常常会遇到项目组某个成员没有按时完成工作，导致整个进度延迟，我们常常会不由自主的批评、指责，而没有主动去寻找导致延迟的原因。风险管理中指出风险的发生必然有其原因和结果（影响）。或者他生了病；或者家里出了事；或者遇到技术难题，不能克服；或者其他。也有些人表面看来个性孤傲、难以接近，也许事实上是太专注于技术学习而没有时间浪费。也有些人对分配的工作推脱不愿意干，事实上他是缺乏完成工作所需的技术能力，让他感觉到了太大的压力，不由自主的抵触。或者心不在焉、错误不断，也许正为某件事烦恼郁闷……人的个性、特点、技能、情绪、爱好等

都会影响到软件项目中其他元素。

关于人力资源的管理理论，一些心理学家已经做了大量的研究，并提出很多有价值的理论，他们认为影响人们工作热情的主要因素包括激励、影响力、权力、效力等等。

项目组织结构确定之后，项目经理应该创建一个既实际又具有凝聚力的团队。优秀的项目领导者、高效工作的团队、有利的组织结构，是一切项目成功的理想环境。

项目组的组建工作主要包括人力资源的到位和项目组内部的组织结构、角色分配与任务分工。人力资源规划主要有：人数要求、技术能力要求、业务能力要求以及各类人员的比例。需要强调的是技术能力与业务能力的要求必须明确，以及各类人员是否需要通过培训来达到技术能力或业务能力的要求。

进行团队管理首先要明确项目干系人，项目干系人是指那些积极参与该项目工作的个体和组织，或者是那些由于项目的实施或项目的成功其利益会受到正面或反面影响的个体和组织。项目管理工作组必须识别哪些个体和组织是项目的干系人，确定他们的需求和期望，然后设法满足和影响这些需求、期望以确保项目能够成功。

项目的主要干系人还有许多不同称谓，例如：项目内部的和项目外部的；项目所有人和投资者、供应商和承包商、工作组成员及其家属、政府机构、媒介、个体公民、临时的或固定的组织，乃至于整个社会，通过对项目干系人命名和分组，我们可以确认哪些个人和组织将自己视为项目干系人。

### 12.5.1 选择人员

人是项目中最重要的资源，一个项目成功与否常常取决于工作人员的才干。软件项目是由不同角色的人共同协作完成的，每种角色都必须有明确的职责定义，因此选拔和培养适合角色职责的人才是首要的因素。选择合适的人员可以通过合适的渠道获取，而且要根据项目的需要进行，高中低不同层次的人员都需要进行合理的安排，明确项目需要的人员技能并验证需要的技能。有效的软件项目团队由担当各种角色的人员所组成。每位成员扮演一个或多个角色，可能一个人专门负责项目管理，而另一些人则积极地参与系统的设计与实现。常见的一些项目角色包括：项目经理，系统分析员，系统设计员，数据库管理员，支持工程师，程序员，质量保证工程师，业务专家（用户），测试人员等等。

团队成员的选择是组建团队的第一步，也是决定这个团队是否能有效工作的关键因素，整个团队的未来业绩将直接取决于选拔到的成员的努力。在选拔过程前，可以先通过心理测评、专业考察、查阅档案等方式获取有关人员的可靠数据，包括专业能力、性格特征、个人经历、人际关系等方面，建立备选人员人才库。

成员选择的基本原则除了要求具有基本的专业素质外，还要求具有较宽的专业知识面，对产品具有整体意识和系统集成的思想，并具有较强的合作精神。而团队领导则要求具有多专业的协调能力及处理团队与其他部门关系的能力，并能够营造好的团队文化。作为一个整体，团队的专业技能组合要达到必要的高度和广度，同时要求团队成员必须具有很好的人际关系能力，注意角色配置，以利于相互交流、彼此理解与通力合作。在挑选团队成员的过程中，既要考虑他们的性格、能力，更要遵循自愿的原则。团队成员还应该包括团队顾问或专题顾问，他们来自各职能部门，不直接参与产品的开发，但提供技术和知识上的支持。

项目人员的选择还要根据项目的需要，参考项目计划进行人员编制，必要时招聘相应岗位的人员，对他们进行相应的培训，然后将他们放到合理的岗位，对他们各自的岗位上的工作进行业绩考评，并将考评的结果与他们的报酬和升迁联系在一起。在选择人员的问题上，要结合实际情况来决定是否选入一个开发组员。并不是一群高水平的程序员在一起就一定可以组成一个成功的小组。作为考察标准，技术水平、与本项目相关的技能和开发经验、以及团队工作能力都是很重要的因素。还应该考虑分工的需要，合理配置各个专项的人员比例。

在选择人员的时候，理想的方法是由开发小组的其他成员一起来面试，如果谁看不上眼，他都不能加入，否则以后会有很多麻烦。这样做的另一个好处是借此机会互相认识一下，经理一定要把新员工介绍给大家，并且小组每个员工都应该过来握手介绍自己，这是起码的招聘礼节。

当工作任务确定后，就要安排人员来完成。合理的人员分配的基础是项目经理对每组员的能力性格有清楚的了解。人员的能力如何，高低所在，哪些人适合做什么事情等。性格如何，哪些人员搭配在一起是能够很好配合。当然项目经理需要一定的时期才能认识到人员的能力和性格。要建议一支能战的队伍，所以需要经过一定的磨合期。而为了缩短磨合期，则需要多交流，多考验。

在组建团队时，首先根据任务的需要，确立团队成员的工作对象和工作方式，项目团队组成人员的多少和比例要根据具体项目来决定。组建项目团队时首先需要定岗，就是确定项目需要完成什么目标，完成这些目标需要哪些职能岗位，然后选择合适人员组成。项目组内各类人员的比例应当协调，那种认为编码人员（软件工程师）占比例越大越好的观念是极端错误的，因为我们的目的是完成软件，而不是完成任意多的程序编码。项目也应该配备相应的需求分析、系统设计、文档编制、质量保证等人员。项目组内的组织结构宜采用小组制，具体人数根据任务划分的大小而定，根据项目任务分解结构，对项目的各项活动进行归类，作为设置小组的依据，从而使得小组的任务明确。项目经理对于小组长要给予适当的授权，授权的大小依项目的阶段和小组长的成熟度而定，一般而言，项目的前面几个阶段由于涉及到项目的技术方案等重大问题，故对小组长技术方面的授权要适度，项目的后期变化通常是局部的，因此在技术方面可以给小组长足够的权限；而当小组长的成熟度高、技术能力强时，可对其授予较大的权限，有时甚至是全权。项目组组建后要及时通过正规的沟通方式（如会议讨论）将项目目标贯彻到全体组员中去，以达成项目目标共识，形成群体目标。项目计划也要充分吸收骨干组员的意见，一方面使得计划更符合实际，另一方面通过参与的形式达成共识，增强他们的归属感和使命感。

多年来，许多的心理专家和管理专家进行一系列的实验，帮助我们理解人们的行为，这些试验可以使项目经理对矛盾根源，人们的动机，生产效率等问题茅塞顿开。有的是测试在冲突时人们的表现：好胜、合作、容纳、回避等。有的是测试人们关注任务和关注人的程度等。其中最受欢迎的是 Myers-Briggs 心理测试方法，或称为 Myers-Briggs 类型指标（Myers-Briggs Type Indicator, MBTI），它是用一系列的心理测试来决定一个人的心理类型。MBTI 将人格分为 4 个维度，每个维度有两种偏好，分别是：外向-内向（extravert-intravert），注重事实-注重感觉（sensing-intuitive），理性-感性（thinking-feeling），决断-思考（judging-perceiving）。

这样，4 个维度 8 种偏好，两两组合，共组成 16 种人格类型，以各个维度的字母表示类型，见表 12-13。

这 16 种人格类型各自具有不同的性格表现，这些表现也适应于不同的工作或者团队。一般说，进行常规工程的人员可以选择 ESTJ（外向，注重事实，理性，决断），进行设计的人员可以选择 ENTJ，INTJ（外向、内向，注重感觉，理性，决断），市场人员可以选择 ESFJ（外向，注重事实，感性，决断）等等。每个项目经理都希望获得理想的项目人员，理想的人员是完全献身于项目，有素养和理解力，充分理解自己的任务目标，认真执行指令，发生未料到的事情时，勇于处理，而且也懂得事情的分寸，不该问的事情决不去问，技术上很强，在自己的专业领域是一位行家里手，当项目经理给他一项任务时，他能够保证有效地、高质量地完成任务。

项目人力资源管理的一项重要任务是根据每个人的专长、特点、爱好来安排任务，充分做到人尽其才。在对项目成员配备工作时，应该依据以下原则：1) 人员的配备必须要为项目目标服务。2) “以岗定员”，保证人员配备的效率，充分利用人力资源，不能以人定岗。3) 项目处于不同的实施阶段，所需人力资源的种类、数量、质量是不同的。要安排一定比例的临时工作人员，根据项目的需要加入或者退出，节约人力资源成本。

表 12-13 MBTI 的 16 种人格类型

ESFP	ISFP	ENFJ	ENFP
ESTP	ISTP	INFJ	INFP
ESFJ	ISFJ	ENTP	INTP
ESTJ	ISTJ	ENTJ	INTJ

在项目的实施过程当中，要根据项目随时可能发生的变化，评价项目组织目前的人力资源状况，并做好未来项目组织对人力资源的需求预测，以及未来人力资源的供给预测，保证项目随时都能够及时获得需要的人力资源。

在项目生存期中，项目相关人员的数量和特点经常会随着项目从一个阶段进入另一个阶段而有所改变，结果使得在一个阶段中非常有效的管理技巧到了另一个阶段会失去效果。人力资源的行政管理工作一般不是项目管理小组的直接责任。但是，为了深化管理力度，项目管理小组必须对行政管理的必要性有足够的重视。

### 12.5.2 项目成员的培训

为增加团队的凝聚力和向心力，增强成员对团队的归属感和责任感，项目团队可以通过帮助成员设计职业发展方向，来帮助成员适应多方面的工作和未来发展的需要，同时使成员为自己的良好发展前景而不愿轻意离开团队。成员加入团队后，根据成员个人的条件和背景，由成员和项目经理共同协商，结合项目特点，研讨一套切实可行的个人职业生涯发展体系，协助成员开发其各种知识和技能，尤其是专业性知识和技能，为成员提供实现个人专长的契机。通过个人职业生涯发展计划，使每位成员对自己目前所拥有的技能、兴趣及价值观进行评估，接着考虑项目的变化需求，使自己的特长及发展方向符合团队的需求。通过团队为成员设计良好的个人发展计划和职业发展阶梯，就会促进团队和成员的发展，降低成员的流动率和流动倾向。对于那些看重学习和愿意获得新技能的成员，由项目团队提供培训机会，鼓励他们，以增加他们的满足感和责任感。

团队建设是实现项目目标的重要内容，而项目成员的培养是项目团队建设的基础，项目组织必须重视对员工的培训工作。通过对项目成员的培训，可以提高项目团队的综合素质，提高项目团队的工作技能和技术水平。同时也可以通过提高项目成员的本领，提高项目成员的工作满意度，降低项目人员的流动比例和人力资源管理成本。

针对项目的一次性和制约性（主要是时间的制约和成本的制约）特点，对于项目成员的培训主要采取短期性的、片断式的、针对性强、见效快的培训。培训形式主要有两种：1) 岗前培训。主要对项目成员进行一些常识性的岗位培训和项目管理方式等的培训。2) 岗上培训是根据开发人员的工作特点，针对开发中可能出现的实际问题，而进行的特别培训。多偏重于专门技术和特殊技能的培训。

### 12.5.3 项目成员的激励

项目团队士气是项目成功的一个因素，项目成员的激励是调动成员工作热情非常重要的手段。

管理者通过采取各种措施，给予项目成员一定的物质刺激、精神激励，去激发项目成员的工作动机，调动员工的工作积极性、主动性，并鼓励他们的创造精神，从而以最高的效率完成项目，实现项目目标。当然激励一定要因人而异，可以适当参照下面做法：

- 薪酬激励：对于软件人员，如果支付的薪酬与其贡献出现较大偏差时，便会产生不满情绪，降低工作积极性，因此，必须让薪酬与绩效挂钩。
- 机会激励：在运用机会激励时，要讲究公平原则，即每位员工都有平等的机会参加学习、培训和获得具有挑战性的工作，这样才不会挫伤软件人员的积极性。
- 环境激励：企业内部良好的技术创新氛围，企业全体人员对技术创新的重视和理解，尤其是管理层对软件人员工作的关注与支持，都是对软件人员有效的激励。
- 情感激励：知识型员工大都受过良好的教育，受尊重的需求相对较高，尤其对于软件人员，他们自认为对企业的贡献较大，更加渴望被尊重。

其他激励，如弹性工作制，由于软件人员的工作自主性特点，宽松、灵活的弹性工作时间和工作环境对于保持创新思维很重要。

目前出现很多的激励理论，比如马斯洛的需求层次理论，海兹波格的激励理论，麦克格勒格的

X-理论、Y-理论、Z-理论、期望理论等等。这些理论各自有不同的侧重点。

### 1. 马斯洛的需求层次理论 (Maslow's Hierarchy of Needs)

马斯洛的需求层次理论认为人类的需要是以层次形式出现的，共有5个层次，见图12-12，生理、安全、社会归属、自尊和自我实现。其中自我实现是最高的层次，低层次的需求必须在高层次需求满足之前得到满足，满足高层次需求的途径比满足低层次需求的途径更为广泛，激励来自为没有满足的需求而努力奋斗。例如：新员工有群体归属感的需要（社会层次上的需要），为满足这方面的需要，可以为其开个热情的欢迎会，新老员工相互介绍与沟通，另外要对其生活方面的困难给予帮助。再比如：青年人常有想多学点东西的愿望，这是寻求自我发展与成长的需要（自我实现层次的需要），项目经理要运用任务分配权，在可能的范围内尽量满足这种愿望，特别是对于进取心强的骨干，应分配给他们新的具有挑战性的任务，这是一种激励的手段，同时也是培训人员的一种方法。阿尔德佛进一步发展了马斯洛的需求层次理论，认为有的需求（如自我实现）是后天通过学习而产生的；人的需求不一定严格地按从低到高的顺序发展，并提出了有名的“挫折—倒退”假设；管理者应努力把握和控制好工作结果，通过工作结果来满足人们的各种需求，从而激发人们的工作动机。其中，最后一点强调了工作结果对各种需求的满足，因此在软件项目的阶段点或最终产品达成时，要非常慎重地予以宣布，并通过多种途径肯定这一成果，如表扬、奖励等，全方位地满足组员的需求，从而激发下一阶段的工作热情。

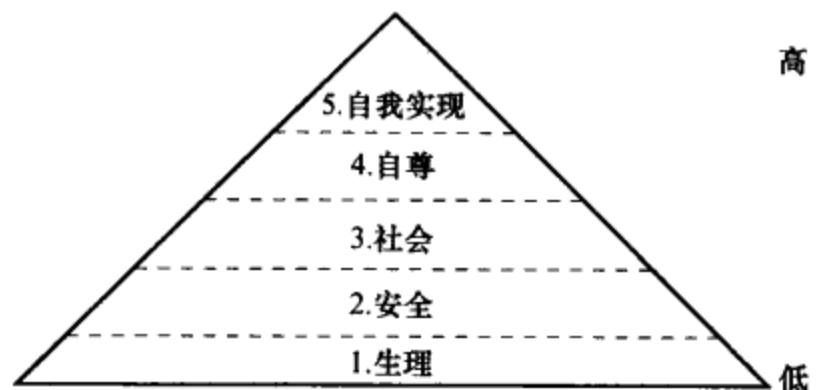


图 12-12 马斯洛的需求层次

### 2. 海兹波格的激励理论 (Herzberg's Motivational and Hygiene Factors)

海兹波格的激励理论认为企业中存在着两组因素：一组导致不满，另一组产生激励。不满因素是工作环境或组织方面的外在因素，满足了这些因素的要求就能避免员工的不满情绪，故称为“保健因素”，它主要包括公司的政策与管理、安全感、工资及其他报酬、人际关系等。而提高员工的工作情绪源于内在因素，通常称为“激励因素”，它包括了成就感、责任感、进步与成长、被赏识等。因此，要从这两个因素的角度来协调管理。从“保健因素”的角度，项目经理要密切注意员工的情绪波动，多与员工沟通，消除与缓解员工的不满情绪；对制度和政策多作解释工作，以消除误解；向上级反映员工的合理要求与建议，以便及时完善有关政策和制度；在项目组内协调好人际关系，对出现的紧张关系要及时地调解；改善工作流程可能有助于组员间更好地协助工作；对员工要公正地评价其表现和安排晋级。从“激励因素”的角度，项目经理要鼓励和帮助员工制订个人成长计划（如多长时间学会哪门技术）；为员工的进步与成长提供机会，包括分配适当的任务和培训；项目经理要对骨干进行适当地授权，尽量放权给员工，对成绩及时地肯定，提高员工的成就感和责任感。

### 3. 麦格雷戈的X-理论 (McGregor's Theory X)

麦格雷戈的X理论对人性假设的主要内容是：1) 人天生是懒惰的，不喜欢他们的工作并努力逃避工作；2) 人天生就缺乏进取心，缺乏主动性，工作不愿负责任，没有解决问题与创造的能力，更喜欢经常的被指导，宁愿被领导，避免承担责任，没有什么抱负；3) 人天生就习惯于明哲保身，反对变革，把对自身安全的要求看得高于一切；4) 人缺乏理性，容易受外界和他人的影响做出一些不适宜的举动；5) 人生来就以自我为中心，无视组织的需要，对组织需求反应淡漠，反对变革，所以对多数人必须使用强迫以至惩罚、胁迫的办法，去驱使他们工作，方可达到组织目标。

X理论强调需要用马斯洛的底层需求（生理和安全）进行激励。这个理论不适合软件项目人员的激励。

X理论就是强势管理。假设你的下属逃避责任、不愿意动脑筋，甚至很讨厌上司给他分派工作，碰到这种下属，就需要一种强势管理。这种强势管理可以对员工产生约束力，提高企业生产效率。但

是，X理论忽视了人的自身特征和精神需要，只注重人的生理需要和安全需要的满足，把金钱作为主要的激励手段，把惩罚作为有效的管理方式，麦格雷戈对人的需要、行为和动机进行了重新研究，提出一种新的假设，即Y理论。

#### 4. 麦格雷戈的Y-理论 (McGregor's Theory Y)

麦格雷戈的Y理论对人性假设的内容是：1) 人天生是喜欢挑战的，要求工作是人的本能；2) 在适当的条件下，人们能够承担责任，而且多数人愿意对工作负责，并有创造才能和主动精神；如果给予适当的激励和支持性的工作氛围，会达到很高的绩效预期，具有创造力，想象力，雄心和信心来实现组织目标；3) 个人追求与组织的需要并不矛盾。只要管理适当，人们能够把个人目标与组织目标统一起来；4) 人对于自己所参与的工作，能够实行自我管理和自我指挥；能够自我约束，自我导向与控制，渴望承担责任；5) 在现代工业条件下，一般人的潜力只利用了一部分。

Y-理论认为需要用马斯洛的高层需求（自尊和自我实现）进行激励。用Y理论指导管理，要求管理者根据每个人的爱好和特长，安排具有吸引力的工作，发挥其主动性和创造性；同时要重视人的主动特征，把责任最大限度地交给每个人，相信他们能自觉完成工作任务。外部控制、操纵、说服、奖罚，不是促使人们努力工作的唯一办法，应该通过启发与诱导，对每个工作人员予以信任，发挥每个人的主观能动作用，从而实现组织管理目标。

Y理论属于参与管理，如果你有这样的下属，他们愿意接受任务，也喜欢发挥自己的潜力，喜欢有挑战性的工作，作为一名管理者你应该给这样的下属一些机会，让他们参与管理。但是，经过实践，人们发现Y理论并非在任何条件下都比X理论优越，管理思想和管理方式应根据人员素质、工作特点、环境情况而定，不能一概而论。这便是超Y理论产生的理论基础。

#### 5. 超Y理论

超Y理论是莫尔斯和洛希在1970年发表的论文《超Y理论》中提出的，其主要观点是：1) 人们是怀着许多不同的需要加入工作组织的，各自有不同的情况：有的人自由散漫，不愿参与决策，承担责任，需要正规化的组织机构和严格的规章制度加以约束；有的人责任心强，积极向上，则需要更多的自治、责任和发挥创造性的机会去实现尊重和自我实现的需要；2) 组织形式和管理方法要与工作性质和人们的需要相适应，对有些人（如懒惰、缺乏进取心者）适用X理论管理，而对另一些人（如富有责任心、工作主动者）则适用Y理论管理；3) 组织机构和管理层次的划分、职工培训和工作分配、工作报酬和控制程度等，都要从工作性质、工作目标、员工素质等方面进行综合考虑，不能千篇一律；4) 当一个目标达到后，应激起员工的胜任感，使他们为达到新的、更高的目标而努力。但是，认真分析和研究之后，人们发现：不论是X-Y理论，还是超Y理论，都存在一个不足之处，就是其理论研究的出发点，多半是从管理当局与员工对立为基本前提。有鉴于此，便产生了Z理论。

#### 6. Z理论 (William Ouchi's theory Z)

Z理论的提出者是威廉·大内——日裔美国管理学家、管理学教授。威廉·大内于1973年开始研究日本企业管理，针对日美两国的管理经验，1981年出版《Z理论》一书。Z理论认为，经营管理者与员工的目标是一致的，二者的积极性可以融合在一起。

Z-理论是基于日本的员工激励方法，它强调忠诚、质量、集体决策和文化价值。Z理论的基本思想是：1) 企业对员工实行长期或终身雇佣制度，使员工与企业同甘苦共命运，并对职工实行定期考核和逐步提级晋升制度，使员工看到企业对自己的好处，因而积极关心企业的利益和企业的发展；2) 企业经营者不单要让员工完成生产任务，而且要注意员工培训，培养他们能适应各种工作环境需要，成为多专多能的人才；3) 管理过程既要运用统计报表、数字信息等鲜明的控制手段，而且要注意对人的经验和潜在能力进行诱导；4) 企业决策采取集体研究和个人负责的方式，由员工提出建议，集思广益，由领导者作出决策并承担责任；5) 上下级关系融洽，管理者对职工要处处关心，让职工多参与管理。

不同于“性本恶”的X理论，也不同于“性本善”的Y理论，Z理论是“以争取既追求效率又尽可能减少当局与职工的对立，尽量取得行动上的统一”。目前管理界还提出了H理论，H即Haier，海

尔创造的是具有中国特色的“H理论”是：主动变革内部的组织结构，使其适应员工的才干和能力，而最终实现人企共同发展。

### 7. 期望理论

期望理论最早是由美国心理学家佛隆在1964年出版的《工作与激发》一书中首先提出来的。其基本内容主要是佛隆的期望公式和期望模式。

佛隆认为，人总是渴求满足一定的需要并设法达到一定的目标。这个目标在尚未实现时，表现为一种期望，这时目标反过来对个人的动机又是一种激发的力量，而这个激发力量的大小，取决于目标价值（效价）和期望概率（期望值）的乘积。用公式表示就是： $M = \sum V \times E$ 。M表示激发力量，是指调动一个人的积极性，激发人内部潜力的强度。V表示目标价值（效价），这是一个心理学概念，是指达到目标对于满足他个人需要的价值。同一目标，由于各个人所处的环境不同，需求不同，其需要的目标价值也就不同。同一个目标对每一个人可能有三种效价：正、零、负。效价越高，激励力量就越大。E是期望值，是人们根据过去经验判断自己达到某种目标的可能性是大还是小，即能够达到目标的概率。目标价值大小直接反映人的需要动机强弱，期望概率反映人实现需要和动机的信心强弱。这个公式说明：假如一个人把某种目标的价值看得很大，估计能实现的概率也很高，那么这个目标发动机的力量越强烈。

怎样使激发力量达到最好值，佛隆提出了人的期望模式：个人努力→个人成绩（绩效）→组织奖励（报酬）→个人需要

在这个期望模式中的四个因素，需要兼顾三个方面的关系。

1) 努力和绩效的关系。这两者的关系取决于个体对目标的期望值。期望值又取决于目标是否合适个人的认识、态度、信仰等个性倾向，及个人的社会地位，别人对他的期望等社会因素。即由目标本身和个人的主客观条件决定。

2) 绩效与奖励关系。人们总是期望在达到预期成绩后，能够得到适当的合理奖励，如奖金、晋升、提级、表扬等。组织的目标，如果没有相应的有效的物质和精神奖励来强化，时间一长，积极性就会消失。

3) 奖励和个人需要关系。奖励什么要适合各种人的不同需要，要考虑效价。要采取多种形式的奖励，满足各种需要，最大限度的挖掘人的潜力，最有效的提高工作效率。

软件人员需要得到关心和尊重，曾经有个例子，某公司开发人员李某由于刚开始学习编程，技术水平差一点，常常受到经理的“另眼相看”，每次软件出现了问题都怀疑是他的原因，而且大家也不尊重他。经理希望通过这种手段能够迫使他自动辞职，然而，这位员工后来工作非常不负责任，把代码写得既长又重复，且在代码中留下大量的隐患，此时，经理却反而不敢过份得罪他了，否则，留下的巨量代码很难维护。

这个例子说明如果认为某人不适合目前工作，为何不另请高明？既然已经请他作了这件工作，就得尊重他。正所谓“尺有所短，寸有所长”。软件人员普遍喜欢自由而宽松的工作环境，最好不要做过多的无谓的规定等。如果确实有人经常上班迟到，工作不认真等，首先应该了解原因，此时多作思想工作是必要的，许多公司的经理们认为“思想工作”是过时的东西了，其实不然，开发人员特别需要有人关心，特别需要心理上的“减负”。

#### 12.5.4 团队建设的方法

一个团队的创建包括四个主要的阶段，组建（forming）阶段、磨合（storming）阶段、规范（norming）阶段和执行（performing）阶段。其中，

1) 组建阶段是项目团队开始组建，个体成员组成一个项目团队，互相介绍以彼此了解的阶段。这个阶段项目经理应该向项目团队成员介绍项目目标、项目计划以及成员的角色和职责，而项目团队成员之间彼此可能会言语谨慎一些。

2) 磨合阶段是项目团队的磨合期，相互之间合作可能不尽如意，个人还没有融入团队，团队内还

没有凝聚力，团队成员可能会出现彼此竞争或者士气低落，甚至有相互对立的现象。

3) 规范阶段是项目成员逐步接受项目的环境，成员之间也加深了解的阶段，在这个阶段，项目团队开始有凝聚力，项目团队成员会表现出信任和友谊，共同解决项目问题，并能够集体做出决定。

4) 执行阶段是项目团队的完善阶段，成员之间坦诚合作，相互帮助，使得项目有很好的绩效，工作效率提高，这个阶段是项目经理追求的最终阶段。

项目组建的这4个阶段不是简单的单向过程，它可能是反复曲折的过程，而且这四个过程可能会被新人加入打乱节奏，再从头开始。如图12-13所示。

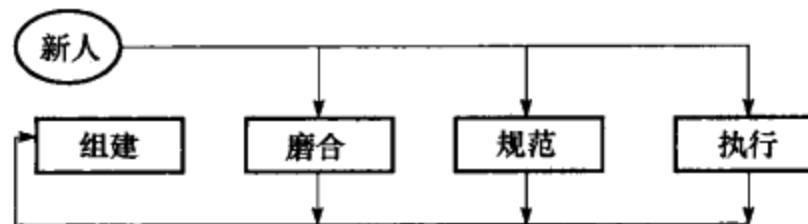


图 12-13 项目组建的 4 个阶段

### 12.5.5 团队管理

大家对球赛应该比较熟悉，一般说，有凝聚力的球队会赢球，缺少凝聚力的球队容易输球。球队是比赛的基本单元，类似的，软件项目队伍就是项目的基本工作单元，队伍的作用非常重要，是顺利实施项目的基础平台，值得花时间研究、探讨与项目成败的关系，以便更好的组建队伍，最大限度地提高工作效率。人是软件项目中最重要的因素，是其他一切的基础，人力成本是软件项目的主要成本，所以，软件项目团队管理就更为重要了。高效的软件开发团队是建立在合理的开发流程及团队成员密切合作的基础之上，团队成员共同迎接挑战、有效地计划、协调和管理各自的工作以至成功完成项目目标。

团队管理非常重要，它是项目顺利进行的基础。对于一个球队来说，要大力培养他们的团队精神，要求队员深刻认识自己球队的特点，团队精神能使球队更具有竞争力，可以打败实力相同而没有团队精神的球队。同理，对于软件项目团队也一样，在开发复杂软件的时候，通常每个人开发不同部分，运行这些软件的设备又可能来自不同的供应商，而事后将软件的不同模块集成在一起，带来的问题会更多。一个软件本身没有问题，但是合在一起可能不能工作，所有这些都需要一个高效合作的团队来共同完成。

为了及时的跟踪团队成员的工作状况，提高工作效率，项目管理者要激励成员、及时反馈信息、解决问题、整合变更。项目管理者注意观察团队成员的状况，必要的时候与团队成员谈话、聊天，及时发现问题和解决问题。定期进行项目评估，确定问题，对于冲突有一定的解决方法。

项目的目的就是实现项目干系人的需求和愿望。项目干系人管理应当从项目的启动开始，项目经理及其项目成员就要分清项目干系人包含哪些人和组织，通过沟通协调对他们施加影响，驱动他们对项目的支持，调查并明确他们的需求和愿望，减小其对项目的阻力，以确保项目获得成功。

另外，自我管理也是团队管理的一个方面，自我管理能充分发挥人的主观能动性，但是缺点是过程难于把握，遇到主动性不强的人则会出现问题。自我管理的一个重要的形式就是承诺管理。是在充分尊重被管理者的基础上进行管理，是以对方的承诺作为工作量度的管理，具有人性化的管理特点。

制度管理则依靠制度，对日常管理和过程进行细化规范的管理。制度管理能让过程在受控下进行，但是却抑制了人的主动性，尤其是对智力性的管理。软件开发就是高密度的智力工作，智力成果是无形的，所以对智力性工作很多时候是很难度量的。所以在软件开发中应该采用一种折中的方法。宏观上采用制度管理，微观上采用自我管理。

项目团队是一个抽象的概念，有时项目成员只看到其中一部分，像盲人摸象一样，只有项目经理能从整体的角度考虑项目队伍，为了建立一个具有高度主观能动性的项目队伍，必须使项目队伍尽可能实体化。团队开发过程中，要开发所有项目相关人的潜能为项目服务，必须承诺实现共同的目标，学会团队精神，分享共同工作的热情，开发高质量的产品。促进团队成员之间的了解、信任、依赖；增加团队的凝聚力；提高项目团队的士气，增加团队战斗力；培养团队以团队目标为自己的奋斗目标的理念；培养内部的团结合作和与其他组织的合作精神；注意团队个人能力的培养和个人的发展。

在团队管理过程中应该注意以下方面。

### 1. 创建有实际存在感的项目团队

项目团队要定期召开会议：会议一方面传达信息，另一方面强调队伍的整体性，坐在一个会议室，彼此相见，互相认识，感到一个实际存在的团队。比如很重要的项目启动会议要宣布项目章程、明确责任、目标、确定成员、明确进度、里程碑、以及成员的联系方式，另外一个重要会议是项目评审会，它是检查项目进展的周期性会议。创造有实际存在感的项目团队的最好办法是营造一个队伍空间，所有开发人员一起工作，一起讨论问题，在这个空间里可以看到组织机构图表，项目进展图标，项目报告，如果可以的话可以创建队伍标志，让大家随时可以看到只属于他们自己的标志。这样，大家进入到这里就宛如进入到一个作战室一样。

### 2. 建立奖励机制

虽然项目经理的权力是有限的，但项目经理可以利用有限的权力为团队成员建立一个最佳的奖励机制。例如为表现优秀的成员写推荐信；利用分配工作的控制权，保证优秀的员工有选择的工作安排；利用进度的控制权，调整工作进度照顾模范的员工，可以为超时工作的员工提供休假；可以推荐给某位员工发公司的奖金；添置设备时；可以先提供给最好的员工使用；可以邀请员工共进晚餐；可以对谁能享受优待提出意见；可以通过允许项目成员向更高层汇报工作来开阔他们的视野；激励团队的人员尽自己的最大努力工作。

### 3. 确立良好人际关系

对员工的表现要有积极的反馈，确保交流的畅通，在公共的场合肯定优秀的工作表现对项目成员要关心，做一个“挽起袖子”式的管理者，要平易近人，清楚地说明你的期望以及工作要求，坚持原则，授以项目成员决策权。关键的里程碑节点完成以后，举办里程碑聚会来庆祝。至少记住两条不能做的事情：不要当众批评项目成员，不要在产生麻烦时责备队伍。如果项目出现了麻烦，那么主要责任者是项目经理，而不是项目成员。

### 4. 设置工作授权系统

在项目进行过程中设置工作授权系统，工作授权系统就是一个用来确保合格的人在正确的时间、以合适的顺序进行工作的方法。在该系统结构下，通过正式的文件和签字授权某人开始进行某个项目活动或工作包的实施工作，同时给予项目人员应有的权力。

大部分对项目成功的定义说的是项目如何按时完成、是否在预算内以及是否满足用户的需要。但是，在如今，衡量项目成功与否的另一个重要手段是项目结束后团队的士气。在项目结束之际，项目团队的各个成员是否觉得他们从自己的经历中学到了一些知识、是否喜欢为这次项目工作，以及是否希望参与组织的下一个项目都是非常重要的。一位资深的垒球队教练曾经说过，获得球员很容易，但是让他们齐心协力打好一场球赛就会很困难。只有专家还不能保证项目成功，必须让他们能团结协作，有时项目困难之处也正在于此。

在人员的管理方面，一方面要求项目组成员相对稳定，以保证开发工作的连续性；另一方面，不搞终身制，经调查经过短期培训后仍不能够胜任本职工作的坚决调换，保证项目整体工作不受影响。通过平常和阶段性的工作考核、评审，对不合格人员进行调换。此外，为了调动成员的积极性，可以从各个方面着手，如在项目启动会上邀请公司高层领导参加，请领导鼓舞士气；尽量根据成员兴趣安排任务；按期进行绩效评定并给予适当奖励。

## 12.6 沟通管理

沟通从一定意义上讲，就是管理的本质。管理离不开沟通，沟通渗透于管理的各个方面。所谓沟通，是人与人之间的思想和信息的交换，是将信息由一个人传达给另一个人，逐渐广泛传播的过程。著名组织管理学家巴纳德认为“沟通是把一个组织中的成员联系在一起，以实现共同目标的手段”。沟通是保持项目顺利进行的润滑剂。

### 12.6.1 项目沟通的方法

沟通中一个重要内容是协商，协商是指与他人交换意见以便得出结论或达成共识，为了达成共识可能需要进行直接的协商或者通过一些辅助手段进行协商，调解和仲裁就是协商的两种辅助手段。项目在许多层次、许多观点上会有多次的协商，在一种典型项目的进行过程中，项目工作人员需要就以下全部或部分内容进行协商：

- 范围、成本和进度目标。
- 范围、成本或进度的变动。
- 合同条款。
- 任务分配。
- 资源等等。

在项目的实施过程中可以根据项目的具体情况，采取合适的沟通方法和沟通技术。

#### 1. 利用先进的技术

可以利用信息化系统来管理项目的信息，并进行有效的沟通。例如可以采用互联网的方式，将项目需要的文档标准、会议纪要、需求变更、客户要求等及时发布和获取，也可以通过配置管理系统实现信息的沟通。

#### 2. 正式沟通

正式沟通是通过项目组织明文规定的渠道进行信息传递和交流的方式。它的优点是沟通效果好，有较强的约束力。缺点是沟通速度慢。

在项目管理过程中一个比较重要的、正式的沟通方式是项目评审，项目评审是项目管理中一个重要的手段。技术评审的对象主要是规范和设计，而项目管理的管理评审，关注的是项目规划和工作报告。

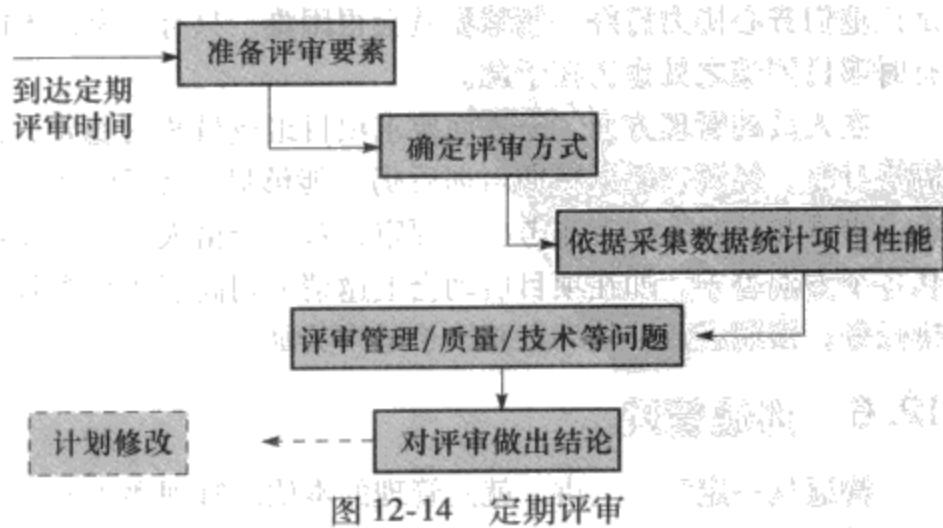
项目评审是通过一定的方式对项目进行评价和审核的过程，通过项目评审可以明确项目的执行状况，并确定采取的管理措施。评审时，需要对进度计划、成本计划、风险计划、质量计划、配置计划等的执行情况进行评价，确认计划中各项任务的完成情况，重新评估风险，更新风险表；明确是否所有的质量、配置活动都在执行，团队的沟通情况如何等。给出当前为止项目的执行结论。

评审会议能突出一些重要项目文件提供的信息和综合的状态信息，迫使人们对他们的工作负责。通过对重要的项目问题进行面对面的讨论，激励职员在自己负责的项目部分上取得进展。项目经理应为评审会议制定基本规则，以控制冲突的数量，并解决潜在的问题。不要忘了让项目干系人一道解决执行中的问题，这是很重要的。

按照评审的时间属性，可以将项目评审分为如下几种：定期评审、阶段评审、事件评审等。

#### (1) 定期评审

定期评审主要是根据项目计划和跟踪采集的数据定期对项目执行的状态进行评审，跟踪项目实际结果和执行情况，检查任务规模是否合理，项目进度是否得以保证，资源调配是否合理，责任是否落实等。根据数据分析结果和评审情况及时发现项目计划的问题，评审相关责任落实情况，对于出现的偏差采取纠正措施。定期评审过程如图 12-14 所示。



在软件构建阶段，项目经理应该每周进行定期评审，以便及时了解项目的状况，评审的主要内容有：

- 从项目计划工具，缺陷跟踪工具，时间统计工具等处收集项目信息。
- 将完成的情况与计划进行比较。
- 将实际的缺陷情况与计划进行比较。
- 将实际的项目规模与计划进行比较。
- 评审并更新风险计划。
- 评审通过匿名渠道反馈上来的信息。
- 评审建议的变更和SCCB批准的变更，同时评审这些变更对项目计划总的波及影响。

基于上述评审，项目经理可以对项目实际执行结果与计划做出比较，如果出现明显偏差，就需要采取纠正措施。

定期评审是用来交流项目信息的定期会议，该会议同时也是一个很好的激励工具。因为项目成员如果知道他们每月的工作情况要向团队或其他层次正式汇报，他们一定会确保完成工作任务。

### (2) 阶段评审

阶段评审（或称里程碑评审）主要是在项目计划中规定的阶段点（或里程碑），由项目管理者组织，根据项目计划，定期评审报告，技术评审报告和SQA评审报告对该阶段任务完成情况和产品进行评审，目的是检查当前计划执行情况，检查产品与计划的偏差，并对项目风险进行分析处理，判定是否可以对产品进行基线冻结。一个好的计划应该是渐进完善和细化的，所以阶段评审之后应该对下一阶段项目计划进行必要的修正。阶段评审一般采用会议形式。评审过程如图12-15所示。

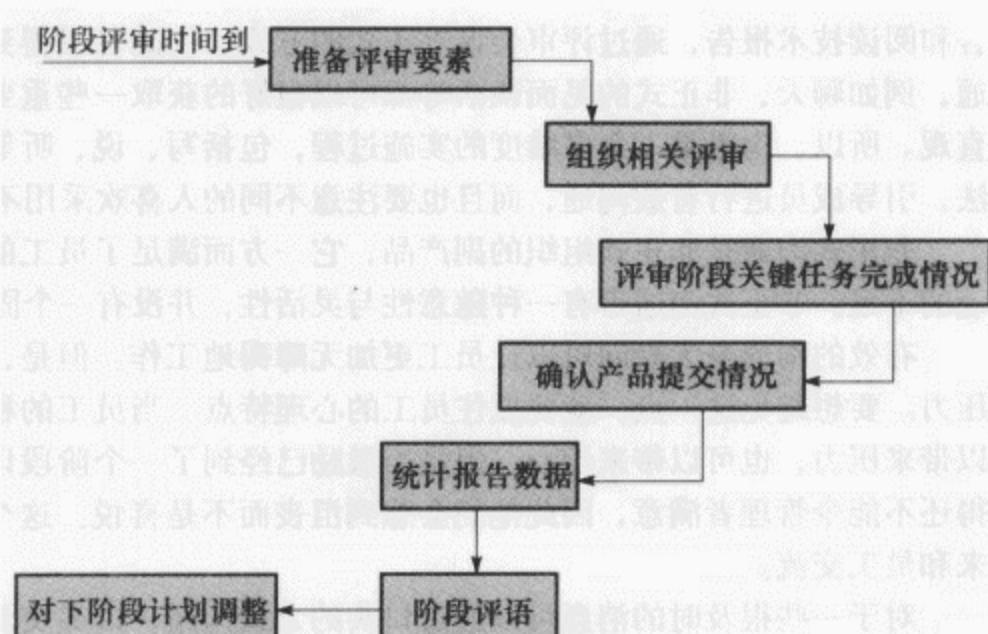


图 12-15 阶段评审

### (3) 事件评审

在项目进展过程中可能会出现一些意想不到的事件，需要项目经理及时解决。事件评审主要是根据项目进行过程中相关人员提交的事件报告（这里的事件主要是指对项目进度和投入成本产生影响的技术事件、质量保证事件、项目管理事件和项目支持事件），对该事件组织相关人员进行评审，目的是通过分析事件性质和影响范围，讨论事件处理方案，并判断是否影响项目计划，必要时采取纠正措施，从而保证整个项目的顺利进行。事件评审过程如图12-16所示。

项目评审包括评审准备、评审过程以及评审报告三个过程。

1) 评审准备。评审准备主要是评审负责人确定评审内容；向评审参与者发送评审内容及有关评审资料；评审参与者审阅评审内容及有关评审资料的过程。

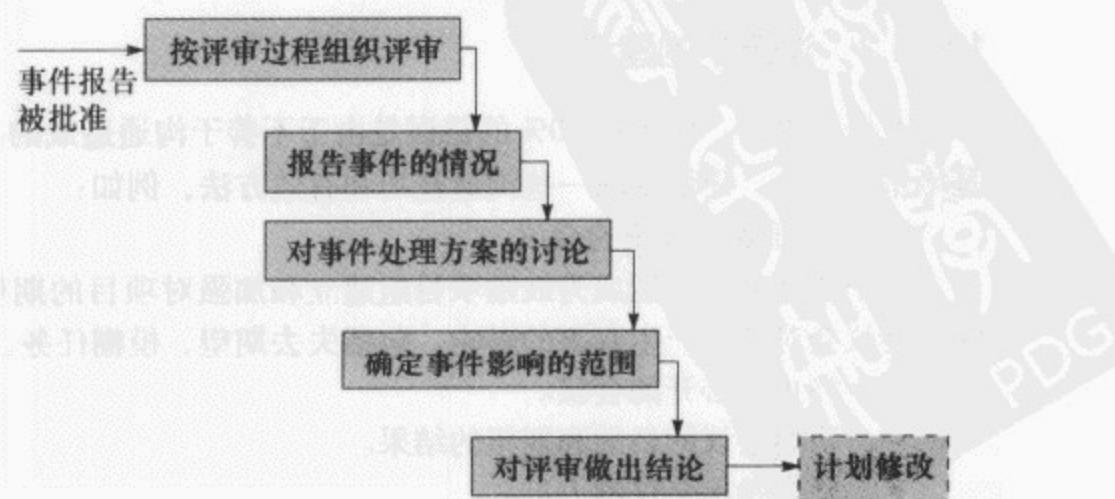


图 12-16 事件评审

评审准备要素可能包括：评审目的、评审内容、文档或产品的名称、评审方式、评审依据的规范和标准、评审议程、评审负责人、评审进入条件和完成标志、评审参加人员的姓名、角色和责任、评

审地点、评审时间安排、评审争议的解决方式、评审报告分发的对象（包括人员、角色和职责）等。

2) 评审过程。评审依据是先前开发的各种项目计划，例如范围计划、进度计划、成本计划、质量计划、配置计划、风险计划、沟通计划等等。评审采取的类型按照时间属性看可以分为定期评审、阶段评审、事件评审。

3) 评审报告。项目评审结束后需要将评审的结果，以评审报告的形式进行发布。评审报告过程是根据评审记录的结果整理评审报告，并根据评审结果和要求整理项目简报和计划修改请求，将以上文档向有关人员报告并归档的过程。评审报告包括定期评审报告、事件评审报告和阶段（里程碑）评审报告。

### 3. 非正式沟通方法

非正式沟通指在正式沟通渠道之外进行的信息传递和交流。这种沟通的优点是沟通方便，沟通速度快，且能提供一些正式沟通中难以获得的信息，缺点是容易失真。

在通过沟通获取相关信息时，除了比较正式的方法，也需要一些非正式的方法。通过提交技术报告和阅读技术报告，通过评审会议等正式的方式，的确可以得到很多的信息。但是，一些非正式的沟通，例如聊天、非正式的见面谈话等都可以很好的获取一些重要信息，而且有时这种方式更为有效和直观。所以，沟通是一个多维度的实施过程，包括写、说、听等。所以项目管理者要善于应用各种方法，引导成员进行有效沟通，而且也要注意不同的人喜欢采用不同的沟通方法。

非正式沟通是非正式组织的副产品，它一方面满足了员工的需求，另一方面也补充了正式沟通系统的不足。非正式沟通带有一种随意性与灵活性，并没有一个固定的模式或方法。

有效的沟通毫无疑问可以让员工更加无障碍地工作。但是，过分正式的沟通反而会带给员工新的压力。要想避免这一点，就要抓住员工的心理特点。当员工的积极性还没有高涨起来的时候，沟通可以带来压力，也可以带来信心。但是当激励已经到了一个阶段以后，正式的沟通会让员工认为自己做得还不能令管理者满意，因此他们会感到沮丧而不是喜悦。这个时候，就需要通过非正式的沟通方式来和员工交流。

对于一些很及时的消息可以通过口头的方式沟通，口头交流有时是很有效的方式，而且也不容易产生误会，而且有更加真切的感觉。人与人之间最常用的沟通方法是交谈。有效、实时的方式发布重要的信息。有统计表明，在面对面的沟通中，58% 的沟通是通过肢体语言实现的。

在非正式沟通中可以注意以下技巧：以询问替代命令、态度要平等、避免无聊空谈、进行愉快的沟通等。

另外在沟通信息过程中，注意介质的选择，硬盘、电话、邮件、会议、还是网络等，不同的介质适合不同的场合和需要，可以根据情况确定，并给予说明。

## 12.6.2 沟通中的建议

有研究表明：管理中 70% 的错误是由于不善于沟通造成的。目前，很多项目沟通管理不畅通甚至是堵塞。为此，建议采取一些沟通技巧和有效方法，例如：

### (1) 召开成功有效的会议

一个成功的会议能成为鼓励项目组建立和加强对项目的期望、任务、关系和责任的工具。一个失败的会议会对项目产生有害的影响，包括失去期望、模糊任务、混淆关系、推卸责任等。为此，建议：

- 终止不必开的会议。
- 明确会议的目的和期望的结果。
- 确定参加会议的人员。
- 在会议之前向参加者提供议程。
- 事先准备好资料、辅助设施和后勤安排。
- 使会议专业化。
- 建立积极的关系。

项目进行过程中，会有很多的会议，但是，有一些会议是没有主题、没有内容、没有结果的，导致了很多会议是没有效率的。所以，项目管理者应该培养召开有效会议的习惯，如果一个会议可以不开，就不要开。如果召开会议，首先要确定会议的目的、结果等，确定谁可以参加会议，提供会议的时间安排，准备会议需要的文件，以及相关的后勤安排，会议需要紧扣主题、同时保持良好的关系和氛围。

### (2) 培养沟通技巧

有些人似乎天生就有很好的沟通技能。有些人则有学习技术技能的诀窍，但很少发现有人天生就拥有上述两种技能。沟通技能和技术技能可以通过学习来提高，有意识的培养自己的听、说、写的能力，观察和学习在某方面有特长的人的特点，在实际工作、生活中学习和提高自己的沟通技能是非常有效的方法。

很多的IT企业容易忽视员工沟通技巧的培训，忽视说、听、写的训练。而且随着企业的日益强大以及全球化，更需要与不同习惯的人、不同文化背景的人沟通，所以，改善沟通技能是很重要的。

所有与良好沟通相关的技巧中，倾听可能是其中最重要的一种。据说，斯多葛学派哲学家爱比克泰德曾在《金玉良言》中说：“大自然给了人类一张嘴，两个耳朵，就是想让人们多听少说。”大多数人并没有意识到他们听的能力有多差，听懂别人所说的不容易。员工不仅要听上级说什么，还要听出上级没说什么。

要学会移情聆听，移情聆听是指以理解为目的的聆听，听者要站在说话者的角度看世界，理解他们的思维模式和感受。移情聆听时，最好重复对方的讲述内容，并反映出感情色彩。

倾听技巧强调主动倾听和反应性倾听。主动倾听是指聆听信号的全部意思，不预加判断或诠释，或者想着接下来我该说什么来响应。反应性倾听是建立在移情基础上的一种沟通工具。它可以帮助我们体验他人的想法和感受；不是为他人承担责任，不是自己说而是让他人说；不用评价、判断或给出建议，而只是略做表示。可以在谈话结束后，再来纠正谈话者的错误。反应性倾听者擅长“开放式问句”，如“你能告诉我更多情况吗？”或“那时你的感受如何”——应避免评论式、事实性或否定的问句。反应性倾听不是直接告诉，不应打断对方，而应帮助谈话者从中去发现问题。经验表明，这一技巧的回报极高。

### (3) 开发过程中的交换角色

在软件开发过程中可以尝试着让开发人员交换角色，明白别人的工作内容，让每个开发人员都清楚自己所做的工作在整个系统中处于什么地位。团队成员能力参差不齐，成对编程会让开发速度更快，代码质量也会更高。而且，最重要的是，这样，很有利于团队协作精神的培养。只要坚持就能形成在重大项目中的团队开发氛围，发挥各人的特点和长处，更能使得项目顺利进行。

### (4) 提供项目模板

IT企业的很多不太喜欢找别人帮忙，喜欢自己处理，但是团队合作是需要协同工作的。所以，在项目过程中为了节约时间、成本，最好为大家提供可以供参考的例子和模板。这样，既方便使用者，也可以便于其他人的阅读和了解，对整个团队的统一也是必须的。

### (5) 有效处理冲突

在项目进行过程中，存在冲突是正常的现象，而且有时也是很重要的方式，很多好的主意，好的选择是通过冲突产生的，而且通过冲突也可以激励更加努力和需要更加协作。没有冲突可能不会有好的想法。研究表明与项目相关的冲突可以提高团队的性能。但是情绪的冲突会有反作用的。

按照冲突解决问题的程度，可以有如下的排序：

- 解决问题（confrontation or problem-solving），直接面对冲突（问题）的方法，共同分析问题，找到最恰当的解决方案，允许受到影响的各方一起沟通，以消除他们之间的分歧。是从根本上解决问题的方法。
- 妥协（compromise），通过讨价还价，试图找到一定程度上使得双方满意的解决办法或者折中方案，双方各让一步，达成一致，虽然没有赢家，但是双方都得到了一定程度的满意。它是

### 第二种冲突解决的方法。

- 强迫方式 (forcing mode)，强调一致性，淡化分歧，努力保持友好的氛围，是求同存异的解决方法，暂时有效，但不能从根本上解决问题。是一种非输即赢的方法来解决冲突。它有一个赢家，一个输家。
- 撤退 (withdrawal)，从一个实际的或可能的不同意见中撤退或让步。冲突中的一方从冲突中退出，虽然不能解决冲突，但是暂时冷却冲突的局面，只适合于某些情况。

### (6) 定期聚会

项目组员之间的交流和沟通是很重要的，经常的交流和沟通能让大家在感情上觉得是一家人。缺少沟通只能让大家感觉是工作上的同事关系。要定期搞一些聚会是不错的选择，很多话可能在非工作场合就很容易说出来。在关系融洽的基础上则能更好地开展思想工作了，只有人心顺了才能工作顺。

下面是一个有关沟通在项目监控过程中作用的案例：这是一个客户高度关注的大型软件项目，进度已经相对滞后。正当所有的人都在紧张地追进度时，客户方的一个负责人突然提出修改进度计划，将其中系统的一部分提前上线。因为项目的数据移植和切换都是作为一个整体来处理的，此方案需要增加接口开发和数据移植的工作量；其次，切换方案的修改会影响已经展开工作，这些都对项目的最终交付日期会造成巨大的风险。但是，无论项目经理如何解释，对方仍坚持要求公司增加人手，部分提前上线。

项目经理与项目总监进行了分析，首先认为这样的决策可能是更高级别做出的。这点很快得到了验证，是客户的一个部门经理的决定；要求变更的负责人只是执行者，跟他解释无法改变决定。

根据这一情况，项目总监与客户方的部门经理进行了沟通，获知了变更计划的真正的原因。由于项目延期，上级对项目能否按期完成忧心忡忡；部门经理决定通过部分提前上线的方式“展示”项目的执行力。看来，决策的症结还不在部门经理这里。

项目总监求助公司高层与客户高层进行了一次沟通，了解到高层真正的担忧并不是“延期”本身，而是项目信息不透明，过程不可控，“没人说清楚到底要延到什么时候！”

至此，状况基本清晰。项目总监再次与客户的部门经理进行沟通。首先，告知客户高层的真正担忧，对于这个反馈信息部门经理非常重视，也很感谢，立刻拉进了与项目总监的距离。其次，项目总监提出，部分提前上线的方式其实没有解决高层担忧的根本问题，且最终的交付风险增大，只是风险后移，最终可能使其处境更为严峻。最后，建议不要修改计划，而是主动将项目的进展状态和问题向高层汇报，使得项目过程透明，将所有人的精力集中到按预期交付项目上。客户接受了建议，沟通和透明增加了高层的信心，改变了项目的处境，而且项目组还获得了很多意想不到的资源。

从一个变更计划的需求之下，反映的是不同的需要：项目负责人，是执行者，需要是坚决的贯彻部门经理的决策；部门经理，是决策者，需要证明项目的执行力，增加高层信心；客户高层，决策的“影响者”，其担忧的内容直接影响决策。可见，一个需求下面有三种“需要”。要统一认识，必须回到整体的项目目标“按期上线”（在这个目标上各层很容易达成一致）。此目标下，最有效的措施不是提前上线，而是增加项目的透明度，而且，这个措施同时可以满足所有人的需要。通过这个措施，将所有人的注意力聚焦到确保最终交付日期的目标之下，从而获得了项目的驱动力。回到案例的起点，如果项目经理仅仅匆忙决策满足客户需求，向公司施加压力要人，及时按期完成了部分系统，最后仍可能成为项目整体延期的罪魁祸首。这就是一个普通项目经理和一个优秀项目经理的本质区别。

### 12.6.3 项目性能报告

项目的性能报告也是沟通的一个重要部分，它使项目干系人知晓为了达到项目的目标，是如何使用资源的，这些资源的状况和对项目的贡献等。

评审报告就是很重要的一种性能报告，项目评审结束后需要将评审的结果，以评审报告的形式进

行发布。评审报告是根据评审记录的结果整理的，并根据评审结果和要求整理项目简报和计划修改请求，将以上文档向有关人员报告并归档的过程。评审报告包括定期评审报告、事件评审报告和阶段（里程碑）评审报告等等。

项目性能报告的输出也常分为状态报告、进度报告、项目预测等，状态报告和进展报告就是很常见的性能报告。状态报告（status reports），它描述了项目在特定的时间点，项目进展如何？是超前还是落后了？进展报告（progress reports）描述了项目团队在规定的时间段内完成任务情况，完成了哪些任务等。项目预测是在过去资料和发展趋势的基础上，预测项目未来的状态和进度。对于一些重要的信息应该有说明的，很多人常常忽略这点。

表12-14所示的项目报告，它说明项目到这周为止项目进展的如何，是一种状态报告。

表12-14 项目周报告

日期：2008.12.20

项目名称	银行业务系统	项目经理	韩万江
评审类型	定期	阶段名称	调试阶段
评审时间	2008.12.20	评审内容	周计划完成情况
本项目进展状态及存在的问题			
<p>本周完成了bis前台、后台的联调工作</p> <p>测试成功交易：</p> <ul style="list-style-type: none"> <li>030101 全行启动</li> <li>030201 部门系统启动</li> <li>030202 部门系统结束</li> <li>030201 部门柜员状态查询</li> <li>030303 柜员修改密码</li> <li>030405 柜员属性维护（添加柜员）</li> <li>030406 柜员权限维护</li> <li>030407 终端属性维护</li> <li>030409 部门权限维护</li> <li>测试案例设计</li> <li>完成交叉评审工作</li> </ul>			
项目后期展望			
<p>下周计划：</p> <ul style="list-style-type: none"> <li>i. 学习网络安全知识，了解CA认证与签名，了解windows2000 server中的证书管理</li> <li>ii. 培训UML面向对象的可视化建模语言</li> <li>iii. 同时解答前台调试中可能出现的问题</li> <li>iv. 完成一卡通定期部分开发环境调试</li> <li>v. 完成零售定期部分：610103, 610104, 610202, 610203, 610402 开发环境调试</li> </ul>			
发布对象	项目组所有人	审批人	韩万江
撰写人	孙泉	审批人	韩万江

#### 12.6.4 问题管理

我们在开发过程中不可能是一帆风顺的，可能不时地会遇到各种各样的问题，而如何来解决问题，或者说是如何想办法尽早的解决问题，这才是关键。不能有了问题而一声不响，闷头苦干，结果几天下来以后，却发现自己还是站在原地，而就算是你通过了几天的努力完成了这个难题。这样不是意味着成功，这样的结果是不但自己的进度没有办法完成，更加延误了整体的开发进度，别的成员或者是小组就可能因为你一声不响地没有成果的努力而不能再继续下面的开发。应该说软件开发过程中遇到问题一声不响、埋头苦干的做法是很愚蠢的，软件开发要求的不是个人英雄主义精神而是团队的整体合作精神。

就开发人员而言，一旦碰到了难以解决的问题，不仅要自己努力调查，想办法解决。一方面也要把存在的问题向管理者反映，让管理者能够知道存在的问题，而管理者可以在进度会议，或者召开临时紧急会议，把问题摆出来，通过大家来寻求解决的方案。一个人的力量毕竟是有限的，而个人的英

雄主义却是团队开发的极大阻碍。

发现和解决问题是项目管理工作非常重要的一个方面，也是项目沟通过程要完成的任务。解决问题包括明确问题和制定解决方案两方面的组合。它所关注的是那些已经出现的问题。（风险管理涉及的是潜在的问题）。明确问题要求将原因和现象进行区分，问题可能出自于内部，也可能来源于外部。问题可能出在技术上，也可能出在管理上或是出在人员上。

项目经理应该时时监控潜在的问题，例如：计划偏差的问题、超预算的问题、客户需求的问题、范围变更的问题、质量的问题、相关支持的问题、资源的问题、角色和职责的问题、骨干人员流失的问题等。及早做好预防，将隐患消灭在萌芽状态。一旦明确了问题的解决方案，就必须实行，解决方案是具有时间性的，如果解决方案制定得太早或太晚，那么即使是正确的解决方案也不一定是最好的解决方案。

项目经理可以根据沟通计划，确定合理有效的沟通方式。对于特别重要的内容要采用多种方式进行有效沟通，其中项目评审是很重要的项目沟通方式。

项目评审后，如果没有问题，一切按照计划进行，项目经理就会很轻松。但如果有了问题，就需要有一个问题跟踪列表，而这个问题跟踪列表正是需要项目经理关注和跟踪的事项。这个问题列表可以放在跟踪控制系统中，格式可以随项目情况而定。表 12-15 是一个问题跟踪列表的例子，这个表中列出项目中存在的问题、责任的落实、规定的解决的期限、以及问题的状态。如果问题是 open 状态表示问题没有解决，当相关责任人解决问题后，此问题变成了 Reserved 状态，表示问题已经解决，如果项目经理认为此问题已经彻底解决，则此问题变为 Closed（关闭）状态。所以，项目经理应该时时关注还有哪些问题是 Open 状态，跟踪去解决，直到问题跟踪列表的所有问题都关闭了为止。

表 12-15 问题跟踪表

序号	问题描述	措施	责任人	期限	状态
001	设计有误	设计者立刻修正	姜岳尊	3.8	Open
002	查询模块测试出现问题	开发人员修改	孙泉	3.2	Closed
003	产品没有及时入配置库	配置管理者负责纠正	岳好	3.15	Reserved
:					

下面是一个项目沟通不利导致项目失败的案例：某项目开展之初，客户和开发方技术人员的积极性还是很高的。前期的几次碰头会上，基本确定了软件功能/具体模块设置等问题。经过了一个月左右的时间，软件的部分模块已经制作完成并交付使用，客户方对进度和软件功能表示满意。但就在此时，开发方团队的技术人员发生了一定的变动。可以想象，在这次人员变动中，技术人员的工作交接出现了一些问题，以至于新的技术人员对于软件需求不甚了解甚至存在误解。在这样的背景下，软件的第二部分集成模块的功能没能得到客户的认可。恰逢此时，对于需求方面，国家有了新的规定，这就导致该软件的开发方向要做出一定的调整。由于沟通的不足，矛盾就此产生。客户认为开发方没能按时提交工作成果，功能上存在问题；而开发方认为客户的需求不断改变，难以及时实现。在随后的几次碰头会上，经常出现双方对于软件应有功能各执一词的现象，双方的分歧越来越大，甚至出现新开发的模块根本没有被部门使用的情况。久而久之，开发方的积极性越来越低，客户对于开发方的信心也愈加不足。最终导致了该软件开发项目的失败。

软件开发团队内部沟通存在问题导致其对软件的需求和评价出现偏差；开发方与使用方沟通存在问题——导致双方不信任感增加，工作积极性下降。在整个项目失败的过程中不难看出，沟通不利是其中相当重要的因素。

总结其中的教训，为了加强软件开发项目中人员的沟通，应该制定完备有效的沟通机制，提高供需双方的沟通意识，采取多种沟通方式，提高沟通的有效性，是从制度上加强沟通的办法。与此同时，通过明确由于沟通不善导致损失的责任归属，在软件需求出现变更时及时调整项目目标及进度，也是

与此相关的工作事宜。人是项目活动中最重要的资源，项目的一切活动都是由人去完成的。只有从思想上加强对于人员沟通的重视程度，才能保证软件开发团队的工作效率，确保项目及时、有效的进行并完成。

## 12.7 风险控制

由于软件项目具有不确定性的突出特性，使它天生具有很高的风险。因此项目经理必须进行风险管理，如图 12-17，风险管理的四个步骤是循环进行的，在项目的进行过程中，需要不断的进行风险识别、风险分析、风险规划和风险控制。

风险控制是实施和监控风险管理计划，保证风险计划的执行，评估和削减风险的有效性，针对一个预测的风险，监视它事实上是否发生了，确保针对某个风险而制定的风险消除步骤正在合理使用；同时监视剩余的风险和识别新的风险，收集可用于将来的风险分析信息的过程。风险跟踪控制是贯穿项目始终的，当变更发生时，需要重复进行风险识别、风险评估以及风险对策研究一整套基本措施。

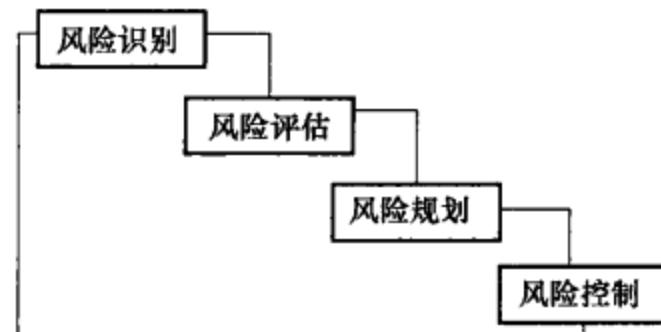


图 12-17 风险管理过程框架

### 12.7.1 风险控制方法

风险控制涉及整个项目管理过程中的风险应对过程，该过程的输出包括应对风险的纠正措施以及风险管理计划的更新。

软件企业在进行项目管理的过程中，必须采用适合自己的风险管理方法进行风险管理，以确保软件项目在规定的预算和期限内完成项目。在软件项目管理中，TOP 10 风险清单是很常见的一种表达风险计划的形式，在控制项目风险的时候，这个清单也起到很好的作用。

风险控制就是通过对风险的规划和对项目全过程的监视，从而保证风险管理能达到预期的目标，是项目实施过程的一个重要工作。其目的是核对风险管理的策略和实施的实际效果是否与预见的相同，同时获取反馈信息，改善风险计划。首先需要建立风险控制体系，然后评审和评价风险。

#### (1) 建立项目风险监控体系

项目风险监控体系的建立，包括制定项目风险的方针、程序、责任制度、报告制度、预警制度、沟通程序等方式，以此来控制项目的风险。

在软件项目管理过程中，应该任命一名风险管理者，该管理者的主要职责是在制订与评估规划时，从风险管理的角度对项目计划进行审核并发表意见，不断寻找可能出现的任何意外情况，试着指出各个风险的管理策略及常用的管理方法，以随时处理出现的风险。风险管理者可以是由项目经理以外的人担任。风险管理人员可以及时与项目的成员和客户沟通项目的情况，将实施的结果保留下来作为数据度量的资料。

#### (2) 项目风险评审评估

确定项目风险监控活动和有关结果是否符合项目风险计划，以及风险计划是否有效地实施并达到预定目标的。有系统的进行项目风险审核是开展项目风险监控的有效手段，也是作为改进项目风险监控活动的一种有效的机制。如果风险事件未被预料到，或后果远大于预料，那么计划的风险策略将会不充分，这时就有必要再次重复进行风险对策研究甚至风险管理程序，需要增加附加风险策略研究。

在项目实施过程中，项目经理应该定期回顾和维护风险计划，及时更新风险清单，对风险进行重新排序，并更新风险的解决情况，这些活动应该包含在项目计划中，以防遗忘。只有这样才能使项目经理们经常思考这些风险，居安思危，对风险的严重程度保持警惕。为了保证项目的透明性，风险清单应该向项目组的所有人公开，同时鼓励所有人员有风险意识，随时上报发现的问题。但是，报告好

的消息很少会出现问题，报告坏的消息则不然，项目组应当建立一个匿名交流渠道，这样，项目组的所有成员可以利用这个渠道报告项目进展和风险消息。例如，开发人员推迟交付代码，测试人员可以上报；如果测试人员没有充分测试就将产品写成书面文件，那么文档撰写者可以上报；如果项目经理向高层经理夸大项目的进展情况，有关的人员可以上报，等等。

通过风险跟踪控制可以时时调整风险计划，监控一个预料之中的风险事件发生或没发生，对风险事件后果进行评估，对风险概率进行评估，以及风险管理方案的其他方面都应进行实时的更新调整。针对风险清单，可以定时追加新的风险，更新风险的排序。表 12-16 是一个风险监控结果。

表 12-16 风险跟踪控制表

本周排名	上周排名	总周数	风险	风险处理情况
1	1	6	需求的逐渐增加	利用原型方法收集客户的需求，将确认的需求纳入变更控制之下，用户签字 采用分阶段提交的方式让用户逐步接受
2	5	3	总体设计出现问题	聘专家评审总体设计，提出修改建议 使用符合要求的开发过程
3	2	5	开发工具不理想	尽可能采用熟悉的工具 加强培训
4	7	3	计划过于乐观	避免在完成需求规格前对进度做出约定 早期评审，发现问题 及时评估项目状况，必要时修订计划
5	3	6	关键人员离职	挽留关键人员 启动备份的开发人员 再招聘其他人员
6	4	5	开发人员与客户产生沟通矛盾	与客户共同组织活动，增进感情 让用户参与部分开发
7	6	4	承包商开发的子系统延迟交付	要求开发商制定负责的联络人

## 12.7.2 常见风险及其处理

软件项目的风险是多种多样的，无处不在的。在项目管理活动中，要积极面对风险，越早识别风险、越早管理风险，就越有可能规避风险，或者在风险发生时能够降低风险带来的影响。特别是在项目参与方多、涉及面广、影响面大、技术含量高的复杂项目，应加强风险管理。如果不主动驾驭风险，就会面临风险。软件项目中常见的风险及其处理方法：

### (1) 项目缺少可见性

当一个项目经理或一名开发者说已经完成了 80% 的任务，您必须保持审慎的态度。因为剩下的 20% 可能还需要 80% 的时间，甚至永远都不能完成。软件项目，往往在项目进度和软件质量方面缺少可见性，项目越缺少可见性，项目就越难以控制，项目就越有可能失败。我们可以通过迭代开发、技术评审、持续集成来增强项目的可见性。

### (2) 新技术引入

技术创新是一种具有探索性、创造性的技术经济活动。在开发过程中引入新技术，不可避免地要遇到各种风险。通过原型开发、充分论证、多阶段评审、同行经验等措施可降低新技术风险。

### (3) 技术兼容性风险

硬件产品之间、系统软件（操作系统、中间件、数据库管理系统）与主机设备之间、系统软件之间、应用软件与系统软件之间以及应用软件之间，都可能存在兼容性问题。往往系统集成的项目越复杂，兼容性问题就越有可能存在。可以通过设计先行、售前产品测试等方法来降低这种风险。

### (4) 性能问题

由于先期设计不足，性能问题往往在系统切换或新系统使用一段时间后暴露。出现性能问题往往

要进行大量的优化工作，甚至局部的或全面的重新设计。无论是用户还是开发者，谁都不希望出现性能问题。可以通过性能规划、性能测试、充足的调试时间等方法来降低这种风险。

#### (5) 仓促上线

在项目实施过程中，系统切换上线环节最容易出纰漏。项目好不容易开发完成了，却在最后时刻功亏一篑。如果项目小，影响面窄倒不怎么重要；如果是影响面大的项目，则千万不可出现问题。在系统切换前，应充分考虑各种可能出现的问题，做好风险对策。可以通过应急预案、分步切换、交叉培训等方法降低风险。

#### (6) 可用性问题

软件的可用性包括软件的使用是不是高效、是否容易学习、是否容易记忆、是否令人愉快、是否不易出错等诸多因素。往往由于软件的可用性差，导致用户不满意，甚至被市场淘汰。在项目开发中应注意可用性问题，避免软件出现可用性方面的风险。可以通过了解用户、参与设计、竞争性分析等方法降低风险。

## 12.8 合同管理

在项目计划中，如果存在外包、外购等需求，就存在合同计划。对于外包、外购这部分而言，项目主体处于甲方的角色，在项目的进行过程中，需要选择合适的乙方、管理乙方合同的进行。当然乙方的角色也需要跟踪合同，对合同进行管理，保证项目符合合同的要求，按照合同的要求交付项目，并进行必要的维护。

### 12.8.1 甲方合同管理

甲方合同管理的工作主要是验收、违约的处理等。

#### 1. 阶段性验收过程

验收过程是甲方对乙方交付的产品或服务进行阶段验收检验，以保证它满足合同条款的要求。验收过程如图 12-18 所示。

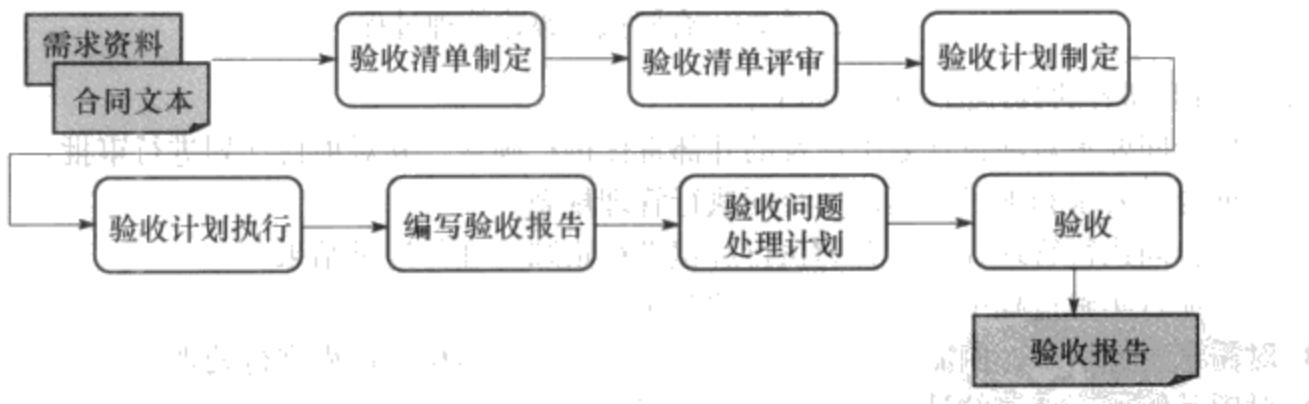


图 12-18 验收过程

具体活动描述如下：

- 1) 根据需求（采购）资料和合同文本制订对采购对象的验收清单。
- 2) 组织有关人员对验收清单及验收标准进行评审。
- 3) 根据验收清单及验收标准制定验收计划并通过甲乙双方的确认。
- 4) 甲乙双方执行验收计划。
- 5) 处理验收计划执行中发现的问题。
- 6) 起草验收完成报告。
- 7) 确定验收问题处理计划，并下达给项目经理执行。

#### 2. 违约事件处理过程

如果在合同的执行过程中，乙方发生与合同要求不一致的问题，导致违约事件，需要执行违约事件处理过程。违约事件处理过程如图 12-19 所示。

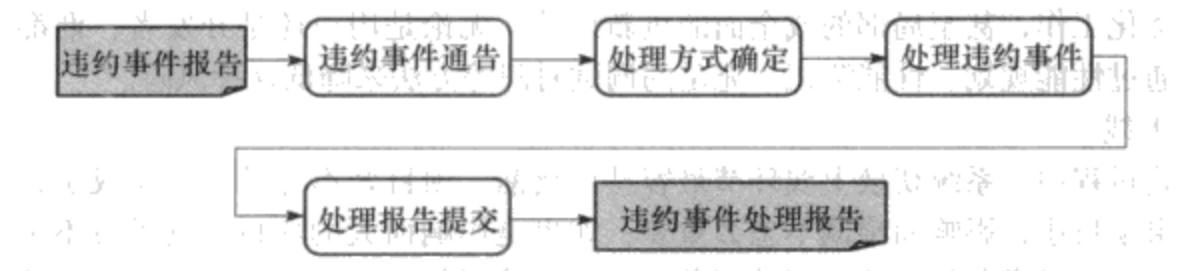


图 12-19 违约处理过程

甲方合同管理者的具体活动描述如下：

- 1) 负责向项目决策者和其他有关人员发出违约事件通告。
- 2) 项目决策者负责决策违约事件处理方式。
- 3) 负责按项目决策者的决策，处理违约事件。
- 4) 合同管理者负责向项目决策者报告违约事件处理结果。

## 12.8.2 乙方合同管理

乙方的环境下，合同管理主要包括合同跟踪管理过程、合同修改控制、违约事件处理过程、产品提交过程和产品维护过程，这些过程具体定义如下。

### 1. 合同跟踪管理过程

合同跟踪管理过程是乙方跟踪合同的执行过程，合同跟踪管理过程如图 12-20 所示。

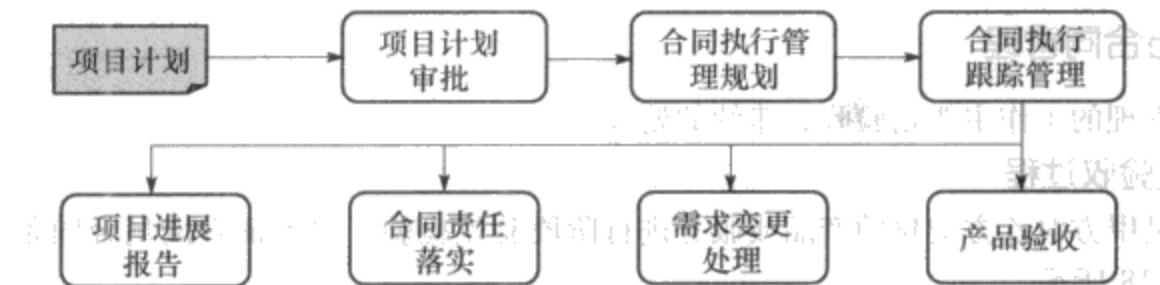


图 12-20 合同执行跟踪管理过程

乙方合同管理者的具体活动描述如下：

- 1) 根据合同要求对项目计划中涉及的外部责任进行确认，并对项目计划进行审批。
- 2) 根据合同要求和项目计划进行合同执行管理规划。
- 3) 依据合同执行管理计划对合同执行过程进行跟踪管理并记录结果。
- 4) 负责项目实施过程中甲乙双方责任的落实。
- 5) 对需求变更请求，向需求管理者下达任务单，并依据相关过程进行处理。
- 6) 对项目管理者提交的计划修改请求进行审批。
- 7) 对项目管理者提交的事件报告进行处理。
- 8) 与用户协调产品验收有关事宜。

### 2. 合同修改控制

在合同的执行过程中，可能发生合同的变更，合同修改控制就是管理合同变更的过程，合同修改控制过程如图 12-21 所示。

具体活动描述如下：

- 1) 评估变更请求对合同的影响，并与用户协商有关问题。
- 2) 根据评估结果提出合同修改建议，确定修改策略。
- 3) 根据修改策略处理修改请求，并记录合同修改备忘录。
- 4) 负责调整因修改请求处理引起的相关变化。
- 5) 向项目决策者提交修改请求处理结果报告。
- 6) 将修改请求处理结果通知有关人员。

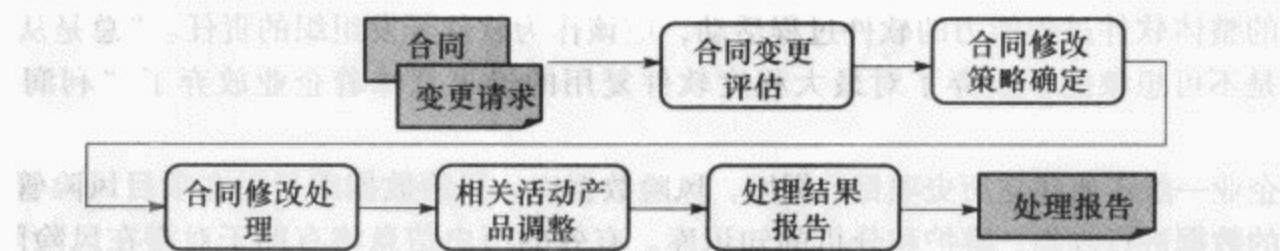


图 12-21 合同修改控制过程

### 3. 违约事件处理过程

类似甲方的情况。

### 4. 产品维护过程

产品维护过程是乙方对提交后的软件产品进行后期维护的工作过程，产品维护过程如图 12-22 所示。

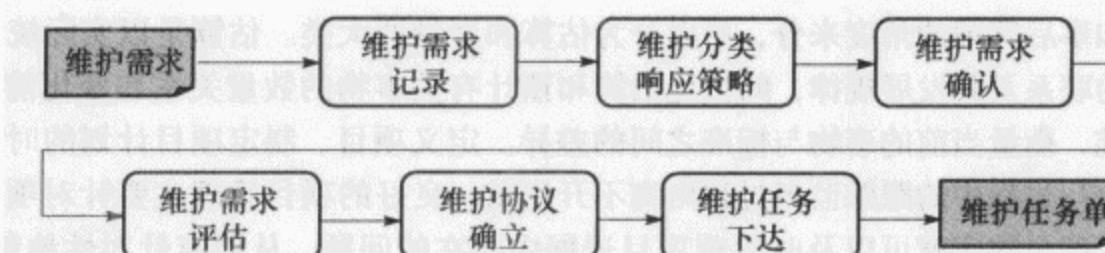


图 12-22 产品维护过程

产品维护过程具体活动描述如下：

- 1) 乙方记录用户的维护需求，对用户的维护需求进行分类，并确定响应策略。一般维护需求和响应策略有如下几种：
  - 适应性维护，按优先级排列。
  - 改善性维护，或按优先级排列，或拒绝。
  - 纠错性维护，根据错误严重程度进行优先级排列。
- 2) 乙方根据响应策略通知用户对维护需求的处理方式。
- 3) 确定乙方产品维护管理者。
- 4) 乙方产品维护管理者负责组织有关人员对用户的维护需求进行确认；对于确认过程中出现的问题，负责与用户进行协商。
- 5) 乙方产品维护管理者负责组织有关人员对用户的维护需求进行评估，估算所需的时间和资源，并将结果提交合同管理者。
- 6) 乙方合同管理者依据评估结果负责与用户进行商务协调，并与用户达成维护协议。
- 7) 乙方合同管理者依据维护协议给产品维护管理者下达产品维护任务单。由产品维护管理者负责完成产品维护过程。

## 12.9 项目度量

传统的项目管理理论著的重点都着眼于九大知识领域，却忽视了一项基础性工作：量化管理。缺乏量化管理，项目管理只能处于一种“混沌”状态。以 IT 项目为例，据称只有 26% 的项目成功地实现了范围、时间和成本目标，剩余的 74% 都有不同程度的失败。而如果采用了量化管理，项目管理的全过程就会变得“可视化”，发现问题也可以“让数字说话”。

软件企业普遍缺乏经验总结，包括个人经验和组织经验。个人经验应该以组织经验为载体，总结出各种项目的成功经验，使之规则化，把具体经验归纳为全组织的标准软件过程。个人经验不能成功组织化的软件企业是非常危险的，因为那就隐含着人员流动决定了其对组织稳定作用的流动。

改进组织的整体软件过程能力的软件过程活动，应该作为软件开发组织的责任。“总是从 0 开始”的软件企业是不可想象的，放弃了对最大程度软件复用的追求意味着企业放弃了“利润最大化”的意愿。

软件企业一般需要建立历史项目数据库，风险数据库。风险数据库是对在项目风险管理过程中收集和使用的数据进行收集、维护和分析的知识库。有效的历史信息将有助于对潜在风险问题的控制，通过对产生风险条件施加影响，最大程度地消除负面影响，使向有利与项目进展的方向发展。在风险事件发生时，在规避、转移、缓解、接受中选择最合理的应对方式。实践证明，历史信息有助于项目进行更为准确全面的计划与控制，历史的经验教训可以使项目少走不必要的弯路，少花不必要的代价，减少项目失败的风险。从项目管理的角度讲，经验是对历史信息的积累。这种总结应该以优化企业软件过程为目的，贯穿于整个项目过程中，而绝不是一份简单的项目总结报告。

软件度量是用量化的方法来评测软件开发中的费用、生产率、进度和产品质量等要素是否符合期望值，包括过程度量和产品度量两个方面。

从事前管理和事后管理的角度来分，可以分为估算和度量两大类。估算以实际统计调查资料为基础，根据事物的联系及其发展规律，间接地估算和预计有关事物的数量关系和变化前景。而度量则是依据特定的标准，衡量当前的事物与标准之间的差异。定义项目、制定项目计划的时候需要进行项目估算，而项目执行过程中的跟踪监督过程则离不开度量。良好的项目管理主要针对项目要素进行跟踪度量，通过分析度量数字就可以及时发现项目进展中存在的问题，从而有针对性地制定解决方案。通常需要度量的项目要素包括：

1) 项目进度度量。对项目进度进行定期的跟踪度量，及时发现当前进度与计划的偏差，可以及时采取措施，及时赶工或调整进度计划。

2) 缺陷度量。项目的成败直接取决于客户满意度，客户满意度是个难以量化的指标，而项目成果——产品的缺陷密度直接影响着客户的满意程度。度量产品的缺陷密度，可以有效地了解项目完成的质量。

3) 项目工作量度量。工作量是衡量项目成本、人员工作情况的基础，准确地度量出项目真实的工作量，既可以掌握当前项目的情况，对于今后估算其他项目数据也有重要意义。

4) 人员生产率度量。人力资源是项目中最为重要的资源，掌握人员的生产能力对于项目管理中人员管理、资源管理都有重要的参考价值。

项目管理是个不断完善的知识领域，引入量化管理，强化量化管理，可以使项目管理知识体系发挥更大的作用。

如何解决日益突出的项目工期、成本、质量等问题，这是大多数项目管理者最为关心的问题。从实践来看加强复用的力度，建立项目复用体系和实施组织变革是效果较好的途径之一。复用能够提高项目的生产率，降低项目风险。通过复用，项目管理者能够快速的进入项目问题定义之中，减少项目开发人员的工作量，从而尽可能的解决项目在时间、资源方面的过载问题。另外一条途径是实施项目团队的组织变革，精简项目管理机构、重新定义工作职责，制定柔性的项目工作流程，改善项目开发人员的沟通状况，提高项目人员的开发效率，努力营造一个良好的项目开发环境。这样才能从根本上解决项目开发的种种棘手问题。

## 12.10 校务通系统案例分析

校务通系统项目计划的跟踪控制过程：在本项目进展过程中，项目经理每天要根据项目计划来跟踪项目实际的执行情况，关注项目的成本、进度、质量、风险等情况，每天记录实际的进展情况，通过工具得出计划与实际的对照情况，发现问题需要及时解决。每周五进行一次定期评审，每个阶段（里程碑）结束时，进行阶段评审，并给出评审报告，评审报告应该说明项目的进展情况、费用情况、质量情况、风险情况等，并对项目将来的情况做出预测，以保证项目能够按时保质保量完成。

### 12.10.1 范围控制

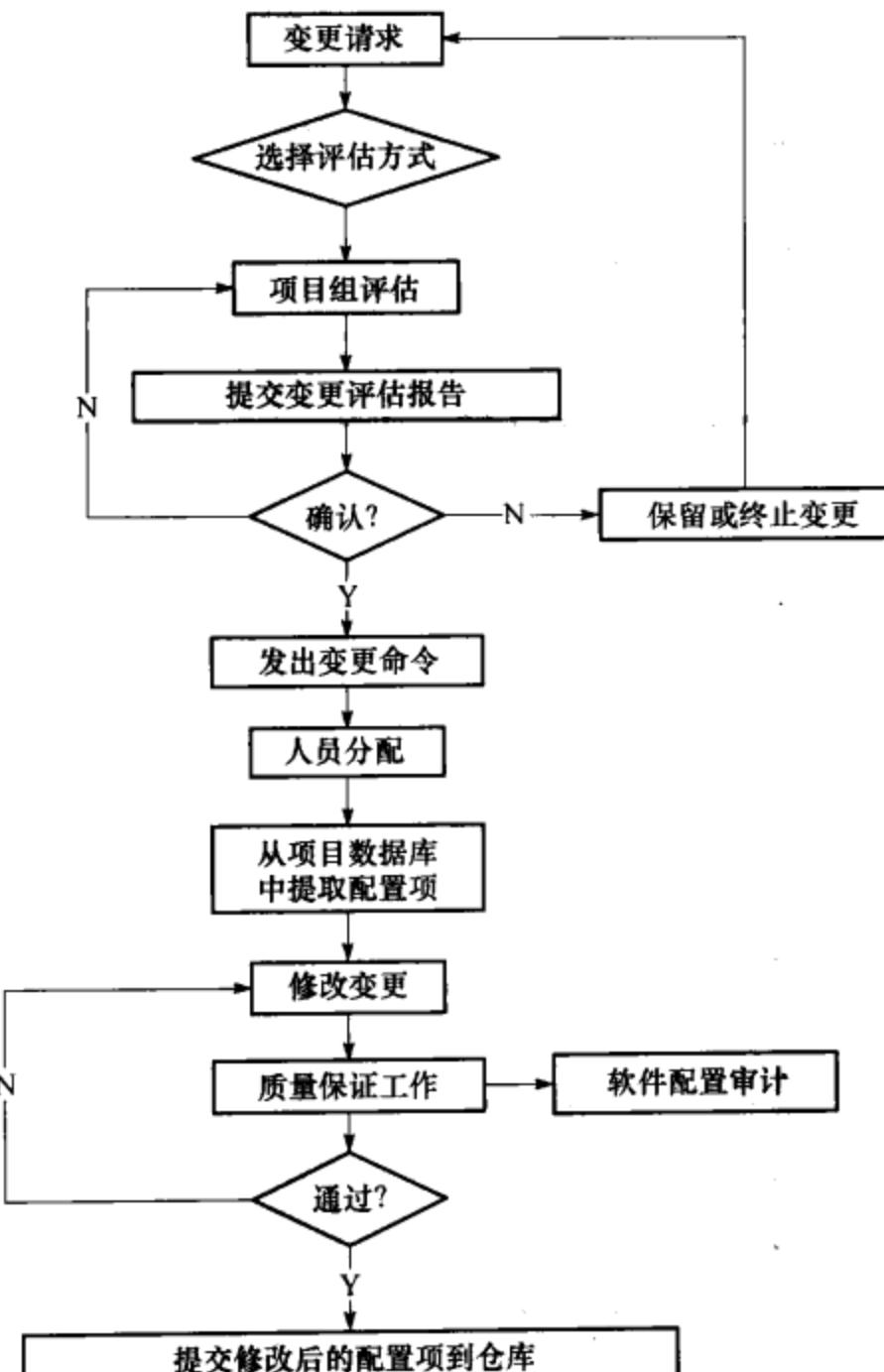
为有效的管理项目的范围变更，确定了变更控制流程系统，如图 12-23 所示，所有的变更首先提出申请，然后进行评估并提交评估报告，如果经过评估，项目组认为拒绝变更，则终止。如果项目组认为可以变更，则分配人员进行变更，变更时从配置库中取出需要修改的配置项，进行变更，变更后由质量保证人员进行审计，如果有问题，重新修改，如果没有问题将修改后的配置项纳入配置库。

本项目采用原型分析法进行需求分析，很快得到用户的确认，由于采用的是进化型原型方法，在后续的开发过程中，沿用了原型中的界面和相关的内容说明，这个非常重要，因为这是经过用户确认的。所以在项目实施过程中需求的变更比较少，而且项目管理过程中对需求变更控制的也比较好。项目进展过程中的变更主要有 2 项，这些变更申请是由用户提出的：

- 1) 增加“二次开发的接口模块”。
- 2) 增加“固定资产管理”和“办公自动化管理”。

需求变更处理如下：

- 增加“二次开发的接口模块”的处理结果：由于增加的功能基本没有波及其他的功能，而且工作量比较少，比较容易实现，所以项目组同意变更。
- “固定资产管理”和“办公自动化管理”的处理结果：此变更相当于增加了额外的功能，我们给出了工作量、成本和时间的估计，说服客户，这部分功能可以放在二期，或者单独成立一个项目来实施。经过协商，用户取消这个增加额外功能的请求。



### 12.10.2 进度、成本控制

图 12-23 变更控制流程

#### (1) 定期评审结果

图 12-24、图 12-25、图 12-26 是项目截止到 2004 年 4 月 25 日（周五，定期评审结果）的跟踪结果图示。图 12-24 显示了项目进展过程中进度情况，有实际的进度与计划基准的比较情况，可以看出进度略有超前。图 12-25 显示了成本跟踪结果，从图中可见，实际成本比计划略有减少（图中差异为负表示实际成本比计划少）。图 12-26 是项目挣值分析的结果，挣值分析表进一步分析了项目的性能情况，CPI 和 SPI 都接近 1。截止到 2004 年 4 月 25 日为止，项目完成了计划规划、需求开发、设计和增量 1 的任务，应该说是项目进展到进度的 20% 左右，CPI 应该是相对稳定的阶段，而 CPI 和 SPI 的指标值都不错，而且从成本差异 (CV) 看项目的成本略有结余。

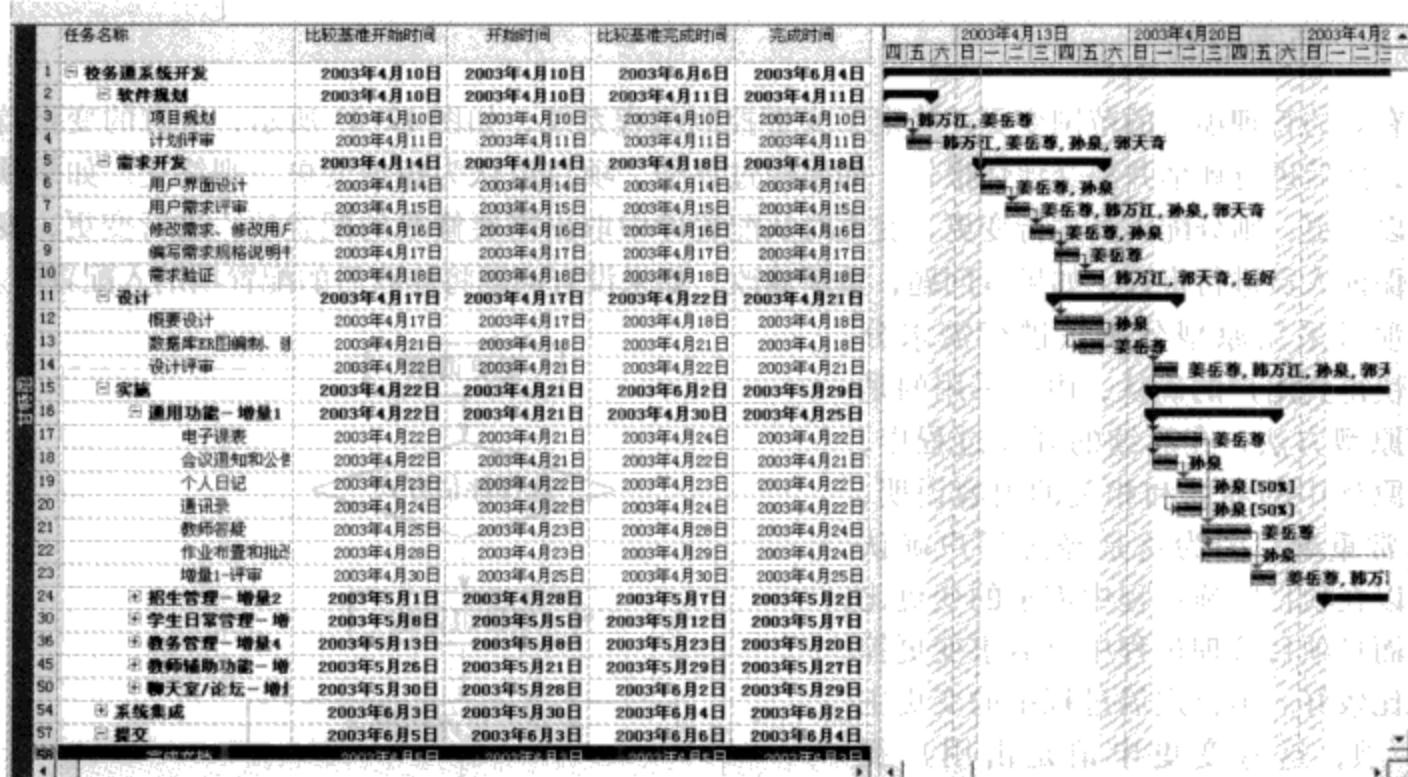


图 12-24 2004/4/25 的进度跟踪

任务名称	开始时间	完成时间	比较基准	实际成本	差异	计划	实际
1 校务通系统开发	2003年4月10日	2003年6月4日	¥75,160.00	¥23,800.00	-¥2,400.00	¥23,800.00	¥23,800.00
2 软件规划	2003年4月10日	2003年4月11日	¥3,320.00	¥3,320.00	¥0.00	¥3,320.00	¥3,320.00
3 项目规划	2003年4月10日	2003年4月10日	¥1,200.00	¥1,200.00	¥0.00	¥1,200.00	¥1,200.00
4 计划评审	2003年4月11日	2003年4月11日	¥2,120.00	¥2,120.00	¥0.00	¥2,120.00	¥2,120.00
5 需求开发	2003年4月14日	2003年4月14日	¥6,240.00	¥6,240.00	¥0.00	¥6,240.00	¥6,240.00
6 用户界面设计	2003年4月14日	2003年4月14日	¥1,120.00	¥1,120.00	¥0.00	¥1,120.00	¥1,120.00
7 用户需求评审	2003年4月15日	2003年4月15日	¥2,120.00	¥2,120.00	¥0.00	¥2,120.00	¥2,120.00
8 修改需求、修改用户界面	2003年4月16日	2003年4月16日	¥1,120.00	¥1,120.00	¥0.00	¥1,120.00	¥1,120.00
9 编写需求规格说明书	2003年4月17日	2003年4月17日	¥560.00	¥560.00	¥0.00	¥560.00	¥560.00
10 需求验证	2003年4月18日	2003年4月18日	¥1,320.00	¥1,320.00	¥0.00	¥1,320.00	¥1,320.00
11 设计	2003年4月17日	2003年4月21日	¥4,120.00	¥4,120.00	¥0.00	¥4,120.00	¥4,120.00
12 框架设计	2003年4月17日	2003年4月18日	¥1,120.00	¥1,120.00	¥0.00	¥1,120.00	¥1,120.00
13 数据库ER图编制、建库	2003年4月18日	2003年4月18日	¥560.00	¥560.00	¥0.00	¥560.00	¥560.00
14 设计评审	2003年4月21日	2003年4月21日	¥2,440.00	¥2,440.00	¥0.00	¥2,440.00	¥2,440.00
15 实施	2003年4月21日	2003年5月29日	¥54,640.00	¥10,120.00	-¥2,400.00	¥10,120.00	¥10,120.00
16 通用功能-增量1	2003年4月21日	2003年4月25日	¥12,520.00	¥10,120.00	-¥2,400.00	¥10,120.00	¥10,120.00
17 电子课表	2003年4月21日	2003年4月22日	¥1,680.00	¥1,120.00	-¥560.00	¥1,120.00	¥1,120.00
18 会议通知和公告	2003年4月21日	2003年4月21日	¥560.00	¥560.00	¥0.00	¥560.00	¥560.00
19 个人日记	2003年4月22日	2003年4月22日	¥560.00	¥280.00	-¥280.00	¥280.00	¥280.00
20 通讯录	2003年4月22日	2003年4月22日	¥560.00	¥280.00	-¥280.00	¥280.00	¥280.00
21 教师答疑	2003年4月23日	2003年4月24日	¥1,120.00	¥1,120.00	¥0.00	¥1,120.00	¥1,120.00
22 作业布置和批改	2003年4月23日	2003年4月24日	¥1,120.00	¥1,120.00	¥0.00	¥1,120.00	¥1,120.00
23 增量1-评审	2003年4月25日	2003年4月25日	¥2,440.00	¥2,440.00	¥0.00	¥2,440.00	¥2,440.00
24 招生管理-增量2	2003年4月26日	2003年5月2日	¥9,000.00	¥0.00	¥0.00	¥0.00	¥0.00
30 学生日常管理-增量3	2003年5月5日	2003年5月7日	¥6,600.00	¥0.00	¥0.00	¥0.00	¥0.00
36 教务管理-增量4	2003年5月8日	2003年5月20日	¥16,040.00	¥0.00	¥0.00	¥0.00	¥0.00
45 教师辅助功能-增量5	2003年5月21日	2003年5月27日	¥5,800.00	¥0.00	¥0.00	¥0.00	¥0.00
50 聊天室/论坛-增量6	2003年5月28日	2003年5月29日	¥4,680.00	¥0.00	¥0.00	¥0.00	¥0.00
54 系统集成	2003年5月30日	2003年6月2日	¥2,920.00	¥0.00	¥0.00	¥0.00	¥0.00
57 提交	2003年6月3日	2003年6月4日	¥3,920.00	¥0.00	¥0.00	¥0.00	¥0.00
58 完成	2003年6月4日	2003年6月4日	¥1,480.00	¥0.00	¥0.00	¥0.00	¥0.00

图 12-25 2004/4/25 的成本跟踪结果

2003.4.25项目挣值分析结果图

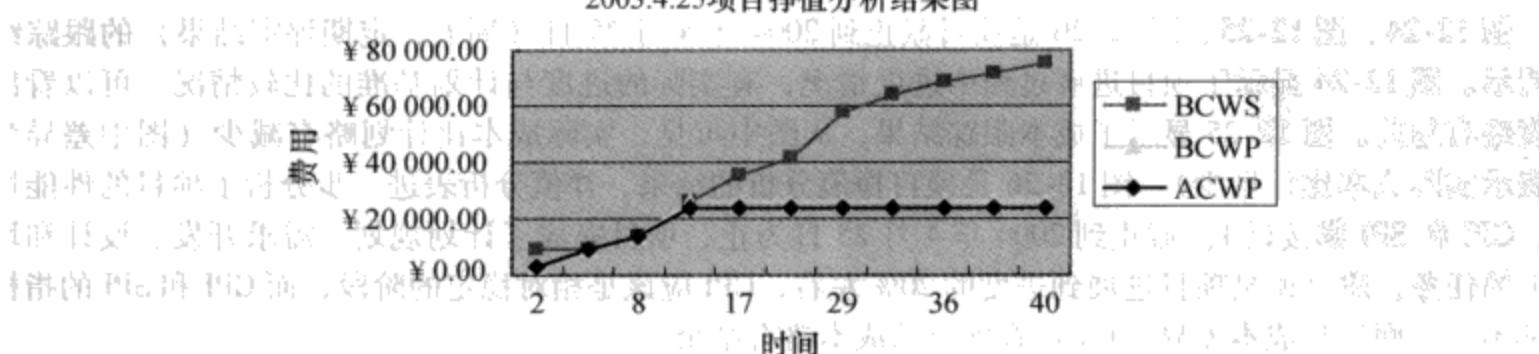


图 12-26 2004/4/25 的挣值分析结果

## (2) 阶段评审结果

在项目计划中规定了每个阶段（里程碑）结束进行阶段（里程碑）评审，表 12-17 是在增量 2 阶段结束之后进行的阶段评审的评审报告。

表 12-17 阶段评审报告

项目名称	校务通管理系统	项目标识	PRJ-SCHOOL
部门/组织名	Dev	阶段名称	增量开发 2
主持人	韩万江	会议地点	会议室
评审时间	2003.5.2	评审次数	1
评审人	姜岳尊, 杨炎泰, 孙泉, 郭天奇, 岳好		
<b>评审项与结论</b>			
评审要素	评审结果	问题和对策	
招生管理 - 实施结果 • 报名 • 招生 • 分班 • 统计查询	满足需求规格的要求	开发人员与用户一起评审，用户对本阶段的提交结果很满意，但是用户越满意越有可能提出更高的要求。	
计划执行	通过	基本按照计划执行，而且还稍有提前，但要注意后期的进度情况，从用户对增量 1 的反馈来看，用户可能会对原有的需求做出一些调整。	
质量情况	通过（见缺陷跟踪表）	满足质量计划的要求	
配置管理	有待提高	运行环境没有纳入配置管理，但是开发人员可以掌控运行环境	
其他问题	问题不大	开发人员增强校务业务学习，以便保证开发顺利进行	
计划调整	完成	根据本阶段的计划执行情况调整下一阶段的计划	
<b>提交产品</b>			
项目计划		<input type="checkbox"/>	
计划跟踪数据		<input type="checkbox"/>	
原码		<input type="checkbox"/>	
增量 2 运行系统		<input type="checkbox"/>	
评审报告		<input type="checkbox"/>	
<b>阶段统计数字</b>			
数据项目	计划	实际	偏差
工期（天）	5	5	0
规模（人时）	128	142.4	-14.4
人力投入（M/D/QA/SCM）	1/2/1/1	1/2/1/1	0
成本（元）	9200	10008	-1008
阶段日期	5.1 - 5.7	4.28 - 5.2	提前 3 天开始，提前 3 天结束
阶段评语	本阶段的成本和进度基本在控制的范围之内，而且进度略有提前。在完成本阶段目标的基础上，开发人员更加深入的熟悉技术和业务，为确定下阶段目标打下了一定的基础。从管理上，体现在生存期的定义、项目计划的细化和确认、项目跟踪日常化等方面。可以为今后类似项目的管理提供宝贵的经验。所以，本阶段除较好地完成了规定的目标，而且进行了许多有益的尝试，本阶段结束后，项目进展及完成情况属正常。		

### 12.10.3 风险管理控制

项目经理根据风险计划实时监控项目风险的情况，表 12-18 是截止到 2003.6.1 为止的风险分析表及其处理方法。

表 12-18 2003/6/1 风险分析表

排序	上周排序	输入	风险事件	风险说明	采取的措施
1	1	客户的 SOW	需求不明确，增加需求，导致需求蔓延	1. 客户提出增加“二次开发的接口模块” 2. 客户（的行政部门）提出增加“办公自动化管理”和“固定资产管理”	1. 增加接口模块，技术上实现不困难，而且波及面不大，工作量也不大，所以，同意用户的要求 2. 增加办公自动化管理和固定资产管理的工作量比较大，而且可以与现有的系统独立运行，所以，经过与客户的多次交流，最后确定作为另外一个合同实施
2	4	WBS	供货商、外包商的质量问题	“网上考试”与本系统联调时存在很多的不能兼容问题	由于“网上考试”从用户的角度看可能使用率很低，而且可能以后不会使用此功能，是可有可无的功能，经过与客户的沟通，删除此功能，同时将增加的接口模块作为补偿
3	3	历史信息	开发人员的流动	开发人员中的一个人由于出国辞职	从其他的项目组借调熟悉本项目技术的人员，而且由于开发过程中过程文档控制的比较清楚，交接工作很顺利

#### 12.10.4 质量控制

从缺陷跟踪图 12-27 可以看出，项目的缺陷在前期比较突出，随着项目的进展，缺陷的发生比例基本在项目的控制线以内。

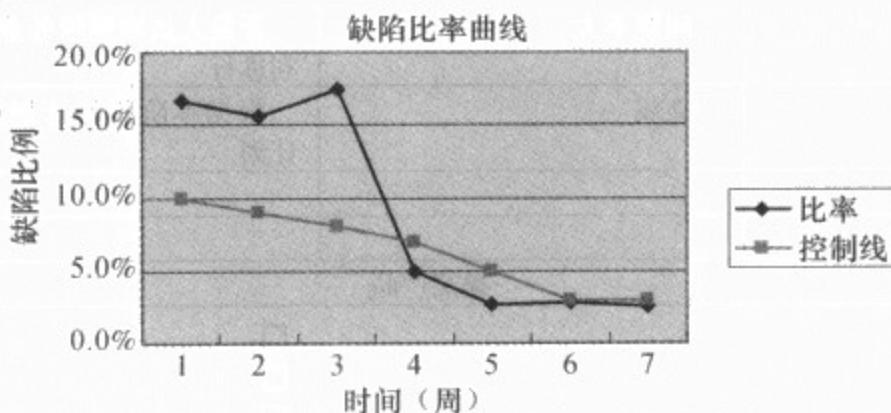


图 12-27 缺陷比例

#### 12.11 小结

项目执行控制是软件项目管理过程中非常重要的一个管理过程，它直接决定着项目的成功与否，也是体现项目管理水平的关键之处。控制是根据采集的跟踪数据，与原始项目计划进行比较，从而判断项目的性能，对出现的偏差给予纠正，必要时修改项目计划。项目的控制包括项目范围、进度、成本、资源、质量、风险、人员等方面控制。本章还重点介绍了分析项目进展性能的两种方法，即图解控制方法和挣值分析法。图解控制方法是综合甘特图、费用曲线以及资源载荷图来分析项目的一种方法。挣值分析法是利用成本会计的方法评价项目进展情况的一种方法，它是从新的角度看待成本和进度差异的一种方法。

#### 12.12 练习题

##### 一、选择题

- 移情聆听需要理解他人的观点，为了展示移情聆听的技巧，项目经理应该（ ）。
  - 检查阐述的内容是否正确
  - 重复他人的内容，并且有感情色彩
  - 评估内容并提出建议
  - 重复

2. 项目将近收尾的时候，项目职员更关注将来能够分配的任务，而不是当前的项目，下列哪项是当前项目职员的需要？（ ）  
 A. 生理需求      B. 受人尊敬      C. 安全      D. 自我实现
3. 项目原来预计 2007.9.9 完成 1000 元的工作，但是目前只完成了 850 元的工作，而为了这些工作花费了 900 元，则成本偏差和进度偏差各是多少？（ ）  
 A.  $CV = 50$  元，  $SV = -150$  元      B.  $CV = -150$  元，  $SV = -150$  元  
 C.  $CV = -50$  元，  $SV = -50$  元      D.  $CV = -50$  元，  $SV = -150$  元
4. 抽样统计的方法中，（ ）  
 A. 应该选择更多的样品      B. 以小批量的抽样为基准进行检验  
 C. 确定大量或批量产品质量的唯一方法      D. 导致更高的成本
5. 在一个项目会议上，一个成员提出增加任务的要求，而这个要求超出了 WBS 确定的项目基线，这时，项目经理提出项目团队应该集中精力完成而且仅需完成原来定义的范围基线，这是一个（ ）的例子。  
 A. 范围定义      B. 范围管理      C. 范围蔓延      D. 范围变更请求
6. 进度控制重要的一个组成部分是（ ）。  
 A. 确定进度偏差是否需要采取纠正措施      B. 定义为项目的可交付成果所需要的活动  
 C. 评估 WBS 定义是否足以支持进度计划      D. 确保项目队伍的士气高昂，发挥团队成员的潜力
7. CPI = 0.90 说明的是（ ）。  
 A. 目前的预期成本超出计划的 90%      B. 投入一元钱产生 0.90 元的效果  
 C. 项目完成的时候，将超支 90%      D. 项目已经完成计划的 90%
8. 从质量控制图的控制上限和下限，可以知道（ ）。  
 A. 可接受的过程偏差      B. 技术规范的界限  
 C. 可以接受的产品偏差幅度      D. 技术性能测量的基准
9. “为什么大家不能都让一步解决这个问题哪？”，这是哪类冲突解决方法的体现？（ ）  
 A. 解决问题（confrontation or problem-solving）      B. 妥协（compromise）  
 C. 强迫方式（forcing mode）      D. 撤退（withdrawal）
10. 当代冲突认为，（ ）  
 A. 冲突是坏事情      B. 冲突常常是有利的事情  
 C. 冲突是由捣乱分子制造的      D. 应该避免冲突
11. 项目中的小组成员要同时离开公司，项目经理首先应该做什么（ ）  
 A. 实施风险计划      B. 招募新员工      C. 与人力资源经理谈判      D. 修订计划
12. 下面说法中除了哪个，其他都不是对 QC 的描述？（ ）  
 A. 质量标准      B. 预防      C. 保证      D. 检查

## 二、计算题

- 1) 一个任务原计划 2 个人全职工作 2 周完成。而实际上只有一个人参与这个任务，到第二周末，这个人完成了任务的 75%，请计算项目的 CPI 和 EAC。
- 2) ABC 公司正在进行一个项目，见表 12-19 项目目前的一些指标。计算总的估算成本 BAC，截至 4/1/2007 项目的状况如何，通过计算 BCWP、BCWS、ACWP、SV、SPI、CV、CPI 来说明。

表 12-19 项目进展数据

任务	估计工作量（人天）	到目前为止的实际工作量（人天）	估计完成日期	实际完成日期
1	5	10	1/25/2007	2/1/2007
2	25	20	4/1/2007	3/15/2007
3	120		5/15/2007	
4	40	50	6/15/2007	4/1/2007
5	60		7/1/2007	
6	80		9/1/2007	

### 三、分析题

表 12-20 一个项目的目前的运行数据，按挣值分析原理计算项目所有的基本度量和运行指标，这个项目的进度还正常吗？假设目前的日期是 4/15/2007。

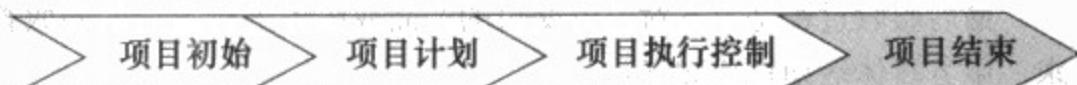
表 12-20 项目进展数据

任务	估计工作量（人天）	到目前为止的实际工作量（人天）	估计完成日期	实际完成日期
1	50	70	1/15/2007	2/1/2007
2	35	20	2/15/2007	2/15/2007
3	20	40	2/25/2007	3/1/2007
4	40	40	4/15/2007	4/15/2007
5	60		6/1/2007	
6	80		7/1/2007	

## 第四篇

# 软件项目结束

经过前面几章的学习，大家已经知道作为项目经理，首先应该做项目计划，然后实施项目计划，跟踪控制项目计划，直到项目管理的最后一步，正式接受一个项目或者项目的一部分，这时项目进入结束过程。本篇进入路线图的项目结束阶段（如下图所示）。



# 第13章

## ■ 项目结束过程

当一个项目的目标已经实现，或者明确看到该项目的目标已经不可能实现时，项目就应该终止，使项目进入结束阶段。项目结束阶段是项目的最后阶段，这一阶段仍然需要进行有效的管理，适时做出正确的决策，总结分析项目的经验教训，为今后的项目管理提供有益的经验，并有效的终止项目。项目结束条件是软件开发商和顾客之间所订立的项目最终验收条件，也是是否完成合同要求的判断标准。这个阶段的过程主要包括合同结束（Close the contract）和项目结束（Close the project）两个概念。

### 13.1 合同结束

合同结束说明完成了合同所有条款或者合同双方认可终止，同时解决了所有问题。合同是甲乙双方的事情，合同结束也应该由甲乙双方共同完成，所以，合同结束包括甲方合同结束过程和乙方合同结束过程。

#### 13.1.1 甲方合同结束过程

甲方是合同结束的主导者，在结束过程中主要包括最后验收、合同终止等过程。

##### 1. 最后验收过程

最后验收过程是甲方对乙方交付的产品或服务进行最后的验收检验，以保证它满足合同条款的要求。甲方具体活动描述如下：

- 1) 根据需求（采购）资料和合同文本制订验收清单。
- 2) 组织有关人员对验收清单及验收标准进行评审。
- 3) 根据验收清单及验收标准制定验收计划，并通过甲乙双方的确认。
- 4) 甲乙双方执行验收计划。
- 5) 处理验收计划执行中发现的问题。
- 6) 起草验收完成报告。
- 7) 双方确定验收问题处理计划，并下达给项目经理执行。
- 8) 最后双方签字认可表示验收完成。

##### 2. 合同终止

当项目满足结束的条件，合同管理者应该及时宣布项目结束，终止合同的执行，并通过合同终止过程告知各方合同终止。具体活动描述如下：

- 1) 按照企业文档管理规范将相关合同文档归档。
- 2) 合同管理者向有关人员通知合同终止。
- 3) 起草项目总结报告。

在项目的末期，与乙方的合同如果还有尚未解决的索赔，项目经理可以在合同收尾之后，采取法律行动。

#### 13.1.2 乙方合同结束过程

在合同终止过程中，乙（供）方应该配合甲（需）方的工作，包括项目的最后验收、双方签字认

可、总结经验教训、获取合同的最后款项、开具相应的发票、将合同相关文件归档等过程。

## 13.2 项目结束

项目最后执行的结果有两个状态：成功与失败。项目范围、项目成本、项目时间、客户满意度四点达到要求，可以作为“成功项目”的标准解释。项目范围、客户满意度主要代表客户的利益，项目成本主要代表开发商的利益，项目开发时间同时影响双方的利益。

一个项目如果生产出可交付的成果，而且符合事先预订的目标，满足技术性能的规范要求，满足某种使用目的，达到预定需要和期望，相关领导、项目关键人员、客户、使用者都比较满意，就是很成功的项目，即使有一定的偏差，但只要多方肯定，项目也是成功的。但是对于失败的界定就比较复杂，不能简单的说项目没有实现目标就是失败的，也可能目标不实际，即使达到了目标，但是客户的期望没有解决，也不是成功的项目。

软件项目关闭前应该完成的任务包括：

- 客户正式接受这个项目。
- 项目记录完整。
- 产品的最后版本必须满足完整的条件。
- 保留必要的项目文档。
- 准备经验学习资料。
- 转移必要的权限。

### 13.2.1 项目结束过程

一旦决定终止一个项目，项目就要有计划、有序地分阶段停止。当然这个过程可以简单地立即执行，即立即放弃项目。但是，为了使项目终止有一个较好的结果，有必要对结束过程像对待项目生存期其他阶段一样，认真执行，包括制定结束计划、完成收尾工作、项目最后评审等过程。

#### 1. 项目结束计划

项目结束计划其实已经包含在原来制定的项目计划中，只是在项目快要结束的时候，需要重新评审和细化项目结束计划，确保项目的正常结束。

#### 2. 项目收尾的内容

收尾过程是一个沟通的过程，因为大量的收尾工作要靠项目干系人同心协力来完成。包括质量评估、验证项目的成果并归档等，这个过程就是发起人和客户对项目产品的正式接收。软件项目收尾工作应该做的事情至少包括如下：

1) 范围确认。项目接收前，重新审核工作成果，检验项目的各项工作范围是否完成，或者完成何种程度，最后，双方确认签字。

2) 质量验收。质量验收是控制项目最终质量的重要手段，依据质量计划和相关的质量标准进行验收，对不合格的不予接收。如果验收人员在审查与测试时发现工作成果存在缺陷，则应当视问题的严重性与开发商协商找出合适的处理措施。如果工作成果存在严重缺陷，则退回给开发商。开发商应当给出纠正缺陷的措施，双方协商第二次验收的时间。如果给验收方带来了损失，应当按合同约定对承包商做出相应处罚。如果工作成果存在轻微缺陷，则开发商给出纠正措施后由双方协商是否需要第二次验收。

项目质量验收看起来属于事后控制，但它的目的不是为了改变那些已经发生的事情，而是试图抓住项目质量合格或不合格的精髓，以使将来的项目质量管理能从中获益。项目质量验收不仅仅是在项目完成后进行，还包括对项目实施过程中的各个关键点的质量评估。

3) 产品交付。当所有的工作成果都通过验收后，乙方向甲方提交最终产品的过程，具体活动描述如下：

- 乙方依据合同要求对提交的产品进行检查，检查内容包括：产品名称、产品版本、产品提交介质、产品提交数量、产品提交形态等。
- 乙方依据合同规定的产品提交方式将产品提交用户。
- 乙方负责完成《产品提交说明》中双方的签字，表明项目正式接收。正式接收是项目发起人或客户签发的表明他们接受项目产品的文件。
- 乙方将最终结果通知项目决策者、项目管理者及财务等有关人员。

4) 费用决算。费用决算是指从项目开始到项目结束全过程所支付的全部费用进行核算，并编制项目决算表的过程，财务人员负责支付合同余款。

5) 项目文档验收。收尾阶段需要花时间来汇集项目的文件，检查项目过程中的所有文件是否齐全，然后进行归档以供将来使用，为企业提供了一个项目准确的历史。

### 3. 项目最后评审

项目结束中一项重要的过程是项目的最后评审，它是对项目进行全面的评价和审核，主要包括：确定是否实现项目目标，是否遵循项目进度计划，是否在预算成本内完成项目，项目过程中出现的突发问题以及解决措施是否合适，问题是否得到解决，对特殊成绩的讨论和认识，回顾客户和上层经理人员的评论，从该项目的实践中可以得到哪些经验和教训等事项。

基于评审结果准备一个项目结束报告。在评审会议上，项目成员可以畅所欲言地说出自己的想法，而且这些想法对企业也可能很有好处。

## 13.2.2 项目总结

项目结束的最后一个过程是项目总结。项目总结是将实际运行情况与项目计划不断比较以提炼经验教训的过程。通过项目总结，项目过程中的经验和教训应该得到完整的记录和升华，成为“组织财富”。

很多项目没有能进行很好的总结，推脱的理由：项目总结时项目人员已经不全了、有新的项目要做，没有时间写，没人看等等。这些理由全是不正确的，无论如何也要进行总结，只有总结当前，才能提高以后。项目的成员应当在项目完成后，为取得的经验和教训写一个《项目总结报告》，总结在本项目中哪些方法和事情使项目进行的更好，哪些为项目制造了麻烦，以后应在项目中避免什么情况等等。总结成功的经验和失败的教训，为以后的项目人员更好的工作提供一个极好的资源和依据。无论项目成功还是失败，项目结束后可以根据项目规模的大小，适当的款待项目成员，比如可以款待项目团队晚餐，给他们放假，或者度假之类的。

最后，要对软件项目过程文件进行总结，将项目中的有用信息进行总结分类，放入信息库。

一个项目的交付验收，并不意味着项目的真正结束，一个优秀的项目管理人员善于在项目结束后进行总结。项目总结工作当然要包括那些没有预料到而发生的需求变更，以及这些变更的应对措施。例如根据项目实际工作中遇到的需求变更管理的问题，可以总结如下：

- 良好气氛下的充分交流。讨论需求及变更需求时，需求人员与客户及用户应该尽量采取协作的态度，良好的工作氛围也会提高工作效率，很难想象双方在“刁难”与“对付”的态度下是一种该有多糟糕的工作场景。确定需求基线的过程也就是与用户交流的过程，而频繁大量的需求变更在很大程度上也是交流不充分的后果。所以，有效的充分的交流尤为重要，需求人员认真听取用户的要求，进行分析和整理。同时还应该有能力设想项目的开发过程中可能会遇到的由该需求导致的问题，同时要让客户认识到如果此时再提出需求变更，将会给整个项目带来的各种影响和冲击。
- 专职人员负责需求变更管理。在具有相当规模的项目中，专职的需求人员和由此组成的需求变更执行小组是项目稳定、进度良好的保证。没有变更管理而直接由开发人员处理的需求变

更将会给项目带来毁灭性的灾难。这些专职人员应该具有专业的需求分析技巧技能，针对用户的变更需求，可以给用户说明利弊，可以按紧迫程度为开发人员提供工作重点，同时，他们应该还能控制需求变更的频率。

- 明确合同约束，限制需求变更。需求在软件项目中的地位已经越来越重要，需求变更给软件开发带来的影响也是有目共睹的，甚至可能因为质量低下的需求或者频繁无控制的需求变更而导致项目的失败。因此，应该让客户明白需求变更给项目带来的工期、成本等各方面的影响，在互相理解的基础上增加合同条款，比如明确说明客户可以提出需求变更的期限，超过期限的需求变更的具体处理细则（如增加开发费用等，需求变更与开发费用本身也是关联的，这个要求并不过分）。
- 良好的软件结构适应需求变更。优秀的软件体系结构可以快速应对不同情况的需求变更，这样就可以适当降低需求的基线（当然是在成本影响的允许范围内），从而来提高客户的满意度。适应需求变更必须遵循一些设计原则，如松散耦合、合理的接口定义等，要力求减少对接口入口参数的变化。

## 13.3 项目管理的建议

随着软件开发的深入，各种技术的不断创新以及软件产业的形成，人们越来越意识到软件过程管理的重要性，管理学的思想逐渐融入软件开发过程中，软件项目管理日益受到重视。项目管理既是一个科学也是一门艺术，不同的项目、不同的项目经理有不同的管理方法和技巧，不可以照搬照抄，要因地制宜。

### 13.3.1 常见问题

目前我国大部分软件公司，无论是产品型公司还是项目型公司，都没有形成完全适合自己公司特点的软件开发管理模式，虽然有些公司根据软件工程理论建立了一些软件开发管理规范，但并没有从根本上解决软件开发的控制问题。软件项目管理常见问题：

1) 缺乏项目管理系统培训。在软件企业中，以前几乎没有专门招收项目管理专业的人员来担任项目经理，被任命的项目经理主要是因为他们能够在技术上独当一面，而管理方面特别是项目管理方面的知识比较缺乏。

2) 项目计划意识问题。项目经理对总体计划、阶段计划的作用认识不足，因此制定总体计划时比较随意，不少事情没有仔细考虑；阶段计划因工作忙等理由经常拖延，造成计划与控制管理脱节，无法进行有效的进度控制管理。

3) 管理意识问题。部分项目经理不能从总体上把握整个项目，而是埋头于具体的技术工作，造成项目组成员之间忙的忙、闲的闲、计划不周、任务不均、资源浪费。有些项目经理没有很好的管理方法，安排不好工作只好自己做，使项目任务无法有效、合理地分配给相关成员，以达到“负载均衡”。

4) 沟通意识问题。在项目中一些重要信息没有进行充分和有效的沟通。在制定计划、意见反馈、情况通报、技术问题或成果等方面与相关人员的沟通不足，造成各做各事，重复劳动，甚至造成不必要的损失。有些人没有每天定时收邮件的习惯，以至于无法及时接收最新的信息。

5) 风险管理意识问题。有些项目经理没有充分意识到风险管理的重要性，对计划书中风险管理的章节简单应付了事，随便列出几个风险，随便地写一些简单的对策，对于后面的风险防范起不到什么指导作用。

6) 项目干系人问题。在范围识别阶段，项目组对客户的整体组织结构、有关人员及其关系、工作职责等没有足够了解，以至于无法得到完整需求或最终经权威用户代表确认的需求。或者是多个用户代表各说各话，昨是今非，但同时又要求项目尽早交付。项目后期需求变化随意，造成项目范围的蔓

延，进度的拖延，成本的扩大。

7) 项目团队内分工协作问题。项目团队内部有时责任分工不够清晰，造成工作互相推诿、责任互相推卸的现象。有时各项目成员只顾完成自己那部分任务，不愿意与他人协作。这些现象都将造成项目组内部资源的损耗，从而影响项目进展。

### 13.3.2 项目管理的经验和建议

软件项目管理比较难做，主要表现在工期短，客户对自己的需求描述不明确，沟通成本高，客户对技术要求有限制等等。实施有效的项目管理绝非易事，对于软件企业而言，这不是一个小的改变，而是一种变革，企业需要为此付出艰苦的努力，同时，成熟有效的项目管理无疑将对企业起着至关重要的作用，项目管理的水平将是企业核心竞争力之一。有些组织在项目管理方面有很多的经验还是值得借鉴的，这里摘录一些简短的经验和建议与大家分享：

1) 平衡关系。需求、资源、工期、质量四个要素之间的平衡关系问题。需求定义了“做什么”，定义了系统的范围与规模，资源决定了项目的投入（人、财、物），工期定义了项目的交付日期，质量定义了做出的系统好到什么程度，这四个要素之间是有制约平衡关系的。如果需求范围很大，要在较少的资源投入下，很短的工期内，很高的质量要求来完成某个项目，那是不现实的，要么需要增加投资，要么工程延期。如果需求界定清楚了，资源固定了，对系统的质量要求很高，则可能需要延长工期。对于上述四个要素之间的平衡关系最容易犯的一个错误，就是鼓吹“多快好省”四个字：需求越多越好，工期越短越好，质量越高越好，投入越少越好，这是用户最常用的错误口号。

正视这四个要素之间的平衡关系是软件用户、开发商、代理商成熟理智的表现，否则系统的成功就失去了一块最坚实的理念基础。

企业实施 IT 系统的首要目标是要成功，而不是失败，企业可以容忍小的成功，但不一定容忍小的失败，所以需要真正理解上述四个要素的平衡关系，建立并遵循一套软件开发规范，确保项目成功。

2) 高效原则。在需求、资源、工期、质量四个要素中，很多的项目决策者是将进度放在首位的，现在市场的竞争越来越激烈，软件开发越来越追求开发效率，大家从技术、工具、管理上寻求更多更好的解决之道。

3) 分解原则。“化繁为简，各个击破”是自古以来解决复杂问题的不二法门，对于软件项目来讲，可以将大的项目划分成几个小项目来做，将周期长的项目划分成几个明确的阶段。

项目越大，对项目组的管理人员、开发人员的要求越高，参与的人员越多，需要沟通协调的渠道越多，周期越长，开发人员也容易疲劳，将大项目拆分成几个小项目，可以降低对项目管理人员的要求，减少项目的管理风险，而且能够充分地将项目管理的权力下放，充分调动人员的积极性，目标会比较具体明确，易于取得阶段性的成果，使开发人员有成就感。

4) 实时控制原则。在一家大型的软件公司中，有一位很有个性的项目经理，该项目经理很少谈起什么管理理论，也未见其有什么明显的管理措施，但是他连续做成多个规模很大的软件项目，而且应用效果很好。大家一直很奇怪他为什么能做的如此成功，经过仔细观察，终于发现他的管理可以用“紧盯”二字来概括，即每天他都要仔细检查项目组每个成员的工作，从软件演示到内部的处理逻辑、数据结构等，一丝不苟，如果有问题，改不完是不能去休息的。正是在他这种简单的措施下，支撑他完成了许多大的项目，当然他也是相当的辛苦。我们并非要推崇这种做法，这种措施也有他的问题，但是，这种实践却说明了一个很朴实的道理：如果你没有更好的办法，就要辛苦一点，实时控制项目的进展，要将项目的进展情况完全的实时的置于你的控制之下。

上述的方法中对项目经理的个人能力、牺牲精神要求是很高，我们需要有一种进行实时控制项目进度的机制，依靠一套规范的过程来保证实时监控项目的进度。如在微软的管理策略中强调的每日构

建，这确实是一种不错的方法，即每天要进行一次系统的编译链接，通过编译链接来检查进度，检查接口，发现进展中的问题，大家互相鼓励，互相监督。

实时控制确保项目经理能够及时发现问题、解决问题，保证项目具有很高的可见度，保证项目的正常进展。

5) 分类管理原则。由于项目具有很多的特殊性，对于不同的软件项目其项目目标差别很大，项目规模不同，应用领域不同，采用的技术路线差别也很大，因而，针对每个项目的特点，不同的组织应该针对自己的特点实施相应的策略，其管理的方法、管理的侧重点应该是不同的。需要“因材施教”、“对症下药”。对于小项目不能像管理大项目那样去做，对于产品开发类的项目，也不可能像管理系统集成类的项目那样去做，项目经理需要根据项目的特点，制订不同的项目管理的方针政策。

6) 简单有效原则。项目经理在进行项目管理的过程中，不要花哨的功能，不要让多余的功能以花哨的面貌出现在项目中。往往得到开发人员这样的抱怨：“太麻烦了，浪费时间，没有用处”，这是很普遍的一种现象。当然这样的抱怨要从两个方面来分析，一方面开发人员本身可能存在不理解，或者逆反心理的情况，另一方面，项目经理也要反思：我所采取的管理措施是否简单有效？搞管理不是搞学术研究，没有完美的管理，只有有效的管理，不要确定不合理的目标。而项目经理往往试图堵住所有的漏洞，解决所有的问题，恰恰是这种理想，会使项目的管理陷入一个误区，作茧自缚，最后无法实施有效的管理，导致项目的失败。

7) 规模控制原则。该原则是和上面提到的其他原则相配合使用的，即要控制项目组的规模，不要人数太多，人数多了，进行沟通的渠道就多了，管理的复杂度就高了，对项目经理的要求也就高了。可以以少数资深人员开始项目。在微软的 MSF 中，有一个很明确的原则就是要控制项目组的人数不要超过 10 人，当然这不是绝对的，也和项目经理的水平有很大关系。但是人员“贵精而不贵多”，这是一个基本的原则，这和我们上面提到的高效原则、分解原则是相辅相成的。

总之，软件项目管理难做是客观的事实，项目经理除了扎实的理论技术，还有对组织关系的敏感程度如何，也是成败的一个重要原因。衷心希望项目经理在遇到难题的时候，能够冷静分析问题的症结，挖掘深层次的原因，解决根本问题。

## 13.4 校务通系统案例分析

### 校务通系统的项目总结

#### (一) 项目总体信息

项目总时间、总成本、总人力，总规模等信息如下：

项目总时间：2003/4/10 – 2003/6/11，共计 45 天，比计划多 3 天。

项目总成本：¥85 528.00

项目总人力：5 人

项目总规模：157.80 人天

规模比例见下图：



**(二) 项目评审记录**

总评审次数：23

其中：

项目计划评审：1

设计评审：2

质量评审：2

定期评审：8

阶段评审：8

事件评审：2

**(三) 产品提交表**

产品名称	版本	阶段	日期	提交人
任务单	1.0	项目规划	2003/4/10	韩万江
项目计划	1.0	项目规划	2003/4/11	韩万江
配置管理计划	1.0	项目规划	2003/4/11	岳好
质量保证计划	1.0	项目规划	2003/4/11	郭天奇
用户界面 demo	1.0	需求开发	2003/4/18	孙泉
需求规格说明书	1.0	需求开发	2003/4/18	姜岳尊
设计说明书	1.0	设计	2003/4/21	姜岳尊
设计说明书	1.1	设计	2003/4/23	姜岳尊
通用功能 - 增量 1	0.1	实施	2003/4/25	姜岳尊
招生管理 - 增量 2	0.1	实施	2003/5/2	姜岳尊
项目计划	2.0	实施	2003/5/2	韩万江
学生日常管理 - 增量 3	0.1	实施	2003/5/7	姜岳尊
教务管理 - 增量 4	0.1	实施	2003/5/20	姜岳尊
教师辅助功能 - 增量 5	0.1	实施	2003/5/22	姜岳尊
聊天室/论坛 - 增量 6	0.1	实施	2003/6/2	姜岳尊
产品包	1.0	实施	2003/6/6	姜岳尊
测试报告	1.0	测试	2003/6/6	岳好
用户手册	1.0	测试	2003/6/11	孙泉
验收报告	1.0	提交	2003/6/11	姜岳尊
项目总结	1.0	提交	2003/6/12	韩万江

**(四) 实际与计划的差异分析****项目时间差异 (天)**

计划工期	实际工期	时间差异
42	45	-3

**项目总规模差异表 (人天)**

计划规模	实际规模	规模差异
138	157.8	-19.8

**项目成本差异表 (元)**

预算成本	计划 总成本	实际成本	预算成本 差异	总成本 差异
75160	90660	85528	-10368	5132

从项目规模、成本差异表看，尽管略有差异，但基本控制在计划以内。

#### (五) 结论

##### 1) 项目管理的评估总结和建议：

- 基本遵循企业的质量体系实施项目管理过程。
- 根据项目的具体情况，对企业的质量体系的执行活动进行了定制。
- 由于大家对项目管理的认识不同，项目管理的磨合时间较长。
- 建议：大家对项目管理过程应该有统一认识；力争有比较客观准确的项目计划做基础，才可更好地实现项目跟踪管理。
- 建议：项目计划前期应增加一个过程：项目计划的规划过程，对项目计划过程做一个时间规划和任务规划。
- 建议：项目计划期间，管理，开发，质量保证三方应相互明确各自任务的内容和相互的职责，以提高项目计划的准确性和透明度，为项目实施过程的相互协作打下基础。

##### 2) 质量保证的评估总结和建议：

- 质量保证在项目中，基本按计划进行；达到了预期的效果，尤其对关键的质量控制。
- 系统测试阶段质量保证人员的参与，对产品的验错起到很好的作用。
- 建议：以后的功能测试应增加质量保证人员。

##### 3) 开发技术的评估总结和建议：

- 开发人员具有一定的敬业精神和实施能力
- 开发人员对项目计划的时间概念不强
- 建议：增强项目计划的时间观念。

## 13.5 小结

俗话说“编筐编篓，全在收口”，项目结束尽管常常不被重视，但也是非常重要的过程。本章强调了项目结束应该执行的过程，包括合同结束和项目结束。合同结束包括甲方合同结束和乙方合同结束。项目结束过程包括编制结束计划、进行收尾工作、最后评审、编写项目总结报告等。通过完成一个项目，应该能够总结出很多的经验和教训，它为你日后做一个更合格的项目经理提供宝贵的财富。

## 13.6 练习题

### 一、选择题

1. 在项目的末期，与乙方的合同还有尚未解决的索赔，项目经理（ ）。  
A. 可能将合同收尾工作转交给合同管理人员  
B. 通过审计来澄清索赔原因  
C. 知道合同不能收尾  
D. 进行合同收尾，合同收尾之后，可能采取法律行动
2. 合同已经结束的正式通知应该由（ ）提出。  
A. 项目经理  
B. 合同管理负责人  
C. 项目发起人  
D. 项目经理或者合同管理负责人
3. 项目的最后评审是（ ）。  
A. 根据项目基准，确定完成项目需要多少资源  
B. 基于过去的绩效，调整进度和成本基准  
C. 得到客户对项目可以交付成果的验收  
D. 对项目进行全面的评价和审核
4. 项目接近结束的时候，如果客户希望对项目范围进行大的变更，项目经理应该如何做？（ ）  
A. 进行变更  
B. 告诉客户变更带来的影响  
C. 拒绝变更  
D. 向SCCB报告

5. 在客户接受项目的交付结果之前，项目经理应该做什么？（ ）
- A. 检查交付结果的质量                              B. 检查交付结果是否有遗漏  
C. 开具项目完成的发票                              D. 给项目成员分配新的项目

二、判断题

1. 项目的最后评审是不必要的。（ ）
2. 项目成功完成了，才说明项目结束了。（ ）
3. 项目的最后验收过程是甲方对乙方交付的产品或服务进行验收检验，以保证它满足合同条款的要求。（ ）
4. 项目的经验教训总结也是项目结束的一个重要输出。（ ）

## 第五篇

# 实验与实践

本书的最后篇——第五篇讲述课程实验和项目实践，第14章介绍项目管理软件，第15章讲述项目实验步骤，第16章描述项目实践过程。

# 第14章

## ■ 项目管理软件

项目管理的复杂性要求有合适的工具辅助项目管理人员进行项目管理工作。如果用得适当可以大大提高项目管理的效率，项目管理软件有助于制定项目计划和跟踪项目计划的执行。人工维护项目管理数据，通常是很麻烦的。利用项目管理软件，数据资源的任何更改都会自动反应到网络图表、成本表以及资源图表等这些项目文件中。无论你的计划做得多么完善，软件开发过程中都可能会发生一些变化。对于大型项目，人工绘制网络图、计算起止时间、资源使用情况等是非常困难的。因此，项目管理软件的帮助是必要的。

有了项目管理软件，你就不用手工进行繁杂的计算工作了，也不用在活动历时估算或依赖关系发生变化的情况下进行假定方案分析。应用项目管理软件的一个主要益处是可以大大提高效率和精确性。

通常，项目管理软件具有预算、成本控制、计算进度计划、分配资源、分发项目信息、项目数据的转入和转出、处理多个项目和子项目、制作报表、创建工作分解结构、计划跟踪等功能。这些工具可以帮助项目管理者完成很多工作，是项目经理的得力助手。

### 14.1 项目管理软件综述

项目管理软件可以收集、综合和分析项目管理过程的数据，通常用来支持项目从启动到收尾的各个方面，是帮助项目进行范围管理、时间管理、成本管理、风险管理等综合功能的管理信息系统。项目管理软件一般包含两块核心的功能——计划和控制。计划系统主要围绕范围、时间、成本三大目标，辅助完成项目计划工作，如任务分解结构（WBS）、进度计划（网络图、甘特图）绘制、关键路径的分析、成本计划等。控制系统主要提供一些控制手段，以协调项目组织的各种要素。

实践证明，透明的项目进展状态有利于避免工程的延迟，超过预算或者质量低下。项目管理软件的效率和成本将直接影响项目管理其他环节的工作效率、质量和成本。软件企业必须结合企业实际，按照项目管理的方法，结合项目管理软件的特点，让项目管理软件为项目管理更好地服务，加强项目控制与项目组建的信息沟通。

显然，项目管理软件在处理大型项目的某些方面（特别是数据方面）确实要比人工简便得多。对于只有少数活动的短期项目，人工方法也许行得通，但如果项目有数以千计的活动、上千种资源、持续几年时间，项目管理软件对如此复杂工作的协助作用就显得必不可少了。

#### 14.1.1 项目管理软件主要功能

应用得当的高效的项目管理软件不仅可以方便项目管理，还有助于统一开发过程和开发规范，方便项目组成员沟通，节省开发时间，提高软件开发的质量。

项目管理软件能很好地保持记录。例如，可以保持有关各个团队成员的进度计划、各项任务及所用资源的数据资料。在准备优质的报表或将来做计划时，会用到这些数据资料。但是，用户必须不断更新文件，使记录保持不失去意义。收集好相关数据资料并且输入后，软件就能非常迅速地进行每一种计算。通常，项目管理软件具有如下功能。

##### 1. 日程表

日程表程序主要用来对项目中各个单项资源或一组资源确定工作时间。可以用这些日程表计算出项目的进度计划。大部分系统软件都对基本工作时间设置一个默认值，比如星期一到星期五，早上8点到下午5点，中间有一小时的午餐时间。对于各个单项资源或一组资源，可以修改此日程表。例如：修改上、下班时间，按非工作时间输入公司假期，输入各种换班（白天、夜晚），包括节假日以及数

量单位（小时、天、周）。汇报工作进程时要用到这些日程表，它通常可以根据每个单项资源按天、周或月打印出来，或者将整个项目的日程打印成一份全面的项目日程表。

## 2. 预算及成本控制

大部分项目管理软件系统都可以用来获得项目中各项活动、资源的有关情况。人员的工资可以按小时、加班或一次性来计算，也可以具体明确到期支付日；对于原材料，可以确定一次性或持续成本；对各种材料，可以设立相应的会计和预算代码。另外，还可以利用用户自定义公式来运行成本函数。大部分软件程序都应用这一信息来帮助计算项目成本，在项目过程中跟踪费用。项目进行过程中，随时可以就单个资源、团队资源或整个项目的实际成本与预算成本进行对比分析，在计划和汇报工作中都要用到这一信息。大多数软件程序可以随时显示并打印出每项任务、每种资源（人员、机器等）或整个项目的费用情况。

## 3. 图形

对于有大量活动的项目，人工制出一份甘特图或网络图，或人工进行修改图是一件极其乏味而又容易出错的工作。当前项目管理软件的一个最突出的特点是能在最新数据资料的基础上简便、迅速地制作各种图表，包括甘特图及网络图。有了基准计划后，任何修改就可以轻易地输入到系统中，图表自动会反映出这些改变。项目管理软件可以将甘特图中的任务连接起来，显示出工作流程。特别是用户可以仅用一个命令就在甘特图和网络图之间来回转换显示。另外，图形和表格通常有以下功能供用户使用：

- 1) 进行任务和关系的交互式操作处理。例如，通过图表连接任务，改变优先关系或通过扩展活动持续显示功能来改变活动持续时间。
- 2) 定制格式，例如图形大小、标题、颜色、字型以及文件布局。
- 3) 显示任务或成本的基本对照表。
- 4) 突出关键路径，显示出任何活动的延缓。
- 5) 放大或缩小显示图像。

## 4. 转入/转出资料

许多项目管理软件包允许用户从其他应用程序，比如文字处理、电子表格以及数据库程序中获得信息。为项目管理软件输入信息的过程叫做转入。例如，你只需在需要时转入某个电子表格的信息就可以了，而大可不必将电子表格中有关人员或机器的成本信息重新键入项目管理软件程序，而且这样还可能会输入相互冲突或错误的资料。同样地，常常也要把你的项目管理软件的一些信息输入到这些应用程序中去。发出信息的过程叫做转出。例如，可能要把对某一承包商的进度计划报告转出到文字处理备注文件中。

绝大部分项目管理软件程序允许把标准 ASC II 码中的信息文件从 Windows Clipboard 转出到 SQL Database, Lotus, Excel, Microsoft Project Exchange, OLE Client/Server, DDE Client/Server 以及一些其他系统中。

## 5. 处理多个项目及子项目

有些项目规模很大，需要分成较小的任务集合或子项目。另一种情况是经验丰富的项目经理同时管理好几个项目，而且，团队成员也同时为多个项目工作，在多个项目中分派工作时间。在这种情况下，大部分项目管理软件程序能提供帮助。它们通常可以将多个项目储存在不同文件里，这些文件相互连接。项目管理软件也能在同一个文件中储存多个项目，同时处理几百个甚至几千个项目，并绘制出甘特图和网络图。

## 6. 制作报表

绝大多数项目管理软件包都有非常广泛的报表功能。项目管理软件包在最初应用时，一般只有少数报表，通常是列表总结进度计划、资源或预算。下面列出一些可制作的报表内容：

- 1) 项目全面汇报报表。
- 2) 项目主要阶段（里程碑）报表。
- 3) 一定时期内的各种信息，例如这一时期内完成和正在进行的任务，以及准备开始的任务。

4) 财务报表，显示全面的信息，包括所有任务、整个项目预算、超支情况、累计预算成本、实际成本以及承付款项。

5) 成本/进度控制系统准则报表。

6) 对项目的各种或一组资源进行资源配置报表。

7) 定制的标准报表、交叉报表、变量基准对比表。

## 7. 资源管理

目前的项目管理软件都有一份资源清单，列明各种资源的名称、资源可以利用时间的极限、资源标准及过时率、资源的收益方法和文本说明。每种资源都可以配以一个代码和一份成员个人的计划日程表。对每种资源加以约束，比如它可被利用的时间数量。用户可以按百分比分为任务配置资源，设定资源配置的优先标准，为同一任务分配各个资源，并保持对每项资源的备注和说明。系统能突出显示并帮助修正不合理配置，调整和修匀资源配置。大部分软件包可以为项目处理数以千计的资源。

## 8. 进度安排、计划

在所有项目管理软件包中，用户都能界定需要进行的活动。正如软件通常能维护资源清单，它也能维护一个活动或任务清单。用户对每项任务选取一个标题、起始与结束日期、总结评价，以及预计工期（包括按各种计时标准的乐观、最可能及悲观估计），明确与其他任务的先后顺序关系以及负责人。通常，项目管理软件中的项目会有很多相关任务。另外，大部分程序可以创建工作分解结构，协助进行计划工作。

在实际工作中，项目规模往往比较大，人工进行进度安排就显得极为复杂了。项目管理软件包能为进度安排工作提供广泛的支持，而且一般是自动化的。大部分系统能根据任务和资源清单以及所有相关信息制作甘特图及网络图，对于这些清单的任何变化，进度安排会自动反映出来。此外，用户还能调度重复任务，制定进度安排任务的优先顺序，进行反向进度安排（从末期到日首期），确定工作轮班，调度占用时间，调度任务，确定最晚开始或尽早开始时间，明确任务必须开始或必须结束日期，或者是最早、最晚日期。

## 9. 排序及筛选

利用排序，用户可以按随心所欲的顺序来浏览信息，比如从高到低的工资率，按字母顺序的资源名称或任务名称。大部分程序有各种排序方式。筛选功能帮助用户选择出符合具体准则的一些资源。例如，某些任务要用到某种具体资源，用户如果想了解这些任务的有关信息，只需命令软件程序忽略未使用这种资源的任务，而只把用到这种资源的任务显示出来就可以了。

## 10. 项目监督及跟踪

项目管理的一项基本工作是对项目进度、实际费用和实际资源耗用进行跟踪管理。大部分项目管理软件包允许用户确定一个基准计划，并就实际进程及成本与基准计划里的相应部分进行比较。大部分系统能跟踪许多活动，如进行中或已完成的任务、相关的费用、所用的时间、起止日期、实际投入或花费的资金、耗用的资源，以及剩余的工期、资源和费用。

## 11. 假设分析

项目管理软件一个非常实用的特点是进行假设分析。用户可以利用这一特点来探讨各种情形的效果。在某一项目的一些节点上，用户可以向系统询问：“如果拖延一周，会有什么结果？”系统会自动计算出延迟对整个项目的影响，并显示出结果。几乎可以对项目中所有变量（人员、工资率、成本）进行测试，观察发生具体情况的影响。这种分析能使项目经理更好地控制有关项目的各种风险。

### 14.1.2 项目管理软件介绍

根据功能和价格，可以将项目管理软件划分为高端和低端。高端项目管理软件的功能强大，但是价格不菲，是供专业项目管理人士使用的软件，例如，IBM 公司的 RPM、Primavera 公司的 P3、Welcome Software Technology 公司的 Open Plan、Lucas Management Systems 公司的 Artemis、北京梦龙公司的智能 PERT 系统等。这些高端项目管理软件能在一个分布式数据环境下共享资料，便于处理多任务项

目，可以连接其他商业信息系统。低端项目管理软件是通用的项目管理工具，例如 Microsoft 公司的 Project、TimeLine 公司的 TimeLine、Scitor 的 Project Scheduler 等，其功能虽然不是很强大，但是价格比较便宜，可以用于一些中小型项目。

下面就一些比较有名的、通用的项目管理软件进行介绍，简要说明了它们的特征。项目经理可以根据自身需求去分析和选择。

### 1. Primavera Project Planner

美国 Primavera 公司是一家专业从事项目管理软件开发与服务的公司，成立于 1983 年。也就是在 1983 年，该公司推出了日后成为项目管理软件领头羊的 Primavera Project Planner（简称 P3），它是高端的项目管理产品。

P3 首先是基于广义网络计划技术的理论编制的项目管理软件。P3 采用目标管理的模式对项目实施控制。它将优化后的计划作为目标计划进行保存，随时可调出来与当前的进度和资源消耗进行比较，可以方便地发现哪些作业超前，哪些作业落后，对整个工期有没有影响。这样，对工程的按期完工很有帮助。

P3 能够根据项目的任务分解结构（WBS）将项目的工作范围从大到小进行分解，直至可操作的工作单元，也可以将组织机构逐级进行分解（OBS），形成最基层的组织单元，并将每一工作单元落实到相应的组织单元去完成。然后 P3 根据不同管理层的要求，在任务分解结构或组织分解结构的任意层次上进行统计和汇总。除此之外，P3 还可以根据工程的属性任意对工作进行筛选、分组、排序、汇总。例如，我们可以将一个项目的作业分成 3 级网络，并将所有 1 级网络作业的网络级别设为 1。我们需要看 1 级网络时，只要选择网络级别等于 1 的作业即可。这样可以帮助项目经理在最短的时间内调出自己所需要的数据。

作为商品化的软件，P3 的数据接口功能齐全。既可以输出到传统的 dBase 数据库、Lotus 文件和 ASCII 格式文件，也可以接收 dBase、Lotus 格式的数据，还可以通过 ODBC 与 Windows 程序进行数据交换。使用 P3 的批处理程序经简单编程就可以执行 P3 的大部分功能。此外 P3 还提供了开发引擎 RA，编程人员使用其他编程工具如 Visual、Basic、Visual C ++、PowerBuilder 通过 RA 来读写 P3 数据。Primavera 还提供与 Oracle 数据库的双向接口 DataStore。

P3 还提供 Primavera Postoffice 邮局软件，项目施工人员可以使用该邮局软件打开总部的工作安排，并将实际进展反馈给总部。Primavera 还提供了 Webster for Primavera，使用该软件的各单位和个人可通过浏览器来访问和更新项目数据。

Primavera 公司开发的 P3 软件的市场占有率是比较高的，目前国内的大型、特大型项目建设几乎都采用了该软件进行工程的项目管理。该软件系列具有通过各种视图，表格同时管理多个项目，通过开放的数据库与其他系统相结合进行分析和决策，能自动生成百余种标准报告进行文档管理等功能，并能通过电子邮件传递信息，进行交流。

### 2. Sure Trak Project Manager

Sure Trak Project Manager 也是 Primavera Systems 公司的产品。是 Primavera 专门针对中小型的工程项目做的项目管理软件，是简化的 P3，采用国际标准的项目管理工具，是一个高度视觉导向的程序，具有优异的放缩、压缩及拖入功能。可以很方便的组织丰富的视图与报表，快速地进行工程的进度计算等工程方面的操作。SureTrak Project Manager 被称为 30 分钟的项目组织者。它的基本结构，比如柱形、图表、色彩和数据结构便于调整，定制模板也容易创建。它的任务分解结构功能优异，便于使用。重复活动处理简便，活动网络图可以分区段储存在磁盘里，并可装入其他程序。

### 3. IBM Rational Portfolio Manager

IBM Rational Portfolio Manager（简称 RPM）是 IBM 收购的一个专业项目管理工具（原先叫做 PM OFFICE），并入 IBM RATIONAL 产品线。RPM 主要的市场领域是 IT 项目管理。IBM RPM 产品化程度非常高，可配置性非常强。由于内部集成了工作流和报表工具，具备很好的定制性。IBM RPM 提供了非常精细化和自动化的管理。如对单个任务进行费用管理，并且能够计算出净值。交付物的权限定义

和控制也非常复杂和精确。IBM RPM 内嵌了一些工具，如报表设计器等，分析展现功能强大。

IBM RPM 优点是产品化程度更高、可配置性非常强、精细化程度高、分析功能强大。缺点是过于注重精细化和自动化，是一个理想管理环境的解决方案，不支持软件估算，价格也非常昂贵。

#### 4. SuperProject

Computer Associates International 公司的 CA-SuperProject 是一个常用的软件，特别是在那些管理公司网络的项目管理人员、在 Unix 或 Windows 环境下的工作人员以及需要高性能程序的人中更受欢迎。这个软件包能支持多达 160 000 多个任务的大型项目。许多评论人员因为它在大型项目及小型项目两方面的优异表现而予以高度评价。CA-SuperProject 能创建及合并多个项目文件，为网络工作者提供多层密码入口，进行 PERT 的概率分析。而且，这一程序包含一个资源平衡算法，在必要时，可以保证重要工作的优先性。它的主要缺点是用户界面不如其他一些程序友好，应当在将来的版本中进一步解决。

CA 公司的项目管理套件：Clarity Portfolio Manager 用于战略管理；Clarity Resource Manager 用于资源管理；Clarity Project Manager 用于项目日常工作管理；Clarity Financial Manager 用于项目财务管理；Clarity Process Manager 用于项目流程管理。它们连续三年在 Gartner 欧洲峰会上荣获“最佳解决方案”和“最佳供应商”殊荣。

#### 5. Artemis Views

Artemis Views 是美国 Artemis 公司推出的企业级项目管理工具。Artemis 公司是全球比较大的专业项目管理工具及服务商。客户包括美国国防部空军基地、英国 Orange 电信、意大利电信、GE、Honda、伦敦运输、纽约州交通部、Arizona 公共服务部、瑞士邮政电话电报公司、日本 Kawasaki 重工等 3000 余家。在全球 27 个国家和地区有 40 个分支机构。

Artemis Views 包括四个模块：Project View、Cost View、Global View、Track View；每一种模块分别针对不同的用户对象。Client/Server 模式，支持数据库包括：Oracle、MS SQL、Sybase 等，服务器平台支持：Windows NT、HP\_UX、Sun Solaris。

Artemis Views 主要功能包括：支持层次结构的的多计划视图；分析多项目计划的成本和资源的需求；可以直接将 MS Project 的数据存到中央数据库；允许 MS Project 的数据进入跟踪模块，来实现活动和时间的自动跟踪；支持 Web\_based 的用户离线填报工时，连上服务器后自动更新数据库的数据；企业级成本计划和控制；提供项目进度、活动和资源的财务角度视图；支持在线成本数据和差异分析；Earned\_value 的项目控制和汇报；支持 ERP 的集成；为不同权限的用户提供不同的使用模块。

#### 6. Microsoft Project

作为微软家庭一员的 Microsoft Project，已经占领了项目管理软件包市场的大量份额。Microsoft Project 是一个易于使用、特性齐全、获奖较多的项目管理软件包。Microsoft Project 是目前在世界范围内应用较为广泛的，以进度计划为核心的项目管理软件，其同样是把计算机网络信息管理的强大功能与工程项目管理相结合的产物，能实现远距离的管理与控制。

它是一个强有力的计划、分析和管理工具，能够让你创建企业范围对具体任务要求较高的项目管理解决方案。该程序通过把一个项目分解为易于管理的步骤，能够对最复杂的计划进行可视化分析，可以让你看到任务是如何相互联系的，这对于你制定全面的计划非常关键。同时可以找到你的瓶颈所在，以及整个项目的未来开销。你也可以将几个项目进行合并，以便对共享资源、团队工作量，以及对你正在同时筹划的多个项目放在一起是否合理进行评估。你甚至可以自动地交流项目的状态。内置的到 Microsoft Exchange 的链接可以让该程序方便地发布一个项目所选定的属性，并且可以连接到 Microsoft Mail、Schedule +、Microsoft Back Office (TM) 或者数以百计的附加程序。Microsoft Project 可以管理费用，可以绘制甘特图、各种统计图形和生成图文并茂的报表。

Microsoft Project 既可供个人使用，也可供团队使用。它的主要优点是帮助用户编制任务进度计划、管理各种资源（人力、设备等）、容易使用。但是 Microsoft Project 几乎可以管理任何行业的项目，这既是优点也是缺点：适用面太广了，就缺乏针对性。

Project 虽然适用性较强，属于项目管理普遍适用的进度管理软件，但其功能不能满足行业项目管

理的实际需要，因此，一般被用来简单的管理进度计划。但由于其简单、方便、廉价，在IT与通信行业里使用的最多的是Project软件，基本能够满足需求。

### 7. Project Scheduler

Scitor公司的Project Scheduler软件是一个易于操作、基于Windows的项目管理软件包，它获得《电脑杂志》的“编辑选择奖”（Editos' Choice Award）。Project Scheduler具备传统项目管理软件的所有特征，图形界面设计完好，报表功能强大，制图方面也是如此。比如甘特图，能用各种颜色把关键任务、正或负的时差、已完成的任务以及正在进行的任务区别开来。任务之间建立图式连接极为方便，任务工时的修改也很容易。资源的优先设置及资源的平衡算法非常实用。对多个项目及大型项目的操作处理也比较简单，与外部数据库的连接也不同一般。

你可在桌面完成基本的工作，或与SQL数据库一起处理大的、复杂的程序。它包括向导、当日窍门、域级帮助等，还具有非常好的灵活性，适合组织、合并及查看项目情况。它还提供一个HTML网页出版程序，使你快速、专业地交流项目的进展。

该软件美中不足的地方是联机帮助和文件编制以及电子邮件功能有限。

### 8. Time Line

《电脑杂志》的“编辑选择奖”的另一位得主是Symantec公司的Time Line软件。尽管这个软件在初学者使用时略微有些困难，但Time Line是有经验的项目经理的首选。它的报表功能以及与SQL数据库的连接功能都很突出。日程表、电子邮件的功能，排序和筛选能力以及多项目处理都是精心设计的。另外，它还有一个叫做Co-Pilot的功能，这是一个很有用的推出式帮助设施，用户界面很好，极易操作。然而，许多评论人员认为Time Line最适于大型项目以及多任务项目，但不如其他软件包便于初学者使用。

### 9. VisualProject

视锐达公司的VisualProject是定位于非工程项目管理的产品，主要应用“软件开发与IT服务”、“新产品研发”等应用领域。VisualProject是一款优秀的专业企业级项目管理软件，覆盖企业从组织级到项目级的项目管理。不同于个人生产率软件，专业企业级项目管理软件是企业管理软件的一个分支。

VP提供了多种颗粒度的管理选择。如报工，可以报工到项目、阶段、任务。费用归集可以到项目，也可以到具体的任务。这些功能有助于与企业管理成熟度匹配，并伴随企业管理成熟度成长。VP有很多功能都全面考虑的国内企业的管理水平和特点，因此适应性比较好。VP支持CMMI，提供了全面的软件项目管理的支持。比如关键的估算环节，VP提供了三种估算方法，包括“规模-生产率”这样的CMMI标准方法。VP还全面支持CMMI中的有关过程域，如“度量分析”等，有效的提升软件项目管理水平，减低管理工作量。IE界面风格，简洁明快，提供二次开发与定制。价格比较低。

VisualProject集成了单机版MS Project作为项目经理在任务管理的客户端工具。

### 10. 梦龙

梦龙公司在管理理论方面始终保持与世界同步并在核心技术方面保持一定的领先，形成了协同工作平台、项目管理平台、招投标等三大系列产品。其中核心产品——项目管理软件又分为公司端与项目端，它是针对公司与项目人员的不同需求而开发的，其中公司端产品主要是面对公司领导，为其进行决策分析使用，项目端产品提供给项目部人员进行日常项目管理。梦龙项目管理软件是建立在公司独立开发的Linkworks平台基础上，基于统一授权管理的基础平台有如下特性：

- 1) 安全性。身份认证、权限管理、加密传输等平台安全方面做了大量的工作。
- 2) 扩展性。系统采用“软件总线”的插槽式结构，符合插件协议的都可以外挂，可以针对用户的不同要求进行插件开发，具有很强的扩展性。

3) 健壮性。平台采用了分服务体系和三层结构，保证了系统健壮性。

4) 可维护性。插槽式结构具有很强的可维护性。此外，平台提供了软件配置功能，保证客户端版本的一致，增加了系统可维护性。

5) 灵活性。平台采用框架模块化设计，系统的各个功能模块都可以在平台框架中独立运行，模块之间通过中央工程数据库进行交互。用户可以根据使用需要在服务端对可用模块进行统一配置，也可以在客户端根据个人使用习惯设置常用模块。

从 2004 年起微软、梦龙两家公司合作共同推出 PM ++ 项目管理平台解决方案，它基于梦龙 Link-Project 与 Microsoft Office Project 进行数据对接而形成的，并命名为“PM ++ 项目管理平台”。它是世界上两大先进项目管理技术的完美结合，是实践项目管理的理想工具。

目前市场上有大量的简单的项目管理软件，也有许多“公开源代码”的项目管理软件。这些软件一般只完成项目管理某一阶段和某一方面如计划安排、人员管理、风险分析等功能。例如 OpenProj 就是开源的项目管理软件，与微软的 Project 类似。用各种图来安排工序，标识关键路径，触发各类事情，工序优化等等。OpenProj 做的较为出色的是它是跨平台的，Windows、Linux、Unix、Mac（这个 Unix 版本较为特殊，一直是专门列出来的）下都能使用。小工程项目可以考虑使用 OpenProj，毕竟可以节省一大笔开支。作为定位为 Project 替代者的产品，其使用方式应于 Project 相似，可以轻松上手。

### 14.1.3 选择项目管理软件的注意事项

企业选择项目管理软件时，应当综合考虑“价格、功能、易用性”，选择购买项目管理软件时可以根据需要决定。注意事项有如下几点。

1) 价格与可利用的功能。一定要考虑系统是否具备项目组织所需要的各种功能，同时价格是可以承受的范围内。例如，软件是否包含任务分解结构以及甘特图和网络图，资源平衡或均衡算法怎么样？系统能否排序和筛选信息、监控预算、生成定制的日程表，并协助进行跟踪和控制？它能否检查出资源配置不当并有助于解决？

2) 容量。容量主要是考虑系统能否处理你预计进行的项目数量、预计需要的资源数以及预计同时管理的项目数量。

3) 操作简易性。操作简易性主要应考虑系统的“观看”和“感觉”效果、菜单结构、可用的快捷键、彩色显示、每次显示的信息容量、数据输入的简易性、现在数据修改的简易性、报表绘制的简易性、打印输出的质量、屏幕显示的一致性，以及熟悉系统操作的难易程度。

4) 与其他系统的兼容能力。在当今的数字化社会里，大量的电子系统日趋统一。如果你的工作环境里，数据储存在各个地方，比如数据库、电子数据表里，这时就要特别注意项目管理软件的兼容统一能力。另外，项目管理软件通过电子信箱向文字处理及图形软件包转入信息的能力也会影响到你的决策。

5) 安装要求。这里主要考虑运行项目管理软件对计算机硬件和软件的要求：存储器、硬盘空间容量、处理速度和能力、图形显示类型、打印设置以及操作系统等。

6) 报表功能。目前各种项目管理软件系统的主要不同之处是它们提供的报表种类和数量。有些系统仅有基本的计划、进度计划和成本报表，而有一些则有广泛的设置，对各个任务、资源、实际成本、承付款项、工作进程以及其他一些内容提供报表。另外，有些系统更便于定制化。报表功能应给予高度的重视，因为大多数用户非常注重软件这种能生成内容广泛、有说服力的报表的功能。

7) 安全性能。有些项目管理软件有相对更好的安全性。如果安全问题很重要，那么就要特别注意对项目管理软件、每个项目文件及每个文件数据资料的限制访问方式。

8) 经销商的支持。要特别注意，经销商或零售商是否提供技术支持、支持的费用，以及经销商的信誉。

对于管理大型的或企业规模型的项目。需应用高端项目管理软件，他们能在分布式数据环境

下共享资料，便于处理多任务项目，连接其他商业信息系统，比如会计、采购、而且通常对进度计划及资源配置有非常完善的算法。对于一般的中小型企业软件项目，通用的项目管理工具更为实用。下面详细介绍 Microsoft Project 和 IBM Rational Portfolio Manager。

## 14.2 Microsoft Project

Microsoft Project 是由微软开发的一个享有盛誉的通用项目管理软件产品，目前占领主要的市场份额。是创建和管理项目的工具，数据库中存储有关项目的详细数据，可以利用这些信息，计算和维护项目日程、成本、以及其他要素，创建计划、跟踪项目。Microsoft Project 不仅能够快速、准确地建立项目计划，使项目管理者从大量繁琐的工作中解脱出来，而且可以帮助项目经理实现进度控制、成本分析、预测、控制等人工无法实现的功能，使项目工期大大缩短，资源得到有效利用，提高经济效益。Microsoft Project 将可用性、强大的功能和灵活性完美地融合在一起。

Microsoft Project 可以帮助用户编制任务进度计划、管理各种资源（人力、设备等）、管理费用；可以绘制甘特图、各种统计图形、生成图文并茂的报表。该软件非常容易使用。通过与熟悉的 Microsoft Office 程序、强大的报表、引导的计划以及灵活的工具进行集成，您可以对所有信息了如指掌，控制项目的工时、日程和财务，与项目工作组保持密切合作，同时提高工作效率。Microsoft Project 具有如下特点：

1) 有效地管理和了解项目日程。使用 Microsoft Project 设置对项目工作组、管理和客户的现实期望，以制定日程、分配资源和管理预算。通过各种功能了解日程，这些功能包括用于追溯问题根源的“任务驱动因素”、用于测试方案的“多级撤消”以及用于自动为受更改影响的任务添加底纹的“可视化单元格突出显示”。

2) 快速提高工作效率。项目向导是一种逐步交互式计划辅助工具，可以帮助您快速掌握项目管理流程。该工具可以根据不同的用途进行自定义，它能够引导您完成创建项目、分配任务和资源、跟踪和分析数据以及报告结果等操作。直观的工具栏、菜单和其他功能使您可以快速掌握项目管理的基本知识。

3) 利用现有数据。Microsoft Project 可以与其他 Microsoft Office 系列程序顺利集成。通过将 Microsoft Office Excel 和 Microsoft Office Outlook 中的现有任务列表转换到项目计划中，只需几次键击操作即可创建项目。可以将资源从 Microsoft Active Directory 或 Microsoft Exchange Server 通讯簿添加到项目中。

4) 构建专业的图表和图示。“可视报表”引擎可以基于 Project 数据生成 Visio 图表和 Excel 图表的模板，您可以使用该引擎通过专业的报表和图表来分析和报告 Project 数据。您可以与其他用户共享您创建的模板，也可以从可自定义的现成报表模板列表中进行选择。

5) 有效地交流信息。根据负责人的需要，轻松地以各种格式显示信息。您可以设置一页日程或其他报表的格式并进行打印。使用“将图片复制到 Office 向导”可以顺畅地将 Project 数据导出到 Microsoft Office Word 中以用于正式文档，导出到 Microsoft Excel 2007 中以用于自定义图表或电子表格，或者导出到 Microsoft Office PowerPoint 中以用于清晰演示文稿。

6) 进一步控制资源和财务。使用 Microsoft Project，您可以轻松地为任务分配资源，还可以调整资源的分配情况以解决分配冲突。通过为项目和计划分配预算，可控制财务状况。通过“成本资源”，可改进成本估算。

7) 快速访问所需信息。可以按任何预定义字段或自定义字段对 Project 数据进行分组。这样会合并数据，使您可以快速查找和分析特定信息，从而节约了时间。可以轻松标识项目不同版本之间的更改，因此您可以有效地跟踪日程和范围的更改。

8) 根据需要跟踪项目。可以使用一组丰富的预定义或自定义衡量标准来帮助您跟踪所需的相关数据（完成百分比、预算与实际成本、盈余分析等）。可以通过在基准中保存项目快照来跟踪项目进行期间的项目性能情况。

9) 根据需要自定义。可以专门针对项目调整 Microsoft Project。可以选择与您的项目日程集成的自定义显示字段。还可以修改工具栏、公式、图形指示符和报表。XML、Microsoft Visual Basic for Applications (VBA) 和组件对象模型 (COM) 加载项有助于数据共享和创建自定义解决方案。

10) 在需要时获得帮助。Microsoft Project 为新手和有经验的用户提供了丰富的帮助内容。它包含强大的帮助搜索引擎、智能标记以及向导。进一步增强了对培训课程、模板、文章等的在线访问功能(需要 Internet 连接)。

Microsoft Project 的 Project Server 是运行在服务器上与 Project Standard 和 Project Professional 一起使用的配套软件，可以用来给整个项目团队提供任务汇报、日程更新、每个项目耗时记录等协同工作方式。可以进行有效的工作组协作，项目经理使用 Project Standard 或者 Project Professional 将项目信息发送到 Project Server，其他人员使用 Project Web Access 查看，项目经理也可以通过 Project Web Access 进行信息更新。它的主要优点：

1) 帮助用户编制任务进度计划、管理各种资源(人力、设备等)、管理费用；可以绘制 Gantt 图、各种统计图形、生成图文并茂的报表。

2) 该软件非常容易使用，不需要专门的培训用户自己就能学会。易用性是 Microsoft 软件产品的共性优点。

3) 在国内，广大计算机用户很容易得到该软件，这也是用户量最大的根本原因。

其实，Microsoft Project 与微软其他产品 (Access、Excel、PowerPoint、Word) 很相似，菜单栏几乎一样，用户的工具栏如出一辙。另外，用户可以在应用文件之间轻易地来回移动信息资料，例如，可以把 Excel 资源表中的成本信息轻易地移到 Microsoft Project 中，Microsoft Project 中的一份甘特图也可通过拖入或连接，轻易地移入 Word 文件中。日常用语 (Tips of the day)、提示卡以及大量帮助范例大大简化了程序的应用，交互式日程系统、电子邮件以及分配设备的功能很强大。现在，还包含应用文件的视象 (Visual Basic for Applications)，便于高级用户设计接口或自动处理重复性工作。

Microsoft Project 的缺点是不支持软件项目中的立项与结项、变更管理、需求管理、质量管理、软件配置管理等重要管理工作。并且它也不如其他一些软件包能处理多个项目及子项目。微软公司的 Project 建立在自身的 Windows 2000 Server、Office、Exchange Server、Outlook 等系列产品上，但由于普通用户难以全面部署与掌握而受到影响，同时产品之间的融合不够紧密。

### 14.3 IBM Rational Portfolio Manager

RPM 主要的市场领域是 IT 项目管理，RPM 提供了合同管理等辅助功能。IBM RPM 产品化程度非常高，可配置性非常强，如界面的自定义配置等，由于内部集成了工作流和报表工具，具备很好的定制性。

IBM Rational Portfolio Manager(简称 RPM) 是为企业级别的用户设计的项目管理解决方案，同时也是 IBM 内部进行项目管理的标准化平台，它不但能够为企业的项目管理、大型项目管理和项目组合管理提供很好的平台支持，还具有良好的伸缩能力和灵活性，可以满足不同企业规模、不同项目管理水平与成熟度、不同项目类型、不同项目规模与项目复杂度的需要。

RPM 内置了模板机制，它是 IBM 在 RPM 中支持项目管理实践经验共享和方法重用的关键策略。在 RPM 的流程模板中，不但包含的项目管理过程所需的任务分解结构、过程文档、任务描述及处理时长、任务参与人员及技能要求、计分卡以及常见问题风险等，而且通过 RPM 和 RMC 内置的导入导出功能，可以直接将 RMC 中定制的项目管理方法导入到 RPM 中，生成可执行的项目管理流程模板，使项目管理团队能够在不知不觉中，轻松愉快的按照组织预定义的项目管理方法，完成项目管理工作。这不但从根本上解决了项目管理流程执行力的问题，还为企业提供了良好的知识共享机制，对提高组织项目管理过程的成熟度有很大帮助。

Rational Portfolio Manager 能够完成基本的项目管理，也能够自动管理整个组织的所有项目。它的主要功能包括：

- 1) 快速入口。用户可根据其关注点，自定义快速入口，观察项目状态，了解分配给自己的工作。
- 2) 工作管理。以项目工作为中心，制定项目目标，创建任务分解结构，描述依赖关系，制定计划，分配资源，并跟踪项目进度。
- 3) 范围管理。记录并管理项目需求、变更请求、缺陷、风险和问题等元素。这些元素可能影响到项目的范围。
- 4) 文档管理。对项目中的文档进行管理，提供版本控制功能，并能够将文档与工作节点或范围管理元素对应。
- 5) 资源管理。组织、管理企业内部所有（人力）资源的基本信息，其中包括地理位置、职位、技能等关键信息。
- 6) 工作流设计。以工作管理元素，范围管理元素，或者文档元素为主体，设计工作流并应用于项目进行过程中。

Rational Portfolio Manager 的体系结构可以分为三个层次，数据库、Web 应用程序和用户界面。最底层是 Rational Portfolio Manager 数据库。Rational Portfolio Manager 所有的数据都存储在数据库中。IBM DB2 和 Oracle 是 Rational Portfolio Manager 支持的两种数据库管理系统。

Web 应用程序层在数据库层与用户界面之间。它从用户界面获得用户请求，处理请求之后提交给数据库层，再将数据库返回的数据处理后发送给用户界面。Rational Portfolio Manager 支持 IBM WebSphere Application Server 和 Apache Tomcat Application Server 等。

Rational Portfolio Manager 提供三种用户界面：

- 1) Windows 客户端。只能在 Windows 系统中使用，能够完成所有的 Rational Portfolio Manager 功能。
- 2) 基于 Web 的客户端——TeamMember。支持 IE 和 Firefox。支持 Windows, Linux 等多种平台。只提供基本功能，适合项目组成员使用。
- 3) Web Service API。用户可以根据需求，编写程序，调用 Rational Portfolio Manager Web Service API，与 Rational Portfolio Manager 应用程序交互。

IBM RPM 的优点：

- 1) 产品化程度更高。IBM RPM 产品化程度非常高，可配置性非常强。如界面的自定义配置等。由于内部集成了工作流和报表工具，具备很好的定制性。
- 2) 精细化程度高。IBM RPM 提供了非常精细化和自动化的管理。如对单个任务进行费用管理，并且能够计算出挣值。报工也是非常精确。交付物的权限定义和控制也非常复杂和精确。
- 3) 分析功能强大。IBM RPM 内嵌了一些工具，如报表设计器等，分析展现功能强大。

# 第 15 章

## ■ 课 程 实 验

为了更好地理解和掌握项目管理的知识，项目管理课程需要完成一些实验，或者完成一些项目实践。本章提出课程实验的内容，下章描述项目实践的内容。本课程要求的课程实验是采用一种项目管理软件工具完成项目管理过程。例如，采用 Microsoft Project 或者 IBM Rational Portfolio Manager 软件工具进行实验。

### 15.1 Microsoft Project 实验

本次实验是通过使用 Microsoft Project 辅助完成项目管理的一些工作，目的是了解 Microsoft Project 工具的使用和项目管理的相关知识。实验内容和步骤如下：

#### 15.1.1 建立项目管理文件

在开始制定项目计划之前，要明确定义项目的一些基本属性信息，或者对项目有一个基本的定义，例如项目的名称、内容、开始时间、结束时间等。例如有一个校园网站项目，内容是通过网站介绍学校的基本情况，及时发布一些信息，同时有论坛等讨论区域。下面开始在 Project2003 中创建项目，实验步骤如下。

##### 1. 新建项目

选择 [程序]→Microsoft Office→Microsoft Office Project 进入 Project2003。选择 [文件]→[新建] 菜单命令，打开 Project 的“新建项目”向导，如图 15-1 所示，单击“空白项目”将出现“任务”向导，如图 15-2 所示。

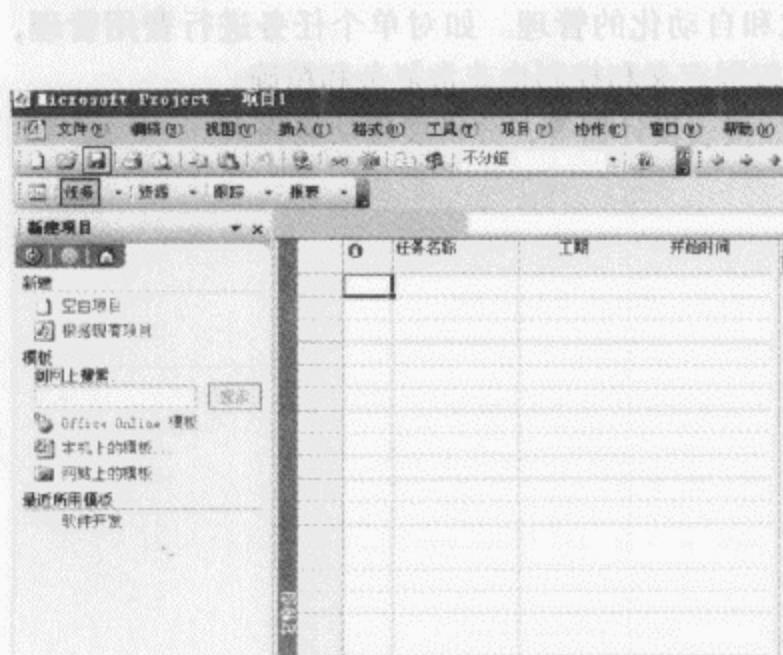


图 15-1 新建项目

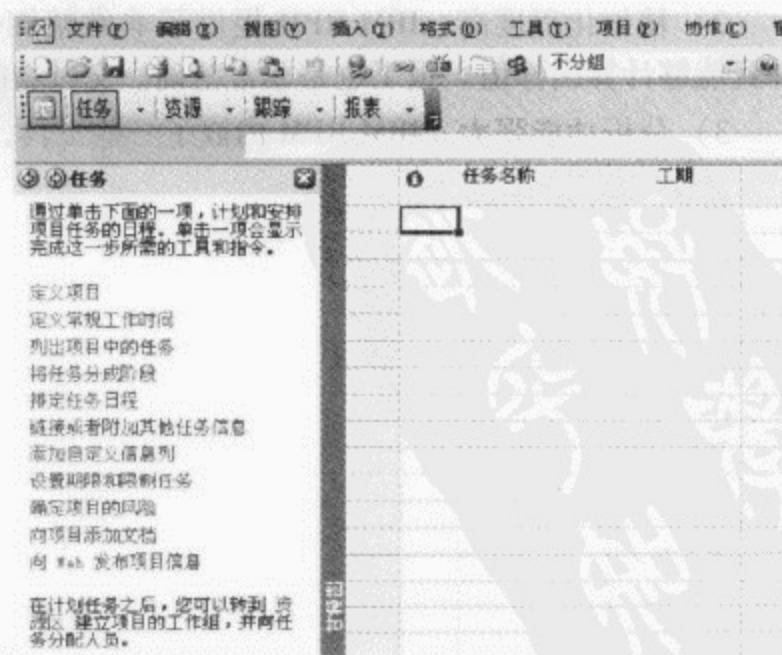


图 15-2 任务向导

##### 2. 定义项目

项目创建完成之后，需要定义项目。单击图 15-2 中的“定义项目”，将显示“定义项目”向导，利用此向导可以很容易地完成定义项目操作。分三步：

- 1) 输入项目的开始时间, 如图 15-3 所示, 将日期调整为项目的预计开始日期。
- 2) 输入项目工作组选项, 如图 15-4 所示, 询问是否使用 Project Server 和 Project Web Access, 本项目比较小, 不需要 Project Server 的协调, 选择“否”。

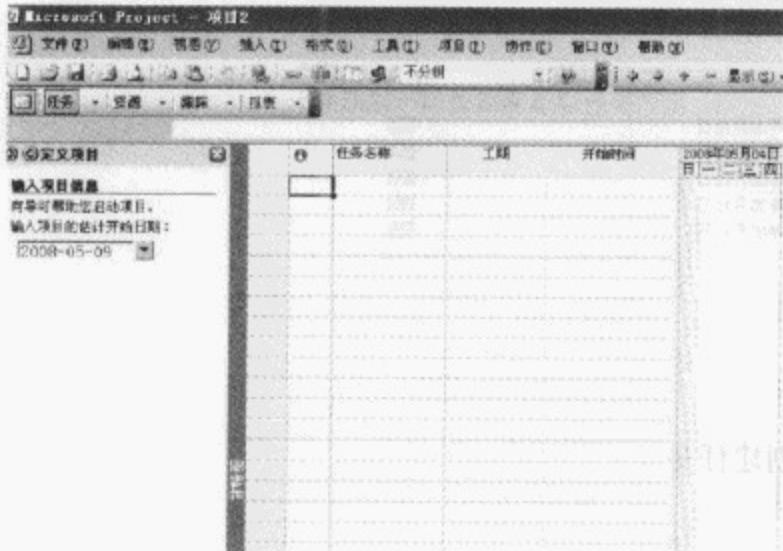


图 15-3 定义项目 - 1

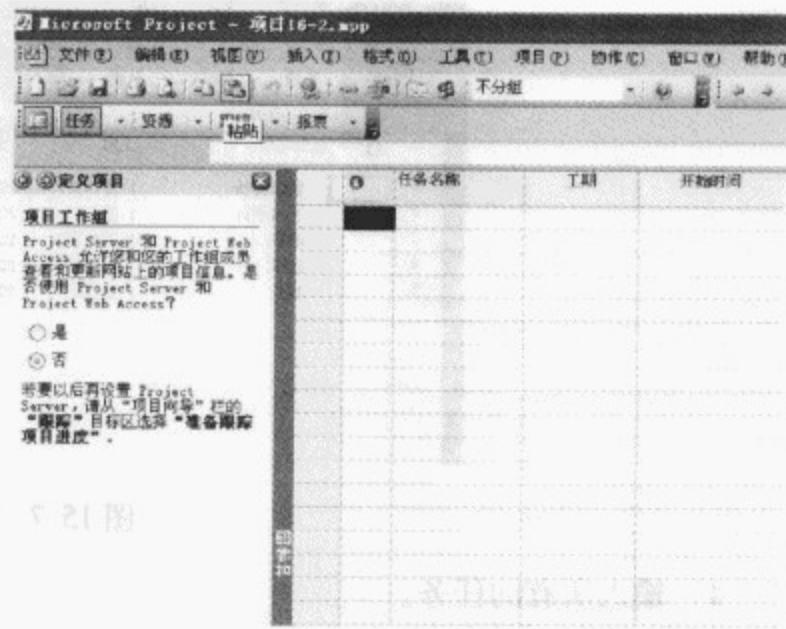


图 15-4 定义项目 - 2

- 3) 保存文件, 将文件保存为 SchoolWebside.mpp。

### 3. 设置环境信息

项目的环境信息包括工期、项目的日历等。选择 [工具]→[选项], 弹出“选项”对话框, 选择其中的“日历”选项卡可以输入、查看或者修改日期、时间等设置, 如图 15-5 所示。选择“日程”选项卡可以输入、查看和修改排定任务日程的首选项, 如图 15-6 所示。

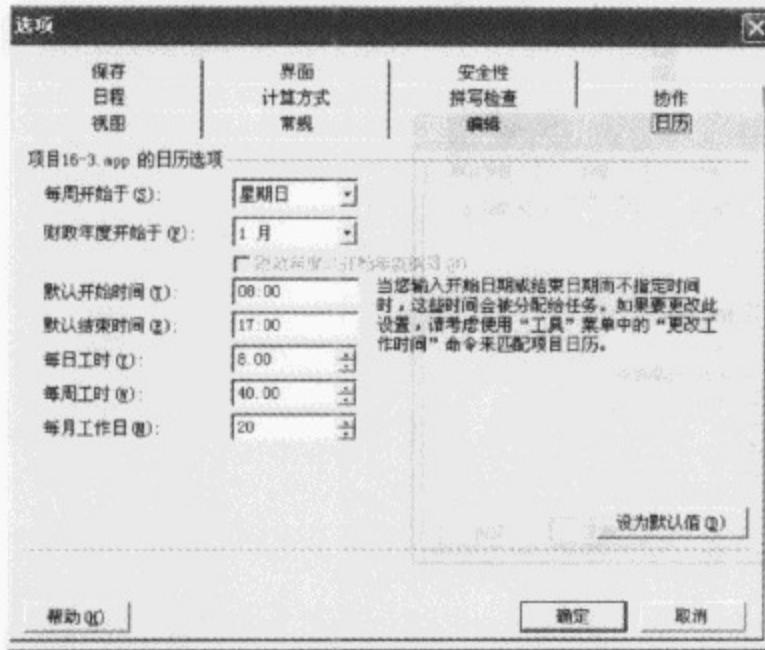


图 15-5 “日历”选项卡

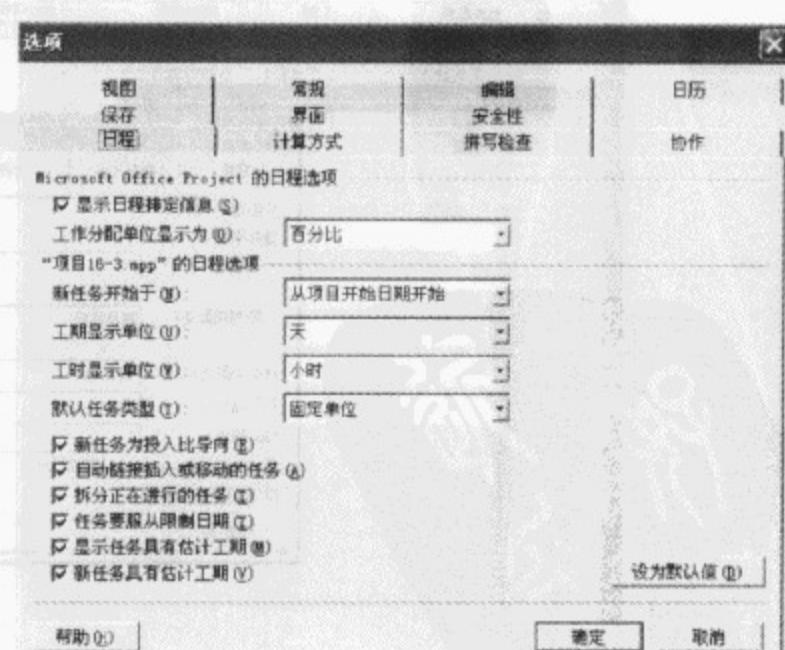


图 15-6 “日程”选项卡

## 15.1.2 创建项目任务

创建一个新的项目文件后, 接下来需要进行任务的创建。任务是构成项目的基本单元, 所有的任务完成了, 项目才可以完成。实验步骤如下。

### 1. 任务建立

- 1) 打开项目文件 SchoolWebside.mpp。
- 2) 选择 [视图]→[甘特图] 菜单命令切换到“甘特图”视图, 默认的视图就是甘特图。

3) 在“任务名称”域中输入项目的任务名称，例如“需求分析”、“系统设计”、“编码”、“测试”、“提交”等，如图 15-7 所示。

	任务名称	工期	开始时间	2008年05月04日	2008年05月11日
	任务	资源	跟踪	报表	显示
1	需求分析	1 工作日?	2008年05月12日		
2	系统设计	1 工作日?	2008年05月12日		
3	编码	1 工作日?	2008年05月12日		
4	测试	1 工作日?	2008年05月12日		
5	提交	1 工作日?	2008年05月12日		

图 15-7 创建任务

4) 输入所有的任务。

## 2. 里程碑建立

根据项目要求设置相应的里程碑，例如需求、设计、编码、测试、提交等可以设置为里程碑。以“编码”里程碑为例，双击“编码”任务，弹出如图 15-8 所示的对话框，选择“高级”选项卡，在左下角选中“标记为里程碑”复选框。

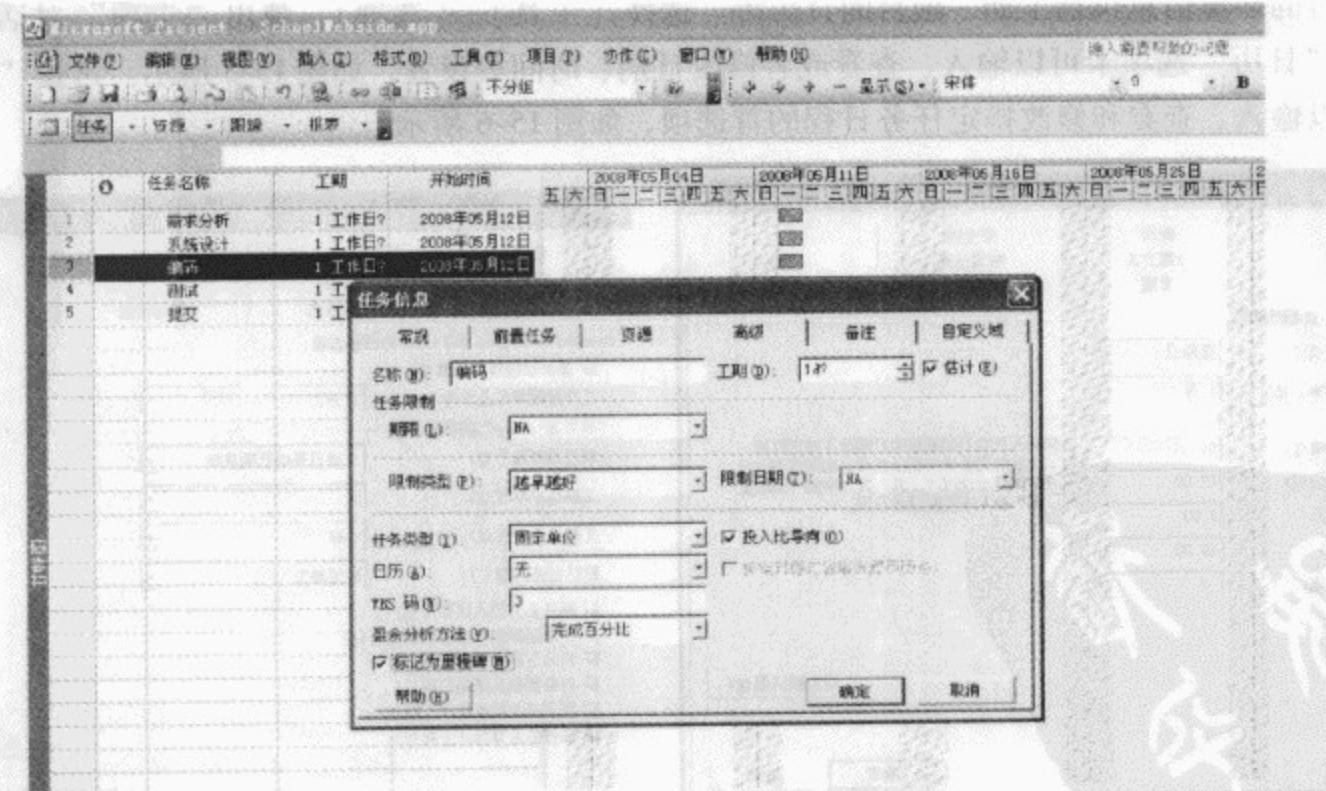


图 15-8 设置里程碑

## 3. 增加任务，建立大纲结构

前面建立的任务基本上是摘要任务，为此需要增加任务。为了很好地组织和管理任务，最好是采用大纲结构，即通过创建摘要任务和子任务来组织项目结构。摘要任务是由子任务组成并对子任务进行汇总的任务。具体步骤如下：

1) 在“任务名称”域中选择“编码”任务，选择 [插入]→[新任务]，建立新任务“界面设计”，同样插入其他任务“结构设计”、“数据设计”等。

2) 重复上步，直到加入所有的任务。

3) 在“任务名称”域中选择“界面设计”、“结构设计”、“数据设计”，单击[项目]→[大纲]→[降级]命令，进行降级，如图15-9所示。这样，就可以将摘要任务“系统设计”和它的子任务清晰地显示出来。

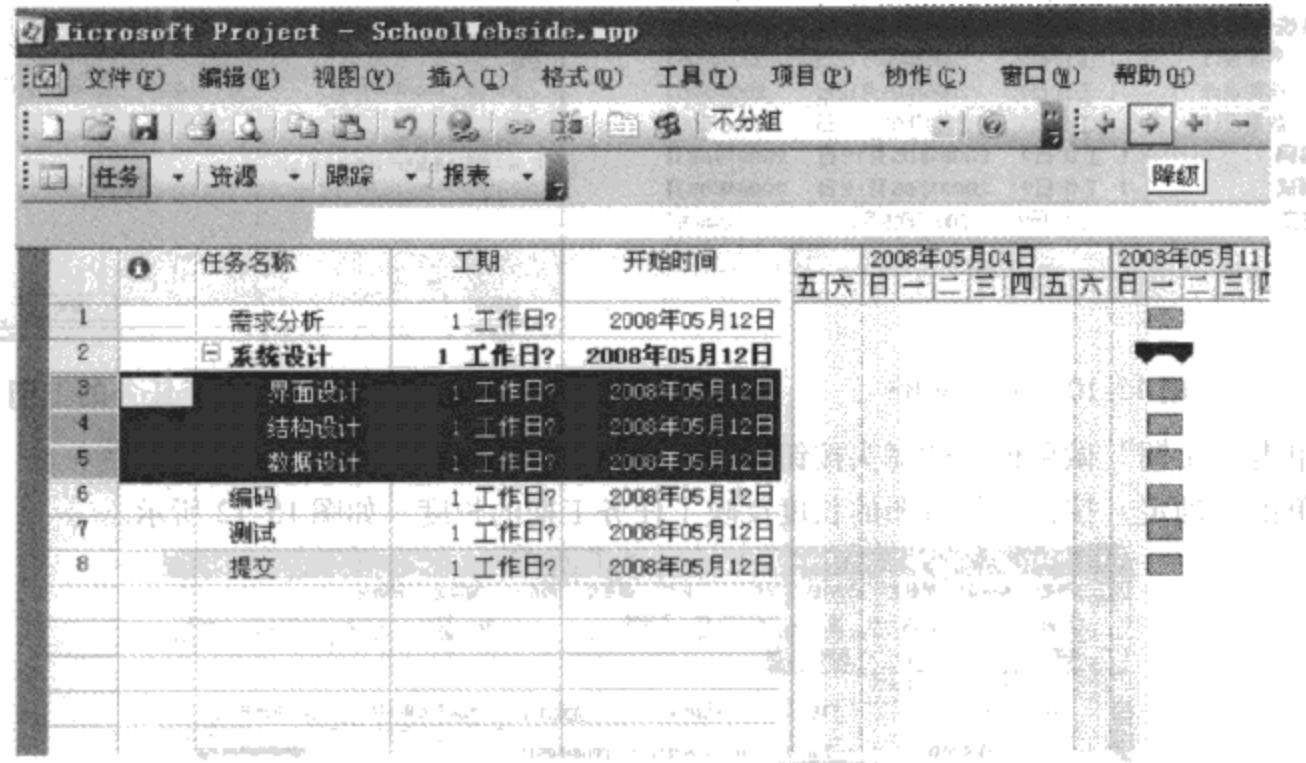


图15-9 建立大纲结构

4) 同理，完成其他任务的升级或者降级。

设置好项目任务的分级结构以后，可以很容易看清楚任务之间的顺序。

#### 4. 任务的移动、复制和显示

当对任务和资源进行复制或者移动时，将重新建立任务链接关系，通常还包括与任务相关的子任务、备注信息和链接或者嵌入对象。

##### 15.1.3 任务分解

创建好的任务可以进一步分解，以完成WBS。在MS Project中，可以为每项任务建立WBS。实验步骤如下：

- 1) 在SchoolWebside.mpp项目中，切换到“甘特图”视图。
- 2) 单击“任务名称”域的列标题。
- 3) 选择[插入]→[列]命令，启动“列定义”对话框。
- 4) 在“域名称”下拉列表框中选择WBS选项。
- 5) 在“对齐数据”下拉列表框中选择“居中”选项。
- 6) 单击“最佳匹配”按钮，生成如图15-10所示的具有WBS的甘特图。

##### 15.1.4 安排任务工期

任务创建和分解后，需要为所有的任务安排工期，以便确定项目的完成时间。首先确定任务的类型，任务类型包括固定单位任务、固定工期任务和固定工时任务。实验步骤如下。

#### 1. 安排任务的进度

可以采用三种方法为任务制定工期：1) 在甘特图表中的“工期”域直接输入工期；2) 用鼠标左键拖动甘特图的任务条形图，通过改变其长度来改变其工期；3) 使用“任务信息”对话框设置工期。具体步骤如下：

- 1) 双击甘特图中“任务名称”域中“需求讨论”子任务，出现如图15-11所示的对话框，设置

子任务的工期。摘要任务是其子任务的时间总和。

WBS	任务名称	工期	开始时间	完成时间
1	需求分析	1 工作日?	2008年05月12日	2008年05月
2	系统设计	1 工作日?	2008年05月12日	2008年05月
2.1	界面设计	1 工作日?	2008年05月12日	2008年05月
2.2	结构设计	1 工作日?	2008年05月12日	2008年05月
2.3	数据设计	1 工作日?	2008年05月12日	2008年05月
3	编码	1 工作日?	2008年05月12日	2008年05月
4	测试	1 工作日?	2008年05月12日	2008年05月
5	提交	1 工作日?	2008年05月12日	2008年05月

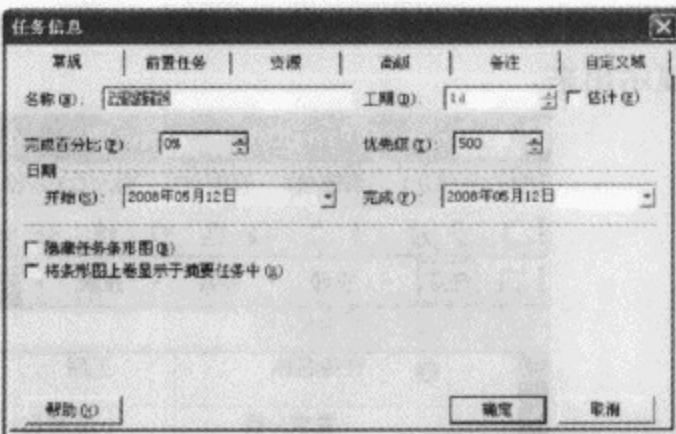


图 15-10 创建 WBS

图 15-11 设置子任务的工期

- 2) 单击“工期”微调框中的箭头按钮可以修改工期。
- 3) 单击[确定]按钮，甘特图的长度反映了任务工期的长度，如图 15-12 所示。

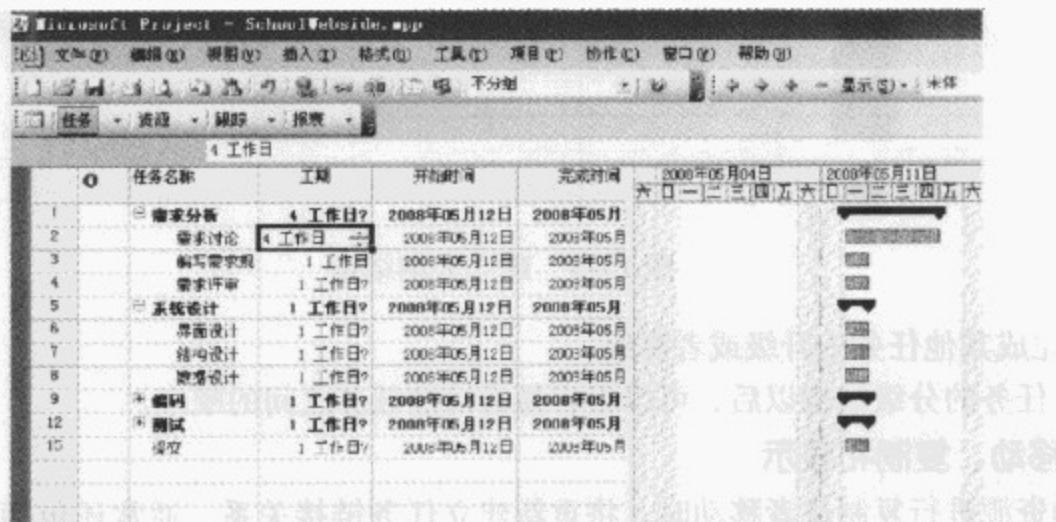


图 15-12 调整任务的工期，项目甘特图

## 2. 周期性任务

在项目的任务中插入周期性任务，例如每周五的周例会，如图 15-13 所示。

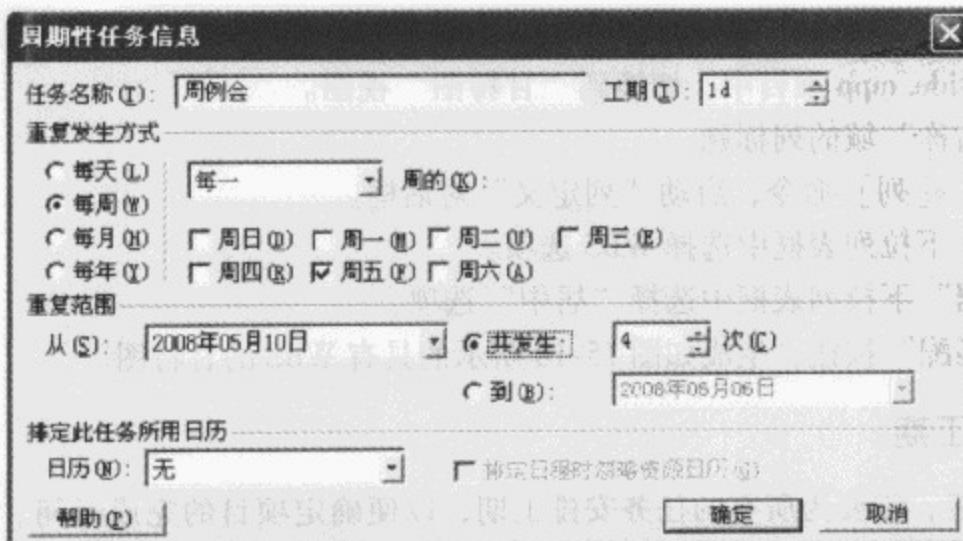


图 15-13 设置周期任务

### 15.1.5 任务的链接

任务的链接是将任务与其他任务的时间安排联系在一起，通过任务的链接而得到任务的开始或者完成时间，特点是任务之间始终被这种特定的关系约束着。实验步骤如下。

### 1. 建立链接

- 在“任务名称”域选择“需求讨论”、“编写需求规格”、“需求评审”三个任务。
- 单击“常见”工具栏中的[链接任务]按钮，即可以完成设置三个任务的“结束-开始”的关系。如图15-14所示。

### 2. 删除链接

- 选择要取消链接的任务“需求讨论”、“编写需求规格”。
- 单击“常用”工具栏中的[取消任务链接]按钮或者选择[编辑]→[取消任务链接]命令。

#### 15.1.6 资源分配

为了完成任务必须为项目中的任务分配资源，在Microsoft Project中可以建立资源库，然后将资源库中的资源与项目的任务联系起来，也就是说分配具体的人员、设备等来完成工作。实验步骤如下。

图15-14 建立任务的链接

##### 1. 建立资源库

- 选择[视图]→[资源工作表]命令，切换到“资源工作表”视图。
- 在视图表中输入项目所有的资源信息名称，如图15-15所示。

	资源名称	资源类型	材料耗益	缩写	组	最大单位	标准费率	加班费率	每次使用成本	成本累加
1	张三	工时		张		100%	\$100.00/工时	\$100.00/工时	\$0.00	\$0.00
2	李四	工时		李		100%	\$90.00/工时	\$120.00/工时	\$0.00	\$0.00
3	王五	工时		王		100%	\$0.00/工时	\$0.00/工时	\$0.00	\$0.00

图15-15 设置资源库

- 双击资源名称“张三”，设置其资源日历，在“资源信息”对话框中的“工作时间”选项卡设置资源日历，将“2008-05-10”周六改为工作时间（默认情况是周六为非工作时间）。

##### 2. 为任务分配资源

合理地向任务分配资源才能有效地完成项目任务。具体步骤如下：

- 切换到“甘特图”视图。
- 选择第一个任务“需求讨论”，选择[工具]→[分配资源]菜单命令。
- 在“分配资源”对话框中，选择分配给该任务的资源：张三、李四。
- 单击[分配]按钮，完成该任务的分配工作。
- 依次选中其他的任务，重复步骤2~4直到所有的任务都被分配了资源，如图15-16所示。

#### 15.1.7 项目管理

建立一个项目后需要对项目进行管理，查看项目进展情况。实验步骤如下。

- 查看项目的各种视图，熟悉使用各种视图。

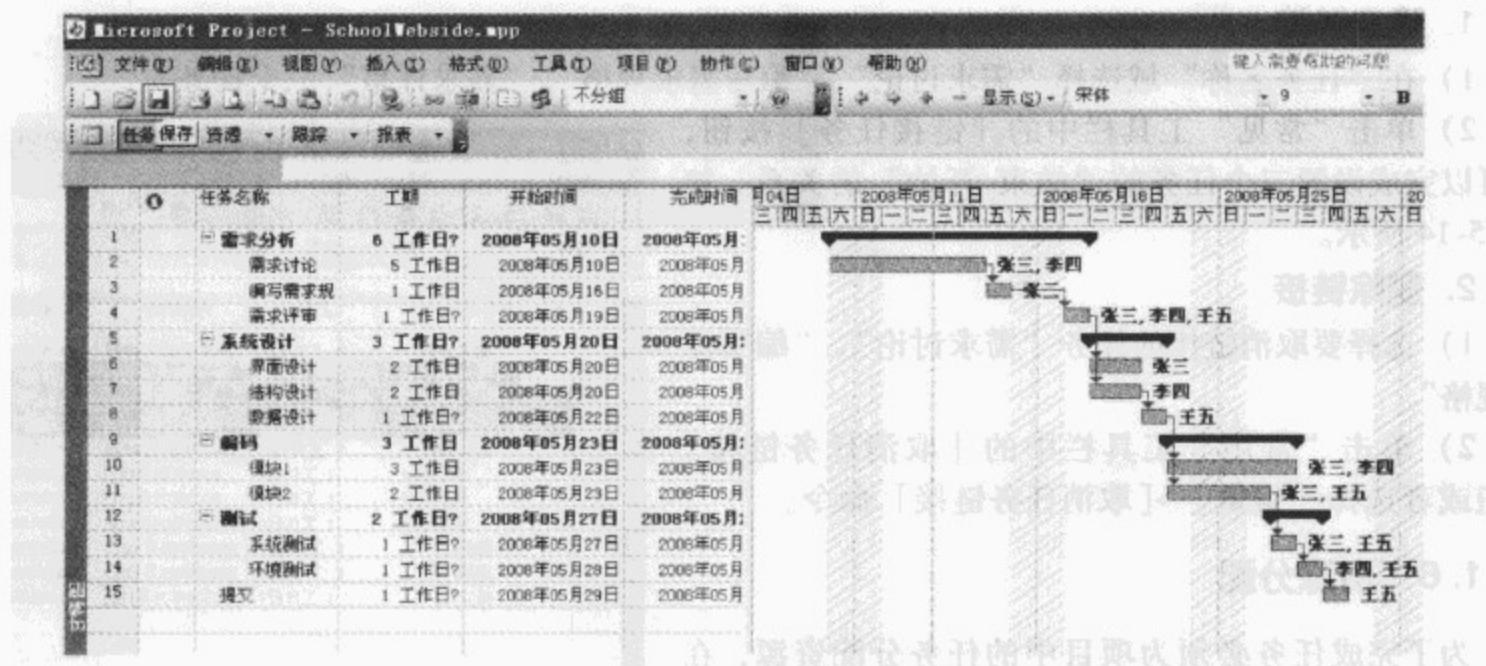


图 15-16 分配资源

- 甘特图：甘特图是默认的视图，是项目进度管理中最常用的视图，通过 [视图]→[甘特图] 菜单命令进入甘特图视图。
  - “日历”视图：通过 [视图]→[日历] 菜单命令可以进入“日历”视图。
  - “网络图”：通过 [视图]→[网络图] 菜单命令可以进入“网络图”视图。
- 2) 查看项目的成本。给任务分配资源后，可以查看任务的成本。具体步骤如下：
- 打开 SchoolWebside.mpp 项目。
  - 选择 [视图]→[甘特图] 命令，切换到“甘特图”视图。
  - 选择 [视图]→[表]→[成本] 命令，显示项目的任务成本信息。

## 15.2 IBM RPM 实验

本次实验是通过使用 IBM Rational Portfolio Manager 辅助完成项目管理的一些工作，目的是了解 IBM Rational Portfolio Manager 工具的使用和项目管理的相关知识。实验内容和步骤如下。

### 15.2.1 登录 IBM RPM

实验步骤如下。

#### 1. 打开 RPM 客户端

在桌面上双击 RPM 客户端快捷方式或点击 [开始] 菜单→[所有程序]→[IBM Rational]→[IBM Rational Portfolio Manager]→[Rational Portfolio Manager]。

#### 2. 配置客户端

1) 在 RPM 登录界面（见图 15-17）上点击“Configuration...”按钮。

2) 在“Profile Configuration”对话框中点击“Add”按钮添加新的 profile。

#### 3) 输入 profile 信息：

- Profile Name：给新的 profile 起一个名字
- Server Name：localhost: 9080/webapp/IBMRPM
- Connection Type：http
- Application Name：IBMRPM



图 15-17 RPM 登录界面

4) 点击 OK 确认。

### 3. 登录 RPM

选择刚才新建的 profile，输入用户名和密码，点击“Sign In”登录，如图 15-18 所示。

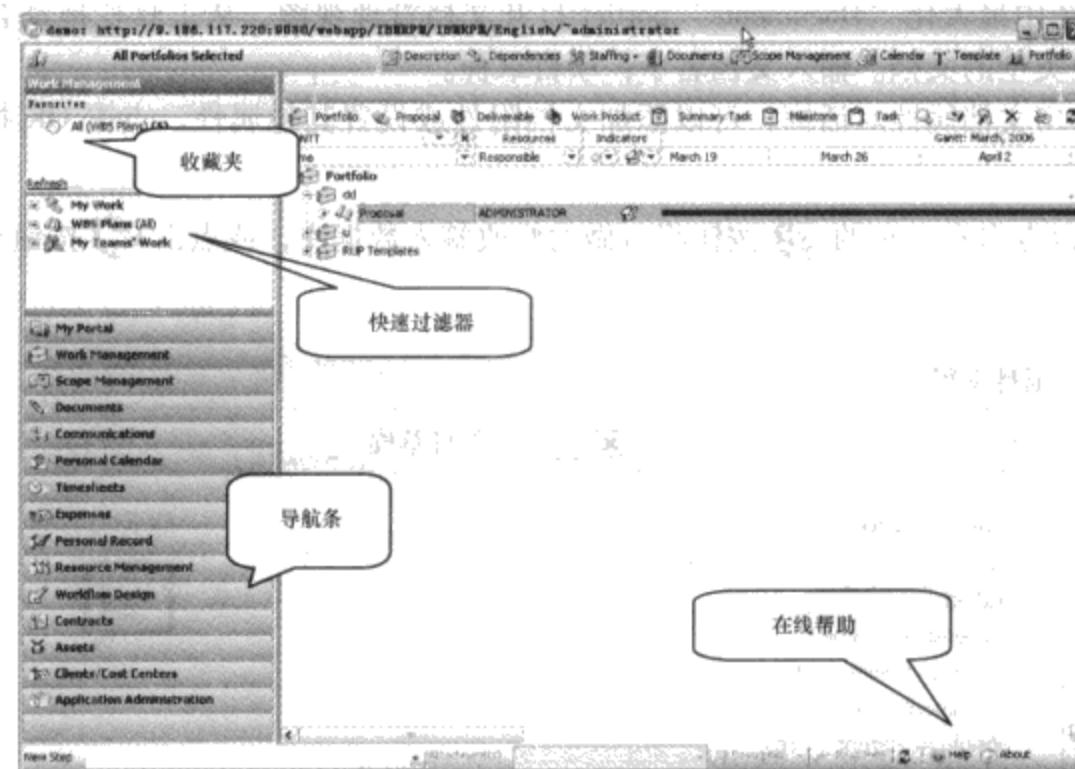


图 15-18 RPM 界面

#### 15.2.2 创建项目提案

实验步骤如下：

- 1) 登录 RPM，选择“Work Management”视图。
- 2) 在工具栏中点击“Proposal”图标 Proposal，拖动到工作区的适当位置。
- 3) 输入提案名字。
- 4) 回车或点击工作区其他位置将弹出“New Proposal”对话框，在其中输入内容或修改默认值，点击“OK”即可创建一个新的项目提案，如图 15-19 所示。

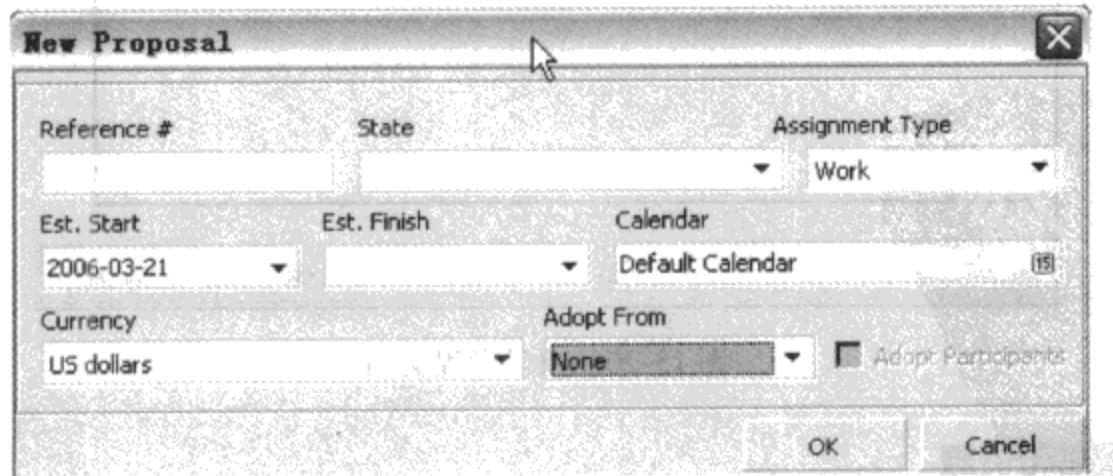


图 15-19 创建新项目提案

- 5) 若在“New Proposal”对话框的“Adopt From”列表中选择“Template”将弹出模板选择窗口，可以选择适当的模板创建提案。
- 6) 双击新建的提案或选定新建的提案后点击工具栏中的“Description”按钮 Description，打开提案的描述视图，点击右上角的箭头按钮可以将提案签出进行修改。

### 15.2.3 文档管理——项目章程

实验步骤如下：

1) 在描述视图中最大化“Documents”portlet 打开文档管理视图，从工具栏拖动合适的图标到视图中即可创建相应类型的文档。输入文档名字：项目章程。

按回车后将弹出文档属性对话框，可以添加文档附件或设置其他属性。点击OK按钮保存修改并上传文档附件。

2) 修改完可以点击右上角的 按钮将所做的修改签入数据库，或点击 按钮放弃修改，见图 15-20。

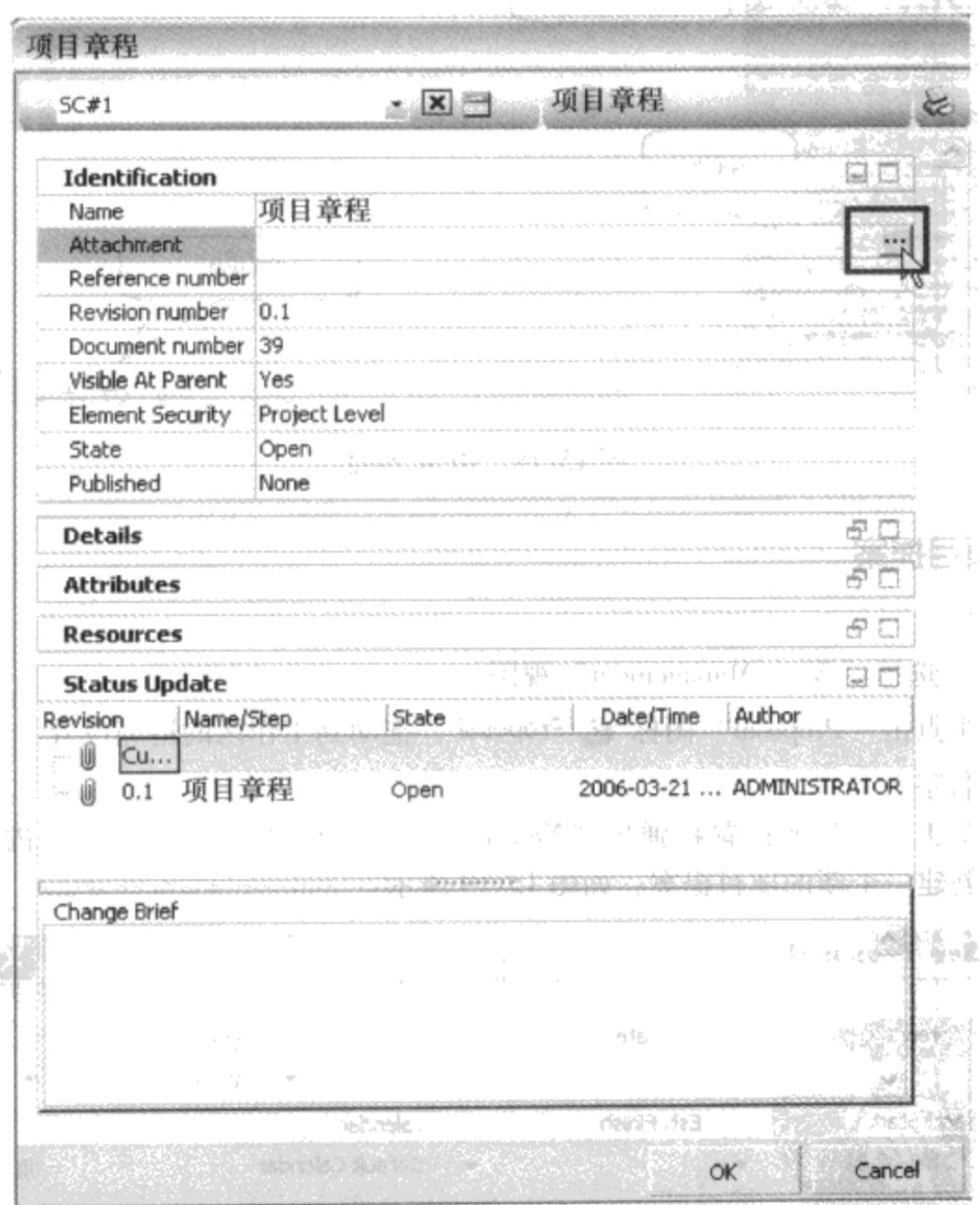


图 15-20 项目章程文档

### 15.2.4 资源池和资源

实验步骤如下：

- 1) 登录 RPM，选择“Resource Management”视图。
- 2) 从工具栏选择资源池图标 Pool，拖动到工作区的适当位置。
- 3) 输入资源池名字即可创建一个新的资源池，见图 15-21。
- 4) 双击资源池打开描述视图，编辑资源池属性。

5) 在工具栏上选择“Resource”图标 Resource，拖动到新建的资源池下方，即可创建新的资源记录，见图 15-22。

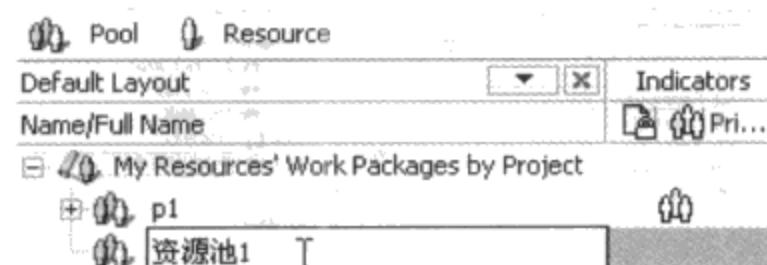


图 15-21 创建资源池

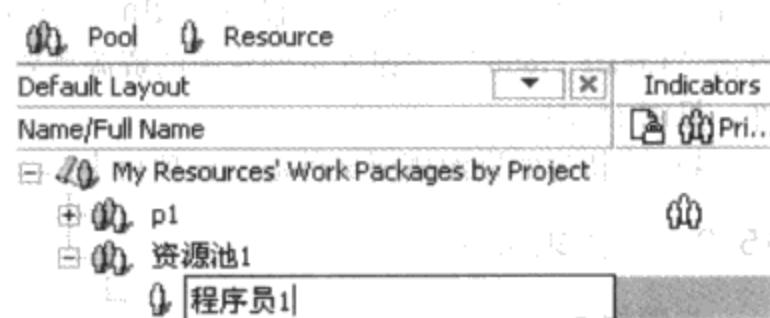


图 15-22 创建新的资源记录

6) 打开资源描述视图，编辑资源属性。其中 Username 即为新的资源登录 RPM 系统的登录名。必须为新建的资源选择一个合适的日历，见图 15-23。

图 15-23 编辑资源属性

- 7) 将“Competencies”portlet 最大化，为资源选择合适的职务。
- 8) 将“Skills”portlet 最大化，为资源设置技能及其等级信息。
- 9) 将“Rates”portlet 最大化，设置资源的单元成本/收益率。
- 10) 将“System Security Rights”portlet 最大化，设置或修改资源的系统安全权限。
- 11) 点击 check in 按钮 将所做的修改签入数据库。
- 12) 以新建的用户名和默认初始密码“ibmrpm”从另一个 RPM 客户端登录。

### 15.2.5 资源需求描述

实验步骤如下：

- 1) 登录 RPM，选择“Work Management”视图。
- 2) 选中一个项目提案，点击 Staffing→Search/Assign 打开人员配备视图。

3) 在“Staffing View”中点击“Search/Create”栏，在“Search/Create”列表中选择“Profile”，在视图左边选择合适的属性，点击视图中间的“Add”按钮添加为选定标准，设置期望的工作量，点击“Create Profile”按钮即可创建新的资源概要文件，见图 15-24。

4) 将新建的资源概要文件添加到项目中。

### 15.2.6 需求管理

步骤如下：

1) 登录 RPM，进入“Scope Management”视图。

2) 从工具栏中选择“Requirement”图标，拖动到与提案名字对应的文件夹下方，见图 15-25。输入需求名称即可创建一个新的需求记录。

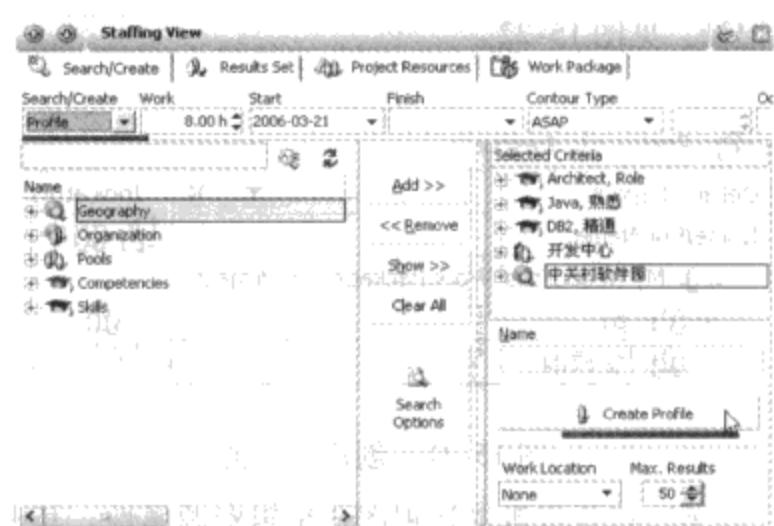


图 15-24 创建新的资源概要文件

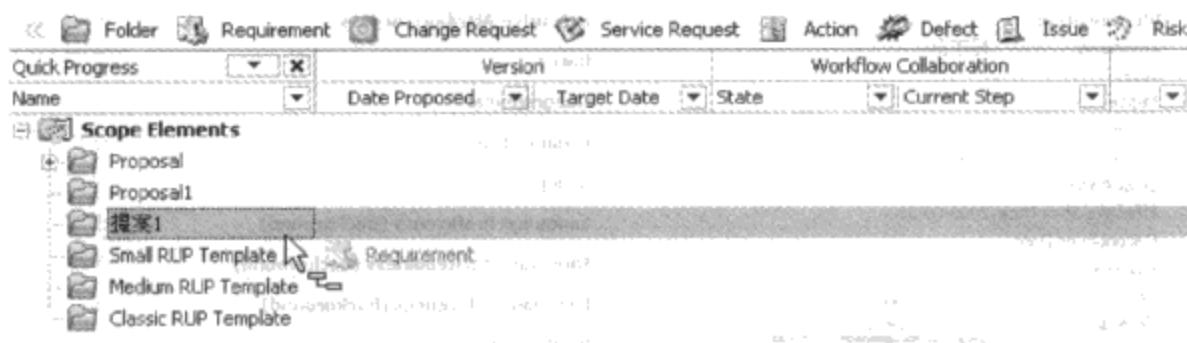


图 15-25 创建需求记录

- 3) 双击新建的需求打开描述视图，修改需求的属性。
- 4) 最大化“Requirement Description”portlet 编辑需求描述。
- 5) 在“Documents”portlet 中可以添加外部参考文档或 URL。
- 6) 在“Financials”portlet 中可以对需求进行财务估计。
- 7) 在“Resources”portlet 中可以为需求安排适当的资源。
- 8) 最大化“Attributes”portlet 可以为需求设置不同的属性值。
- 9) 右键点击需求项并选择菜单“Convert to ...”→“Requirement Task”可以将需求转换为需求任务。
- 10) 进入“Work Management”视图，选中提案后点击右上角的“Scope Management”图标 即可在“Work Management”工作区中打开范围管理视图，从范围管理视图中选中需求任务并将其拖动到工作区中提案的下方即可将需求任务加入项目提案的 WBS，见图 15-26。

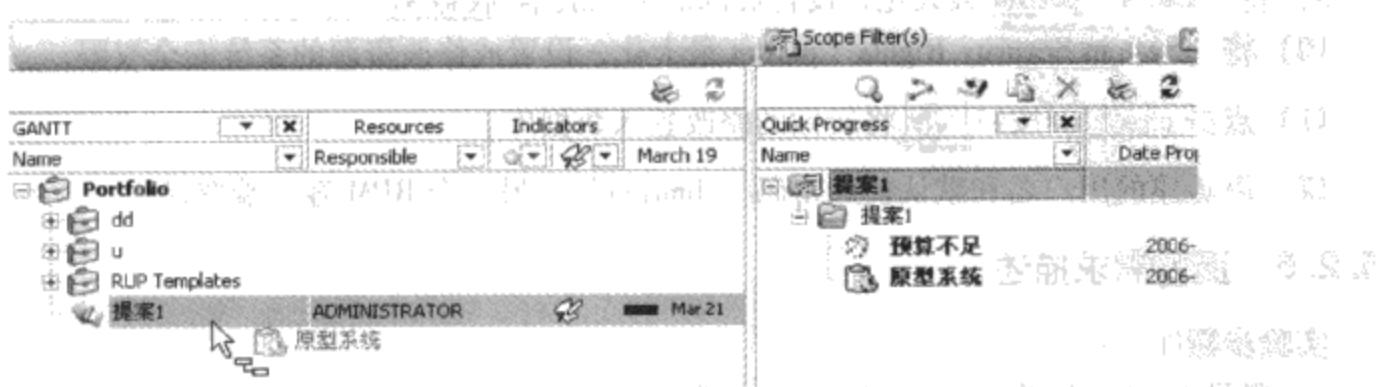
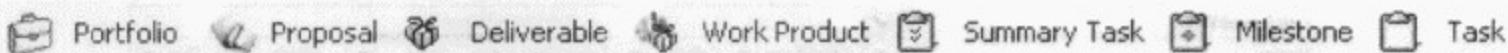


图 15-26 将需求任务加入 WBS

### 15.2.7 创建WBS

步骤如下：

- 1) 登录RPM，选择“Work Management”视图。
- 2) 从工具栏中选择想要创建的任务图标，拖动到工作区中适当项目/任务的下方，即可在该项目/任务下创建一个子任务，依此类推可创建项目WBS。



- 3) 在工作管理视图中，点击右上角的Template图标，打开模板浏览器。

选择想要的项目模板或WBS模板片断，拖动到工作区中适当的项目/任务的下方即可根据模板创建WBS。

- 4) 选择一个任务，打开描述视图，最大化“Schedule Dates”portlet，设置任务的工期或开始/结束时间，设置结束后最小化portlet回到描述视图，见图15-27。

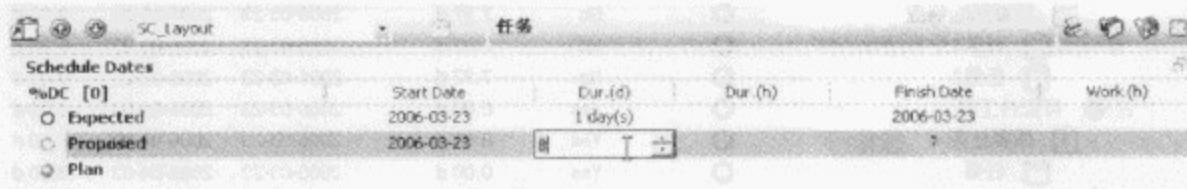


图15-27 设置任务的工期

- 5) 最大化“Constraints”portlet，设置任务的约束类型/约束日期。
- 6) 将所做的修改签入数据库，关闭描述视图。
- 7) 重复步骤2~4可以设置其他任务的进度。
- 8) 在工作区中选中一个任务，选择右上角的“Dependencies”图标，打开依赖关系视图，从最上面一栏选择项目/任务，点击“Add”按钮将其添加为当前工作区中选中的任务的前置任务，修改依赖关系类型和lead/lag时间，见图15-28。



图15-28 设置依赖关系

9) 设置完所有依赖关系以后回到“Work Management”工作区，右键点击项目提案，选择菜单 Calculate/Level→Calculate，系统将自动根据设置的工作量、约束条件、依赖关系等计算项目进度。选择一个日期作为计算的起始时间。

10) 在沟通视图中查看计算进度。

11) 计算完成后回到工作视图，在布局栏中选择“Calc/Level View”查看计算结果，见图 15-29。

Name	No	Critical	Total Slack	Start	Finish	Dur.
提案1	○	Yes	0.00 d	2006-03-23	2006-04-03	8.00 d
原型系统	○	Yes	0.00 d	2006-03-23	2006-03-23	0.03 d
预算不足	○	No	8.00 d	2006-03-23	2006-03-23	0.03 d
预算风险响应	○	No	7.97 d	2006-03-23	2006-03-23	0.03 d
任务1	○	No	7.97 d	2006-03-23	2006-03-23	0.03 d
任务2	○	No	7.97 d	2006-03-23	2006-03-23	0.03 d
可交付工件	○	Yes	0.00 d	2006-03-23	2006-04-03	8.00 d
概要任务	○	Yes	0.00 d	2006-03-23	2006-04-03	8.00 d
任务	○	Yes	0.00 d	2006-03-23	2006-04-03	8.00 d
工作产品	○	No	7.00 d	2006-03-23	2006-03-23	1.00 d
里程碑	○	Yes	0.00 d	2006-04-03	2006-04-03	
概要任务	○	No	7.00 d	2006-03-23	2006-03-23	1.00 d

图 15-29 查看计算结果

12) 在布局栏中选择“GANTT”可以查看甘特图。右键点击甘特图，点击“Today”，可以将甘特图滚动到当前日期，在右键菜单中还可以对甘特图进行缩放或其他设置操作，点击菜单项“Show Calc/Level”，可以在甘特图中显示计算结果。

13) 在工作区中右键点击项目提案，选择菜单 Calculate/Level→Copy Calculated Results 将计算结果复制到项目计划。

## 15.2.8 项目人员配备

步骤如下：

1) 登录 RPM，选择“Work Management”视图。

2) 选中一个项目提案，点击 Staffing→1Search/Assign 打开人员配备视图。

3) 在“Staffing View”中点击“Search/Create”栏，在“Search/Create”列表中选择“Resource”，设置期望的工期，添加其他资源属性作为搜索标准，点击“Search”搜索满足条件的资源，见图 15-30。

4) 在“Results Set”栏可以查看搜索的结果。选择合适的资源，点击“Assign”按钮可以将资源预留到项目中或添加为项目参与者，见图 15-31。

5) 关闭人员配备视图回到“Work Management”工作区，选中一个项目提案，点击 Staffing→1 Search/Assign 打开人员配备视图。

6) 选中一个资源概要文件，点击右上

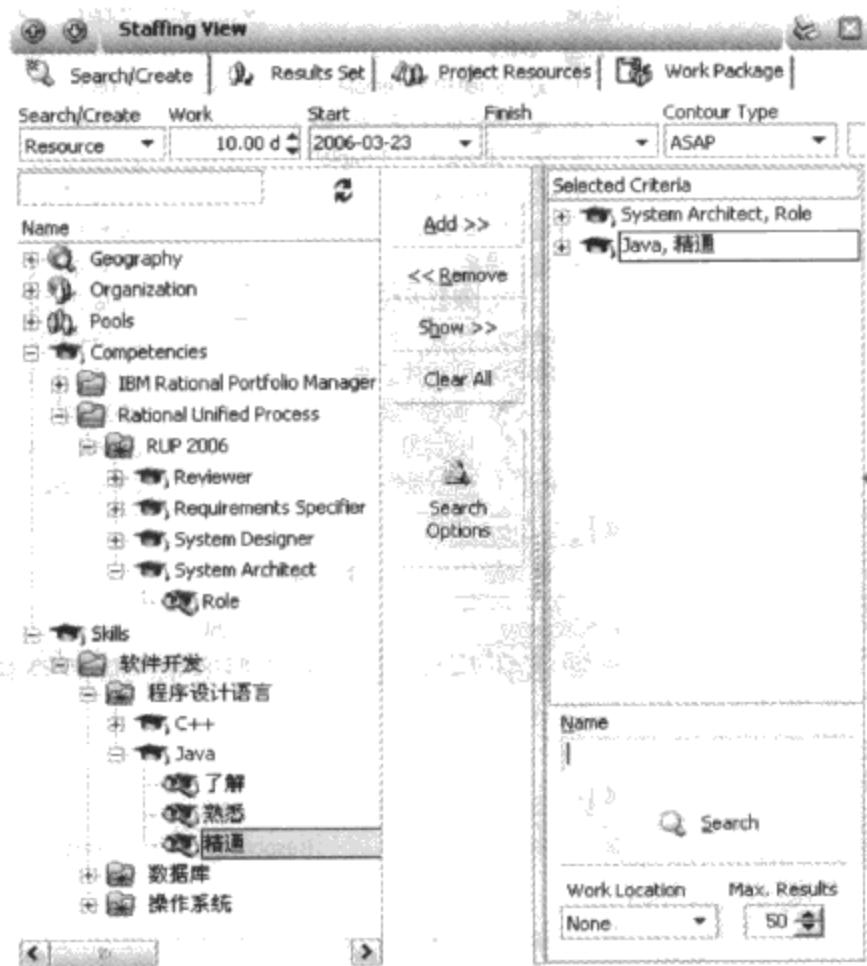


图 15-30 搜索资源

角的 按钮，打开资源概要文件，替换视图。

- 在“Search/Create”页中可以看到在资源概要文件中设置的工期和其他资源属性已经自动作为搜索标准，点击“Search”按钮搜索满足条件的资源。
- 在“Results Set”页中可以查看搜索的结果，选中合适的资源，点击“Replace”按钮即可将项目中的资源概要文件替换为真实的资源。

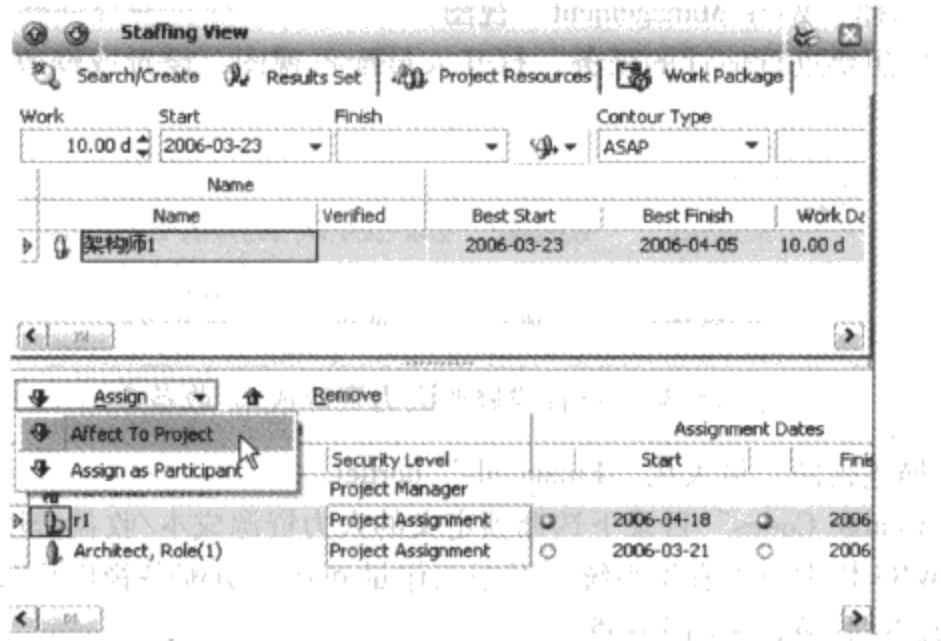


图 15-31 分配资源到项目中

- 7) 在“Work Management”工作区中选中一个项目提案，在右键菜单中选择发起一个人员配备请求。
- 8) 以刚才选中的资源经理身份登录 RPM，在“Communications”视图中查看人员配备请求。
- 9) 此时项目经理可以在他的“Communications”视图中查看从资源经理发回的响应并点击附件链接打开概要文件替换视图，对每个候选资源，项目经理可以在“Accepted”栏中选择接受或拒绝，如图 15-32 所示。

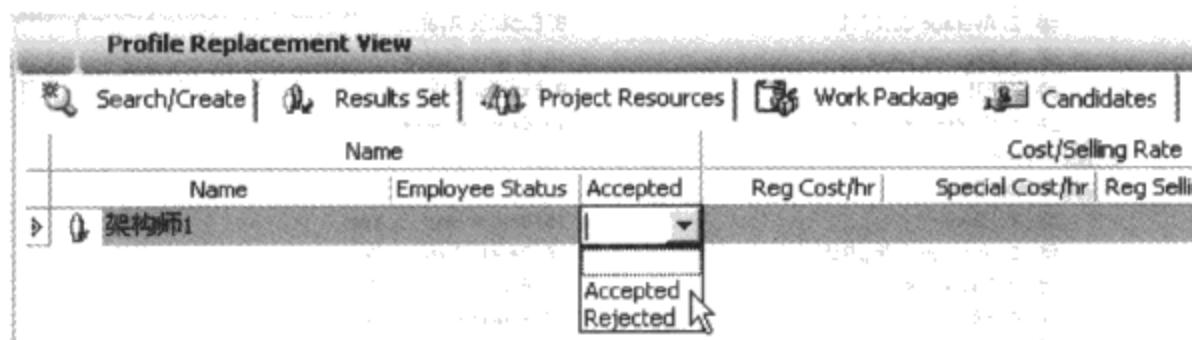


图 15-32 接受或拒绝候选资源

- 10) 完成候选人选择后关闭“Profile Replacement View”回到人员配备视图，点击“Submit Accepted Candidate”按钮提交选择结果，系统将发送一个“Resource Assignment”请求给资源经理，见图 15-33。

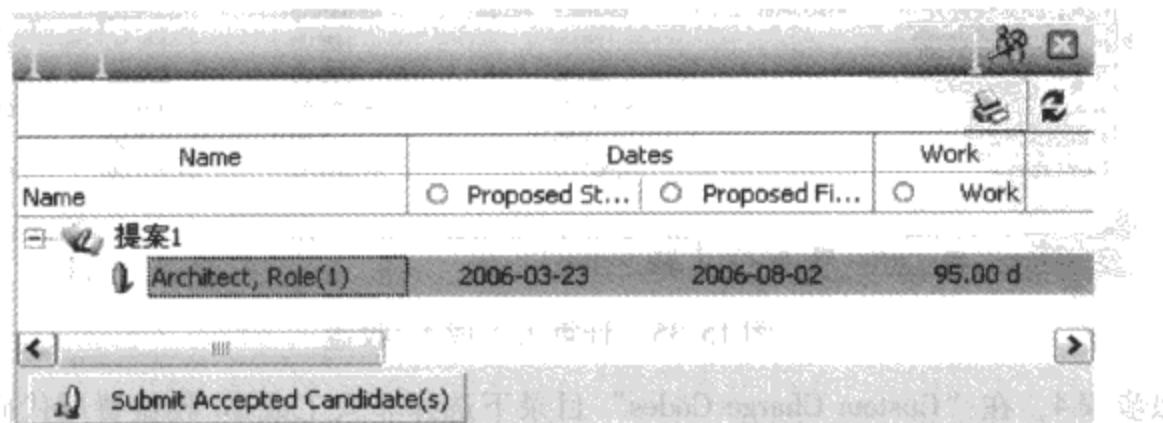


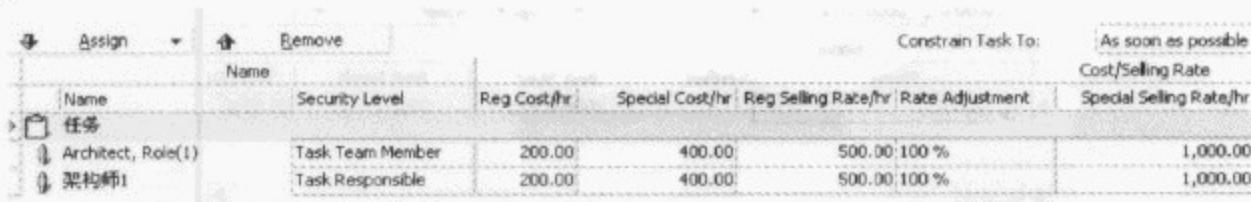
图 15-33 提交选择结果

此时，根据权限设置，项目经理或资源经理可以在资源概要文件替换视图中点击“Replace”按钮将资源概要文件替换为真实的资源。

### 15.2.9 项目任务的财务估算

项目任务财务估算的实验步骤如下：

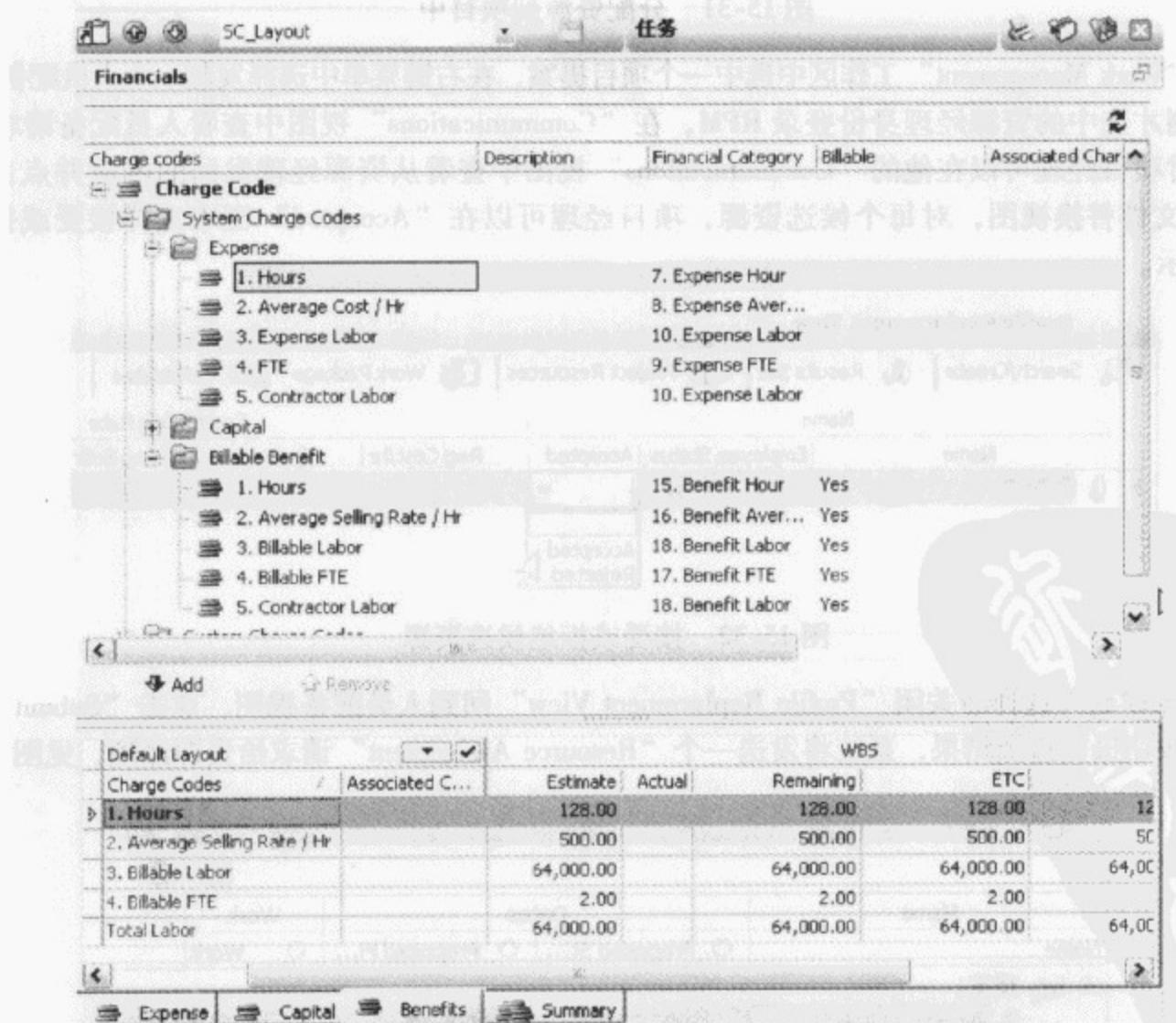
- 1) 登录 RPM，选择“Work Management”视图。
- 2) 在工作区选中想要进行估计的任务，打开人员配备视图，设置或修改人力资源成本/收益率，见图 15-34。



Name	Security Level	Reg Cost/hr	Special Cost/hr	Reg Selling Rate/hr	Rate Adjustment	Constrain Task To:	Cost/Selling Rate	As soon as possible	Special Selling Rate/hr
任务									
Architect, Role(1)	Task Team Member	200.00	400.00	500.00	100 %			1,000.00	
架构师	Task Responsible	200.00	400.00	500.00	100 %			1,000.00	

图 15-34 设置或修改人力资源成本/收益率

- 3) 打开任务的描述视图，最大化“Financials”portlet。
- 4) 在“System Charge Codes”目录下选择预定义的人力资源成本/收益费用代码并点击“Add”按钮添加。在下面的 WBS 栏中可以看见系统已根据工作量和在人员配备视图中设置的人力资源成本/收益率自动计算人工成本/收益，见图 15-35。



The screenshot shows the "Financials" portlet in the "SC\_Layout" workspace. The top navigation bar includes icons for Back, Forward, Home, and Help. The main area displays a tree view of charge codes under "Charge codes". The "Expense" category is expanded, showing items like "1. Hours", "2. Average Cost / Hr", "3. Expense Labor", "4. FTE", and "5. Contractor Labor", each associated with a description and financial category. Below the tree view is a table for "WBS" estimation. The table has columns for "Default Layout", "Charge Codes", "Associated C...", "Estimate", "Actual", "Remaining", and "ETC". The first row, "1. Hours", is selected and highlighted in grey. The "Estimate" column for "1. Hours" contains the value "128.00". The "Remaining" and "ETC" columns also show "128.00". Other rows in the table include "2. Average Selling Rate / Hr" (500.00), "3. Billable Labor" (64,000.00), "4. Billable FTE" (2.00), and "Total Labor" (64,000.00). At the bottom of the portlet, there are tabs for "Expense", "Capital", "Benefits", and "Summary".

图 15-35 计算人工成本/收益

- 5) 类似步骤 4，在“Custom Charge Codes”目录下选择非人工成本/收益费用代码，点击“Add”按钮添加，在视图下部输入估计数值。

- 6) 继续上述步骤，设置其他任务的财务估计。
- 7) 在项目提案或概要任务的“Financials”portlet中可以查看汇总的数据，在Time-Phased页中可以查看根据工期设置自动计算的分时间段财务估计数据，见图15-36。

The screenshot shows a table titled 'Financials' with columns for 'Name', 'Source', and months from January to April. The table is divided into sections: 'Expense', 'Revenue/Benefits', and 'Capital'. Under 'Expense', there are rows for 'Total Labor' and 'Capital'. Under 'Revenue/Benefits', there are rows for 'Total Labor'. The 'Capital' section is currently empty. The 'Revenue/Benefits' section shows data for 'Total Labor' with 'Estimate (WBS)' values of 56,000.00 and 8,000.00 for January and February respectively, and 'Actual (WBS)' values of 56,000.00 and 8,000.00 for March and April respectively.

图15-36 财务估计数据

### 15.2.10 风险分析

风险分析的实验步骤如下：

- 1) 登录RPM，进入“Scope Management”视图。
- 2) 从工具栏中选择“Risk”图标 Risk，拖动到与提案名字对应的文件夹下方，输入需求名称即可创建一个新的需求记录。
- 3) 双击新建的风险项打开描述视图，修改需求的属性。
- 4) 在“Risk Description”portlet中输入风险描述。
- 5) 在“Closure Criteria”portlet中输入关闭标准。
- 6) 在“Scorecard”portlet中可以利用计分卡对风险的影响进行量化评估。
- 7) 在“Financials”portlet中可以评估风险对项目财务的影响。
- 8) 在“Mitigation/Impact”portlet中设置风险的影响。
- 9) 在“Risk Matrix”portlet中设置风险矩阵，见图15-37。
- 10) 在“Scope Management”工作区中，从工具栏选择合适的任务图标，拖动到风险项的下方，即可为风险项创建响应包。
- 11) 进入“Work Management”视图，选中提案后点击右上角的“Scope Management”图标 Scope Management 即可在“Work Management”工作区中打开范围管理视图，从范围管理视图中选中风险项并将其拖动到工作区中提案的下方即可将风险响应包加入项目提案的WBS。

The screenshot shows a table titled 'Risk Matrix' with columns for 'Impact', 'Probability', 'Precision', 'Consequence Cost', 'Probability Cost', and 'Exposure %'. The values are: Impact 20, Probability 20, Precision Low, Consequence Cost 0.00, Probability Cost 0.00, and Exposure % 20. The 'Impact' row has a dropdown menu set to 'Green'.

图15-37 设置风险矩阵

### 15.2.11 工作流

步骤如下：

- 1) 登录 RPM，选择“Workflow Design”视图。
- 2) 在工具栏中选择“Process”图标  Process，拖动到工作区中“Project”的下方，输入工作流名字创建一个新的工作流，见图 15-38。
- 3) 双击新建的工作流打开工作流设计视图。
- 4) 点击工具栏上的  Steps 图标，然后点击设计工作区的适当位置创建一个新的步骤。
- 5) 双击新建的步骤打开步骤配置对话框，在“General”页上设置步骤名称、步骤描述、所处的状态以及权限控制，见图 15-39。



图 15-38 创建一个新的工作流

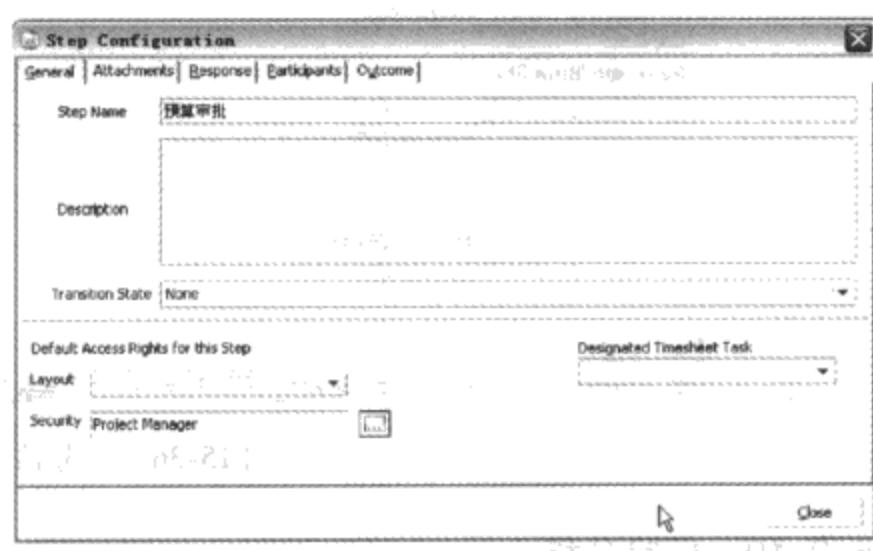


图 15-39 步骤配置对话框

- 6) 在“Attachments”页选择需要与步骤通知同时发送的附件链接。
- 7) 在“Response”页中首先选择响应类型，不同类型的响应需要设置不同的内容。
- 8) 对于表决类型，要设置表决的问题、期望的响应时间以及可能的响应；对“CheckList”类型则需要指定一系列应该完成的任务。
- 9) 对于表决类型和“CheckList”类型还需要在“Participants”页指定参与者。可以指定所有项目团队成员参与，或点击“Participants”按钮搜索指定的人员参与，或点击“Security Roles”按钮选择具有指定安全权限的人员参与，或点击“Workflow Roles”指定特定的工作流角色（项目经理可以将工作流角色映射到不同的团队成员）。
- 10) 在“Outcome”页中设置步骤结果。步骤包括两个部分即条件和结果，条件可以是表决结果或其他状态条件，结果可以是跳转到另一个步骤、变为新的状态、启动新的工作流或其他更复杂的动作。
- 11) 设计完所有步骤以后，关闭设计视图，回到工作区，右键单击新建的工作流并选择菜单项“Activate”激活工作流。
- 12) 为了在项目中使用工作流，项目经理还必须在项目中选择工作流。比如进入“Work Management”视图，打开项目提案的描述视图，最大化“Workflow Element Association”portlet，选中新建的工作流，签入所做的修改并关闭描述视图。
- 13) 在“Work Management”工作区中，右键点击项目提案并选择菜单“Communications”即可选择启动不同的工作流，见图 15-40。

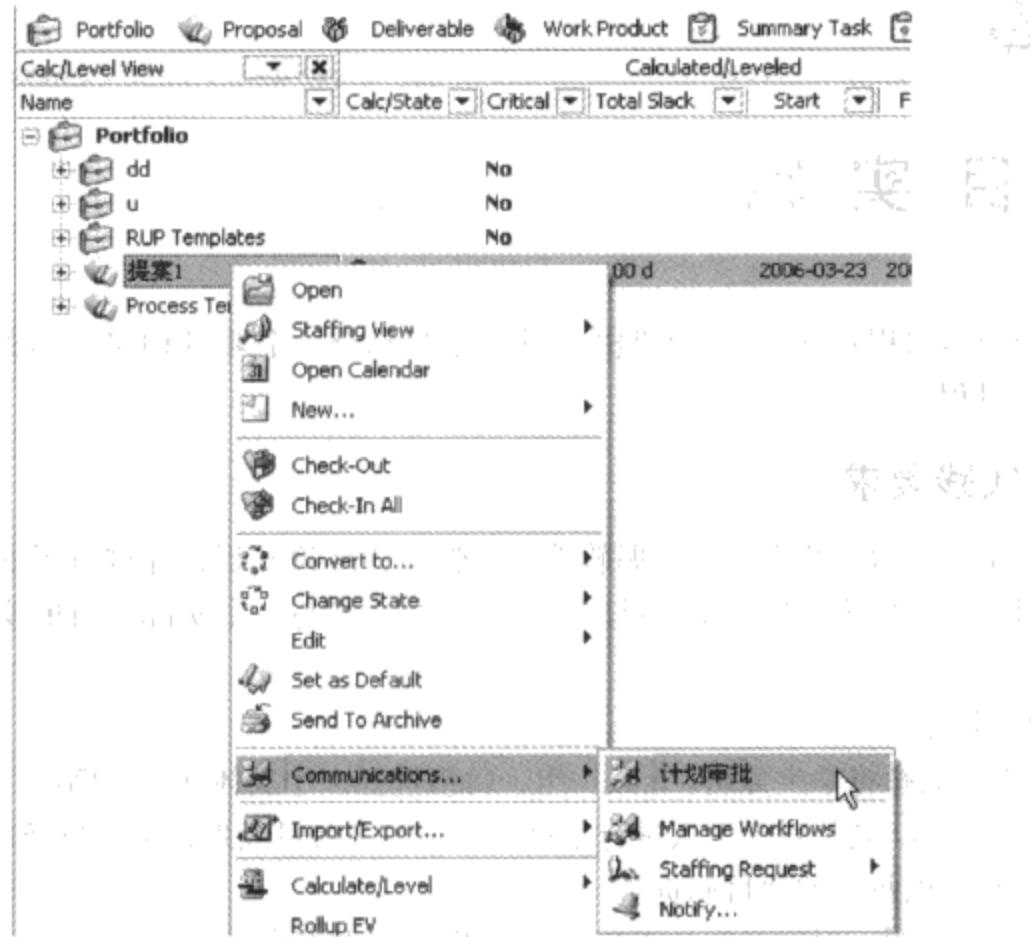


图 15-40 启动工作流

14) 工作流启动以后，每个参与者都可以在他的“Communications”视图中查看到正在进行的步骤并进行响应。

# 第 16 章

## ■ 项目实践

本章首先介绍本课程对学生项目实践的要求，然后结合一个采用 RPM 工具的项目案例，描述一个项目的管理实践过程。

### 16.1 项目实践要求

通过前面章节的学习，我们已经掌握软件项目管理的理论知识，并进行了课程实验，掌握如何使用工具进行项目管理。作为最后一章，可以让学生分组，参与一个实际项目的实践活动，具体实践要求如下。

#### 1. 团队组织建设

- 1) 项目分组。老师将学生分组，5人一组，每组选择一个具体项目，例如《学生成绩管理系统》。
- 2) 进行角色分工。老师对5个人进行角色分工，一个项目经理，由项目经理指定开发经理和质量经理，然后进一步明确每人的项目职责。
- 3) 编制组织结构图。根据角色分工和相关的职责，编制项目的组织结构图。
- 4) 设定开发过程规范。事前明确软件开发过程规范，包括需求过程、设计过程、编码过程、测试过程、提交过程，同时明确变更控制流程。

#### 2. 项目管理工具的搭建

根据具体情况选择项目管理工具，建议选择以下工具，同时搭建环境。

- 1) 搭建 Rational Portfolio Manager 环境。
- 2) 搭建 Microsoft project 环境。
- 3) 熟悉工具的使用。

#### 3. 编制项目计划

- 1) 项目经理组织人员编制项目计划。包括范围计划，成本计划，进度计划。
- 2) 采用 Rational Portfolio Manager 或者 Microsoft project 工具开发上述项目计划。
- 3) 计划评审。组织项目人员评审项目计划，必要时，修正计划，最后将计划存为基准计划。

#### 4. 项目跟踪管理

- 1) 按照项目计划组织团队实施项目。项目经理按照计划组织项目组的人员实施计划，开发软件项目。
- 2) 跟踪项目的进展。项目进行过程中，项目经理及时收集项目时间、资源、范围等信息，将这些信息输入到项目管理软件中。
- 3) 提交性能分析报告。根据收集的项目，利用项目管理软件提供的报表，编制项目的性能报告。
- 4) 进行项目评审，提交评审报告。每周进行定期的项目评审，最后提交项目的评审报告。
- 5) 处理项目的变更。项目经理根据变更控制流程处理项目变更信息。

#### 5. 演示汇报

项目结束后，以项目组的形式，演示汇报项目的整个管理过程。

- 1) 准备讲演文件。准备 PPT 文件讲述项目初始、计划、执行控制、结束等过程。
- 2) 现场展示系统，并讲述项目管理过程。现场展示开发的项目结果，并说明项目管理的过程，出现的问题以及如何解决的。
- 3) 提交项目总结报告。

## 16.2 基于RPM的项目管理实践案例

本节的案例是一个个人网上银行贷款系统，下面介绍采用RPM进行项目管理的过程。

### 16.2.1 项目初始

步骤如下：

- 1) 项目发起人按照项目管理中心（PMO）的流程指定项目经理，这个部分操作流程如图16-1所示。
- 2) 项目经理在RPM客户端底部的“新步骤”行中收到通知，如图16-2所示，然后做出响应，选择“通知已接收”，项目正式由项目经理负责管理。
- 3) 项目经理在项目描述中把本项目涉及的一些（网上银行应用、电子银行内部管理、个人信贷管理等）应用信息作为时间代码加入到本项目中，此时间代码供项目组成员报工时使用。

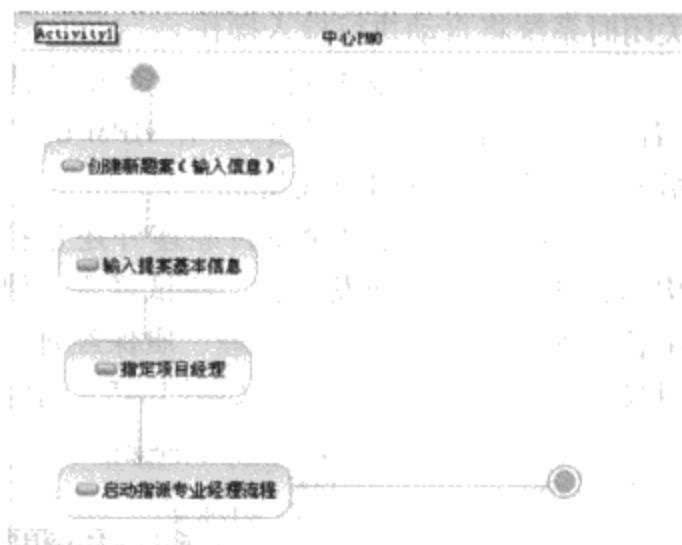


图16-1 指定项目经理流程

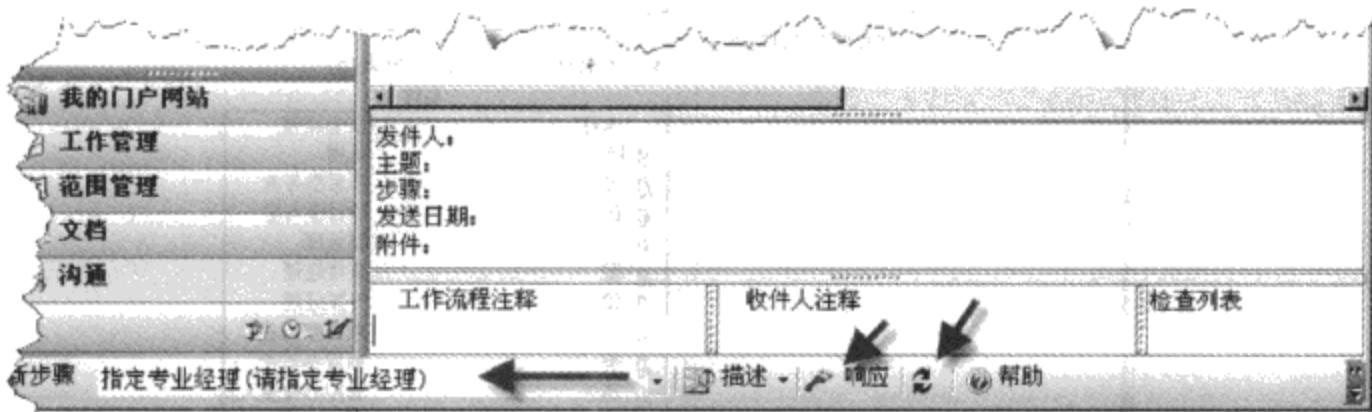


图16-2 接收通知

- 4) 项目经理登录RPM后选择“工作管理”视图，确认已选择“WBS计划→所有”过滤器。
- 5) 项目经理选中需要定义的项目，单击或选择客户端右上角的“描述”打开项目描述视图，选择“项目经理视图-快速状态”，然后在描述视图的右上角选择“检出”。如图16-3所示，维护项目基本信息。其中，项目的版本、项目编号、功能点等信息已经在下达任务前由项目管理办公室（PMO）录入。

This screenshot shows the 'Project Description View - Quick Status' (项目经理视图-快速状态) in the RPM system. The title bar reads '项目经理视图-快速状态' and '网上银行个人贷款项目贷款申请部分'. The main area is divided into several sections:

- 标识** (Identification):
 

名称	网上银行个人贷款项目贷款申请部
版本计划	2008年4月份版本
总行项目编号	2006584
引用号	2007879
是否能达到系统	<input checked="" type="checkbox"/>
交付后质量预估	0.00
已发布	到团队
估算开始日期	2007-06-19
估算完成	2008-05-15
状态	版本投产
块省任务类型	固定工期
工作量驱动	
- 项目功能点** (Project Function Points):
 

项目功能点估算	0.00
项目功能点概算	669.00
项目功能点预算	529.44
项目功能点决算	529.44
- 项目DELPHI评估** (Project DELPHI Assessment):
 

项目工作量估算(人日)	0.00
项目工作量概算(人日)	0.00
项目工作量预算(人日)	0.00
项目工作量决算(人日)	0.00

图16-3 项目描述视图

6) 将“时间代码”设为全屏后为项目录入时间代码,如图 16-4 所示,时间代码包括项目涉及的应用及项目涉及的各工作阶段,检入新修改后关闭项目描述视图或直接关闭项目描述视图,关闭时系统将自动执行检入操作。

7) 项目经理根据目前人力资源的情况(各部门上报的项目组成员),在 RPM 的“人员配备”中分配项目组成员。

8) 项目组成员中除项目经理外还包括架构师、开发经理、开发组长、测试经理、测试组长、配置管理员、环境经理、质保经理、项目参与者(开发人员)等,如图 16-5 所示。

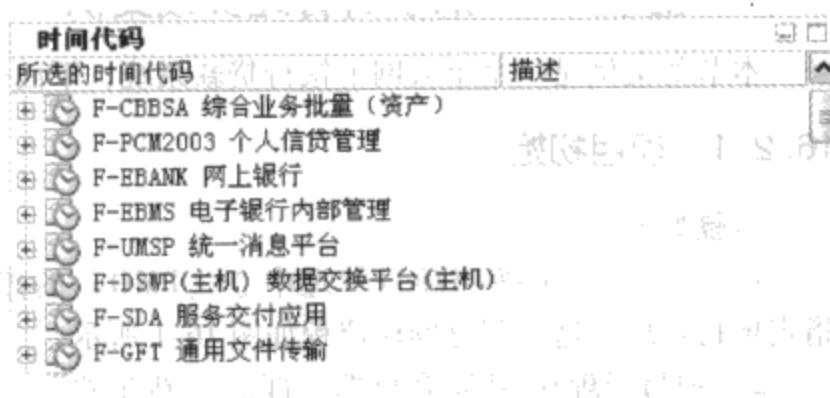


图 16-4 录入时间代码

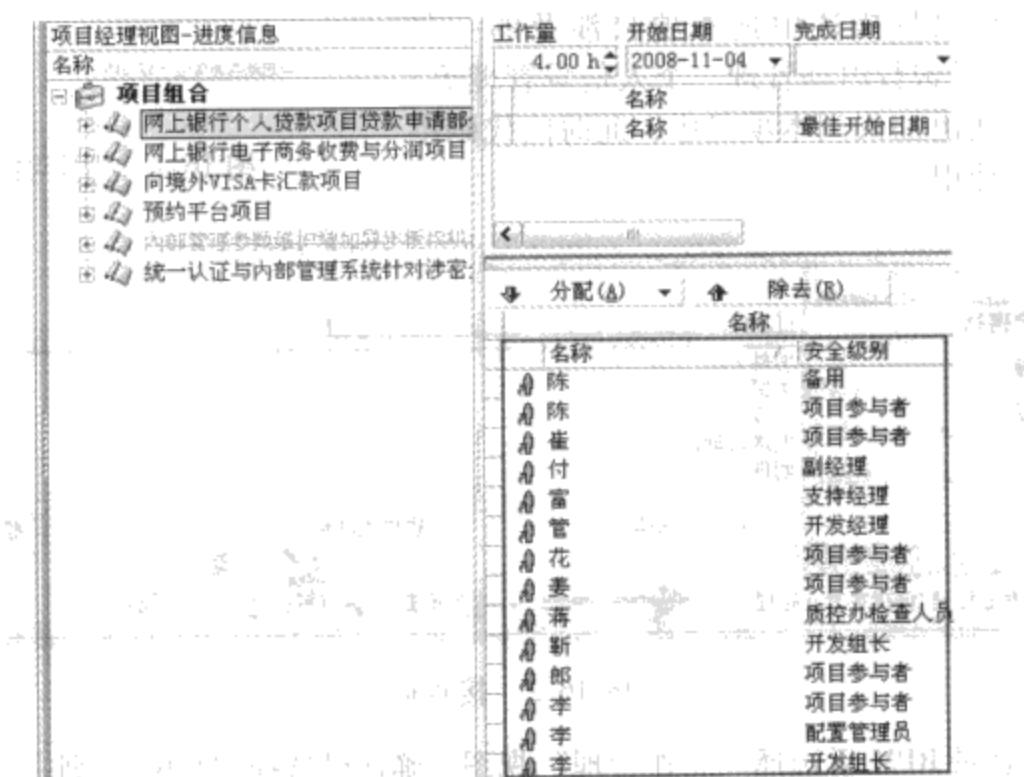


图 16-5 项目组成员

9) 人员分配后,项目经理回到项目提案页面,选中项目,右键沟通,选择“分配文档库权限”,此时项目组成员则能使用自己的用户名登录看到此项目,并在将来可以填报工时。

## 16.2.2 制定项目计划

步骤如下:

1) 由于本项目使用迭代模型,则项目经理需要在“工作管理”视图中选择 PMO 新创建提案后,点击客户端右上角的“模板”,在进行第一次计划会签之前,项目经理只需要为提案创建“定义阶段计划”。根据项目的开发模型,从模板中拖曳“项目定义阶段”、“项目内部例会、会议纪要”及“项目周报”计划到提案中。如图 16-6 所示。

2) 当项目前期计划会签通过且项目定义阶段工作完成后,项目经理需要依据项目定义阶段的项目计划时间点,在定义阶段计划的基础上创建完整的项目计划,方法同上述前期计划的制定,从模板中将“开发阶段计划”/“交付后计划”工作包拖拽到提案中。

关闭“模板”视图,在项目的右键菜单中选择“全部展开”或按 Shift + \* 展开项目计划,对项目任务进行调整。如果要删除任务,可以直接按 Delete 键或选择客户端右上角的删除图标。

3) 按照项目管理过程中的实际情况,项目经理通过两种方式来增加新的任务或子任务。

a) 拖拽的方式从客户端顶部将相应的图标拖放到项目计划的合适位置。



图 16-6 拖曳模板到提案中

b) 点中项目右键选中新增后点击任务或其他对象类型。

4) 为了能自动对任务与任务之间建立依赖关系，要双击该任务或选择任务后按下客户端右上角的“描述”，以此方法修改每个任务属性，通过任务的描述视图定义每个任务的开始、结束时间以及任务约束，如图 16-7 所示。

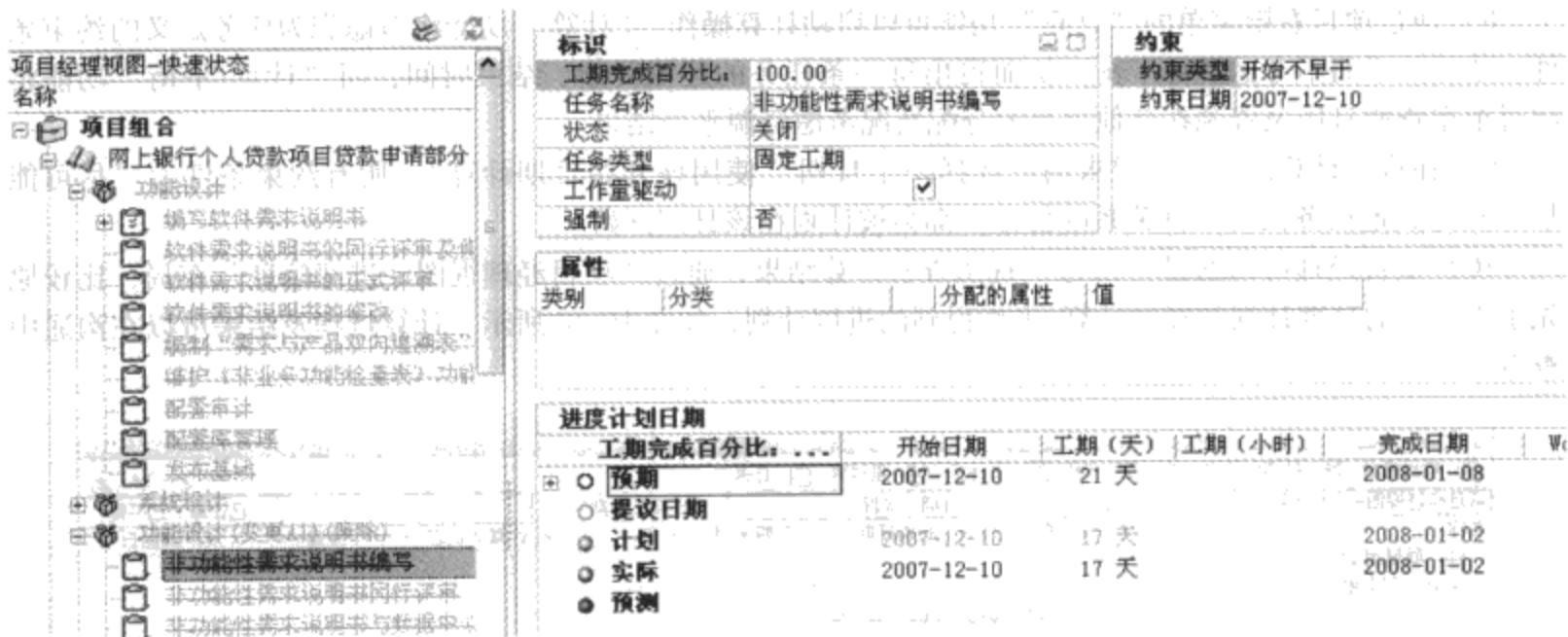


图 16-7 定义任务之间的依赖关系

5) 缺省情况下，所有任务的“约束类型”均为“尽可能早”，但如果项目经理手工调整了任务的开始时间而不更改“约束类型”后检入该更改，则“约束类型”将自动更改为“开始不早于”，如图 16-8 所示，约束日期则变为您录入的任务开始时间。如果手工调整了任务的结束时间而不更改“约束类型”后检入更改，则“约束类型”将自动更改为“完成不早于”，约束日期则变为您录入的任务结束时间。

6) 项目经理需要为任务定义“完成 - 开始”类型的依赖关系。操作如下：

- 关闭任务描述视图，在“工作管理”视图中按下客户端右上角的“依赖关系”按钮。
- 先用鼠标选中第一个任务，再将鼠标拖到第二个任务上，待图标出现后松开鼠标。
- 重复上述操作可以为多个任务快速创建“完成 - 开始”类型的依赖关系。

7) 定义完任务的约束条件和依赖关系之后，项目经理在项目上运行“计算”功能，计算考虑了依赖和任务约束之后的项目进度情况；“计算”及“计算/平衡”功能将为项目计算关键路径，该关键路径可以通过“任务过滤器→关键任务”查看，如图 16-9 所示。

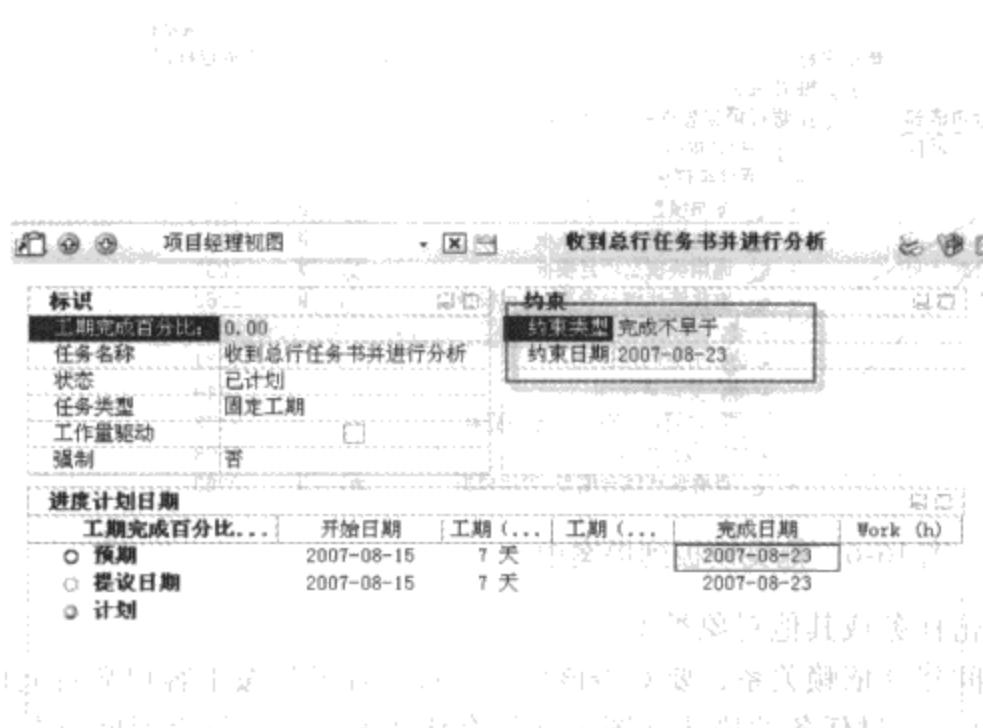


图 16-8 定义约束类型

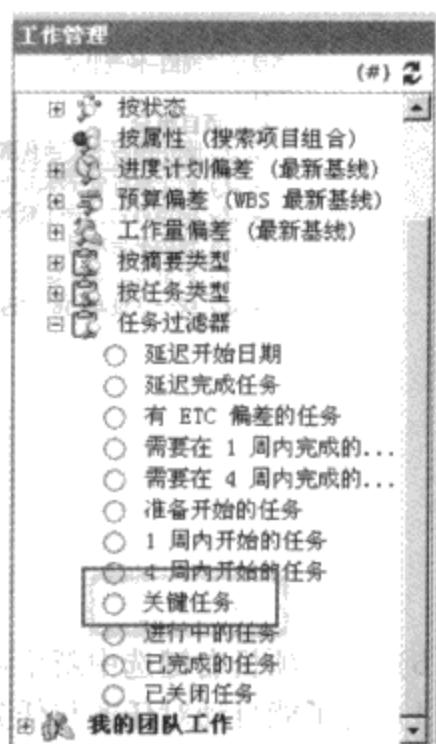


图 16-9 确定关键路径

8) 通过项目右键菜单的“计算”功能可以启动计算操作。“计算”功能将考虑您为任务定义的约束条件、依赖关系而不考虑资源平衡，从而得出每一条任务的建议开始、结束时间，而“计算/平衡”功能将同时根据资源日历及资源在其他项目的分配情况考虑资源平衡结果。

9) 在进行计算之前，会要求用户选择一个日期“使用该开始日期计算”。所有约束条件为“尽可能早”且与其他任务没有依赖关系的任务，都会被计划在该日期开始。

10) 项目经理可以通过“沟通”视图查看计算结果，通过“项目经理视图-进度信息”布局，比较原先手工定义的进度计划与经过计算/平衡得到的进度计划，如图 16-10 所示。计算过程及结果可以在沟通中查看。



图 16-10 进度计划

- 11) 计算完后，项目经理选择“复制计算结果”。
- 12) 确定 WBS 以后，项目经理要为任务分配资源，只有被分配到某个任务上的项目组成员，才可以在该任务上填报工时。

13) 项目经理登录 RPM 后，进入“工作管理”视图，首先选择要分配资源的任务，然后选择“人员配备→搜索/分配”，如图 16-11 所示。

14) 点击后，项目经理可以对需要分配工时的人员进行搜索。RPM 将在“结果集”页面显示搜索的结果，如图 16-12 所示。如果搜索时选择不考虑资源在其他任务的分配情况，项目经理将通过查看“可用时间段”，了解该资源在同一时间段是否已被分配。如果搜索时选择考虑资源在其他任务的分配情况，则系统将自动把“最佳开始时间”和“最佳完成日期”在其他任务的完成时间基础上顺延。

名称	工作...	日期(实际)		日期(预期)		状态	开始日期	完成日期	工期	工时	计算/状态
		日期	时间	日期	时间						
网上银行个人贷款项目贷款申请部分	版本...	2007-06-19		142.65	d 20...	20...	2...				
功能设计	需求	2007-10-08	2007-10-23	43.40	d 20...	30...	1...				
系统设计	需求	2007-10-15	2007-11-02	35.75	d 20...	30...	1...				
功能设计:实现部分(需求)	需求	2007-06-19		45.65	d 20...	20...	1...				
功能设计:实现部分(需求)	需求	2008-01-09		21.45	d 20...	20...	1...				
功能设计:实现部分(需求)	需求	2007-12-10	2008-01-09	31.65	d 20...	20...	1...				
已计划	2008-03-01			69.52	d 20...	20...	1...				
已计划	2008-03-20			46.59	d 20...	20...	6...				
已计划	2008-03-20			8.60	d 20...	20...	1...				
已计划	2008-03-31			11.82	d 20...	20...	1...				
待确认	2008-04-08	2008-04-15		5.78	d 20...	20...	3...				
单体测试(自测)	已计划			20...	d 20...	3...					
更新“需求与产品双向追溯表”	已计划			20...	d 20...	0...					
配置审计	已计划			20...	d 20...	1...					
配置库管理	已计划			20...	d 20...	1...					
集成测试版本和系统测试版本的制作	已计划			20...	d 20...	1...					
待确认	2008-03-31	2008-04-17		12.57	d 20...	20...	1...				

图 16-11 人员配备

名称	工作量	开始日期	完成日期	工作量类型	最佳可用性					
					分配人	预算	技能数据透视表	最佳开始日期	最佳完成日期	
林					林					
刘					刘					
付					付					
俞					俞					
肖					肖					
陈					陈					
周					周					
叶					叶					
李					李					
黄					黄					
顾					顾					
陈					陈					

分配(A)		删除(B)		任务负责				
名称	开始日期	完成日期	工作量	名称	开始日期	完成日期	工作量	名称
单体测试	2008-04-15	2008-04-17	0.57 d 4.00	4.00				
周长胜	任务负责	2008-04-15	2008-04-17	0.57 d 4.00	4.00	19		

图 16-12 人员搜索

15) 如果需要了解搜索到的资源的详细信息，项目经理可以在资源的右键菜单中选择“打开资源记录”，查看该资源的基本信息及在其他项目的被分配情况。如果需要了解该资源在某段时间的忙闲状况，项目经理可以选择“打开资源日历”，如图 16-13 所示，查看该资源的任务分配情况。

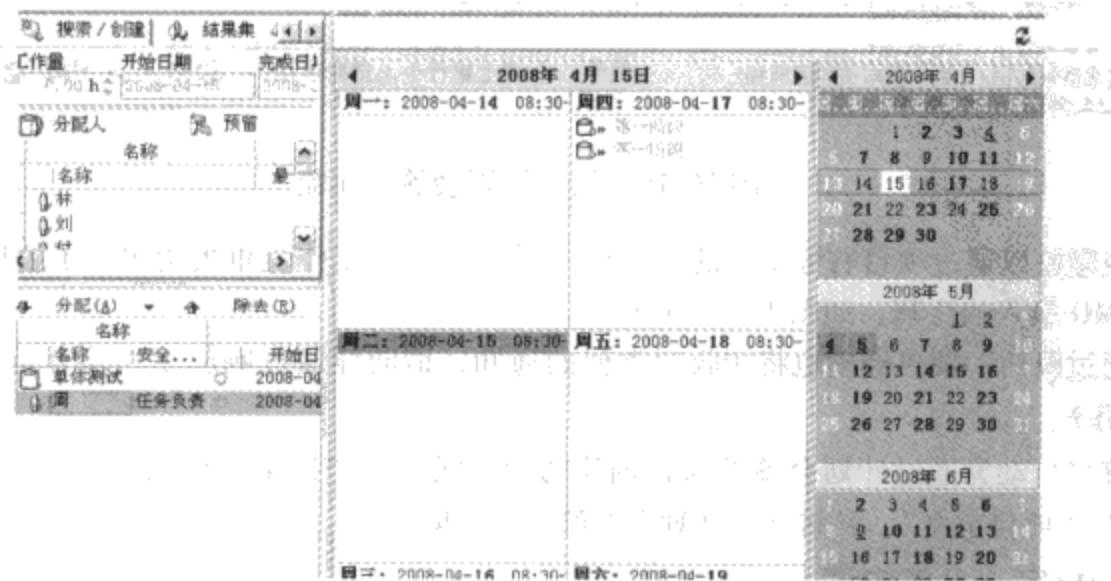


图 16-13 资源日历

16) 如果要将原先已分配到项目上的资源分配到新的任务上,项目经理也可以在人员配备视图的“项目资源”页面中找到相应资源,快速的将该资源分配给其他任务选中要分配的人,填上工作量,双击人员名称,工时即已分配。对于分配错误的,点击分配过的人员,点击除去即可,如图 16-14 所示。

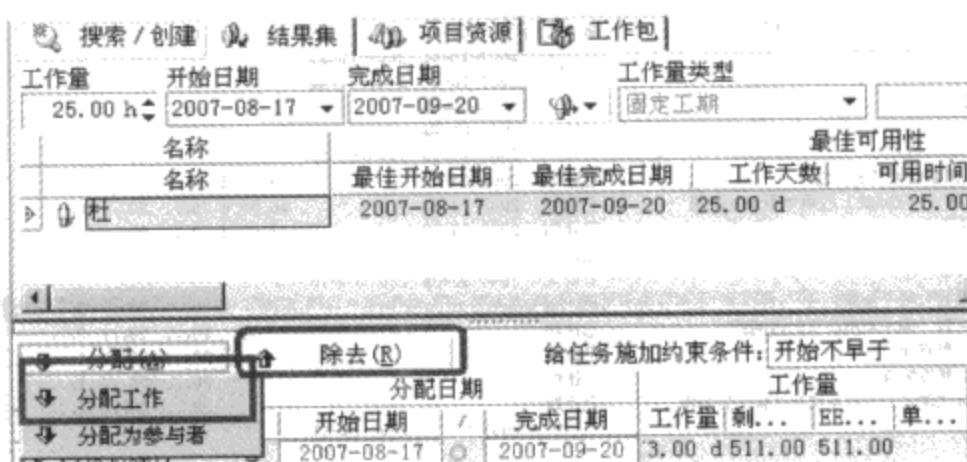


图 16-14 任务资源分配

### 16.2.3 计划会签

#### 1. 初次计划会签申请

步骤如下:

- 1) 启动计划会签申请,选中网上银行个人贷款项目,从右键菜单中选择“沟通→计划会签”。
- 2) 启动该流程后,项目经理将在 RPM 客户端底部的“新步骤”行中收到通知,如图 16-15 所示,请响应该步骤,选择计划会签的类别。

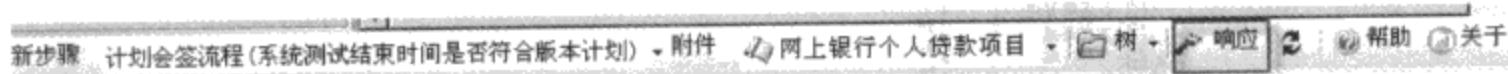


图 16-15 会签通知

- 3) 项目会签后,各部门会收到计划会签的邮件,项目组各专业经理分析项目计划是否可行,若可行,会签同意,如果各部门均同意,则经过中心 PMO 审查,无规范问题后,会签通过,项目经理即会收到项目会签完成的邮件。若某个部门不同意项目计划,会签不同意,则项目计划被驳回。
- 4) 若项目经理误启动了“计划会签”流程,可通过响应上述工作流并选择“取消操作”,中止工作流。

- 5) 当计划会签申请在任一个环节被否决时,项目经理都将接收到“申请驳回通知”,此时项目经理需要响应“驳回”通知,选择“通知已接收”结束该工作流,如图 16-16 所示。

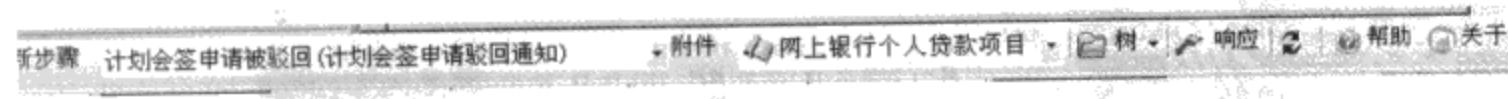


图 16-16 结束计划会签工作流

- 6) 项目经理可以通过项目右键菜单的“历史→过程”查看工作流审批历史,了解申请被驳回时部门 PMO 或中心 PMO 录入的注释,如图 16-17 所示。
- 7) 会签通过以后,项目经理也将接收到类似的通知,请同样通过响应工作流并选择“通知已接收”的方式结束工作流。
- 8) 工作流将自动为通过计划会签的项目创建初始基线,将“提案”转变为“项目”,将项目状态由“立项”转换为“项目定义”,并将项目计划发布给项目团队。

#### 2. 项目目标定义会签申请

项目经理完成项目目标定义书以后,项目经理将 POD (项目目标定义书) 上传到 RPM 中并为该文档

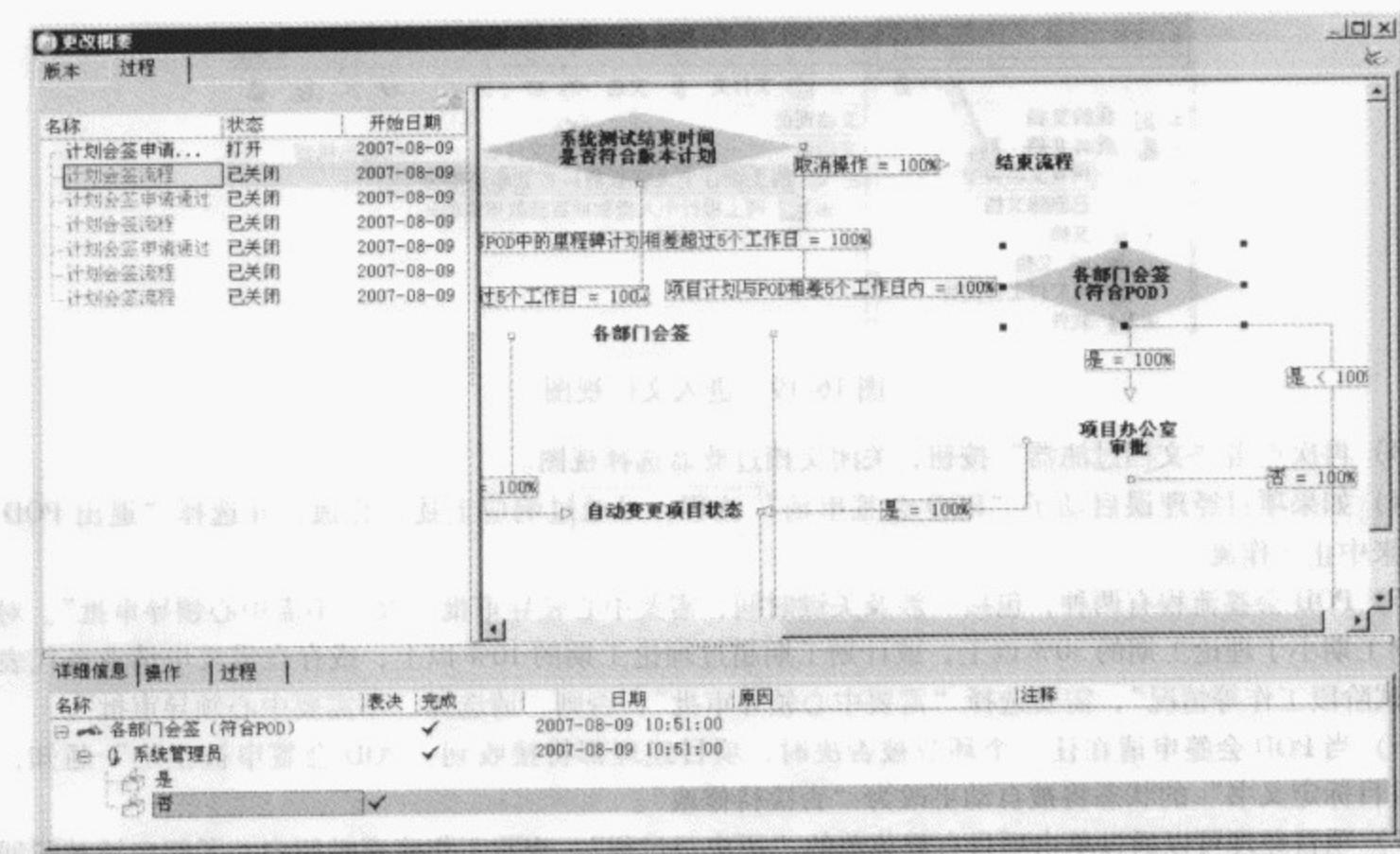


图 16-17 查看工作流审批

启动“POD 会签申请”流程，POD 需经正式审批通过，项目计划才正式生效。“POD 会签申请”流程如下，如图 16-18 所示。

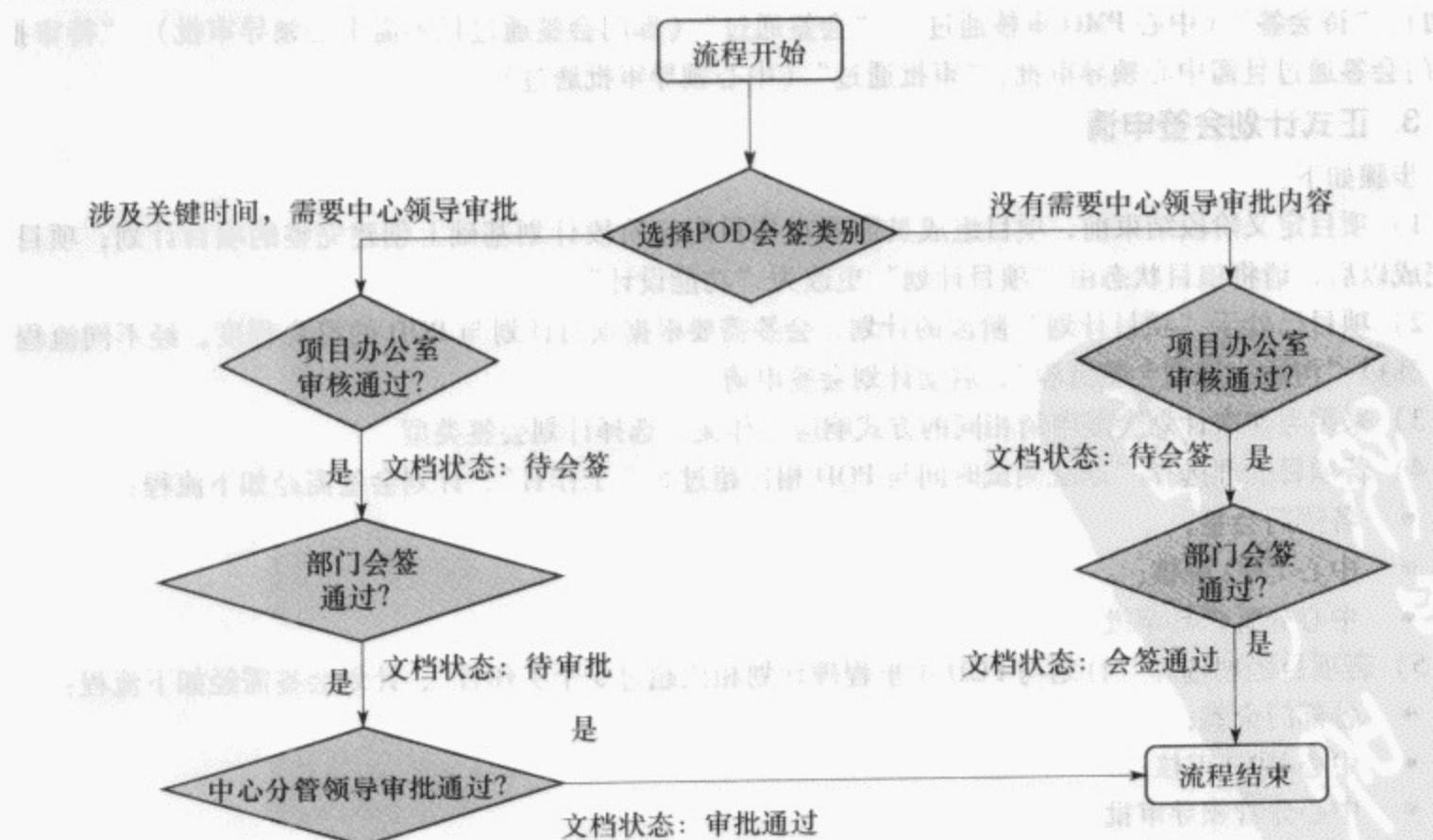


图 16-18 POD 会签申请流程

步骤如下：

- 1) 首先需要将 POD 上传到 RPM 中，选择需要创建项目目标定义书的项目，进入文档视图。
- 2) 选择“文档过滤器”，打开文档过滤器选择视图，确认“文档过滤器→所有文档→所有文档类型”已选择，如图 16-19 所示。

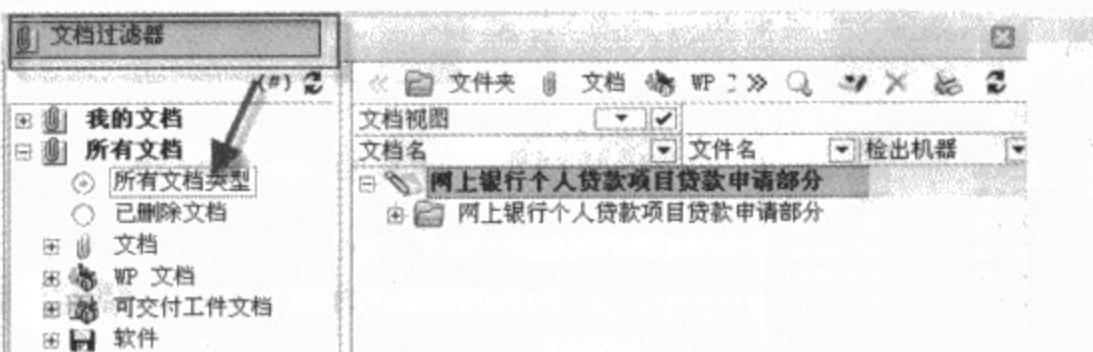


图 16-19 进入文档视图

- 3) 再次点击“文档过滤器”按钮，关闭文档过滤器选择视图。
- 4) 如果项目经理误启动了“POD 会签申请”流程，可通过响应上述工作流，并选择“退出 POD 流程”来中止工作流。
- 5) POD 会签流程有两种，包括“涉及关键时间，需要中心领导审批”及“不需中心领导审批”。对于“计划工期小于理论工期的 30% 以上，或计划工期超过理论工期的 10% 以上，或存在需要申请业务代表参与测试阶段工作等情况”，需要选择“需要中心领导审批”，否则，请选择“不需要中心领导审批”。
- 6) 当 POD 会签申请在任一个环节被否决时，项目经理都将接收到“POD 会签申被拒绝”通知，且“项目目标定义书”的状态将被自动更改为“否决待修改”。
- 7) 项目经理可以通过单击项目右键菜单的“历史→过程”，查看工作流审批历史，了解申请被驳回时部门 PMO 或中心 PMO 录入的注释。请响应“拒绝”通知，选择“通知已接收”结束该工作流。
- 8) 会签通过以后，项目经理也将接收到类似的通知，请同样通过响应工作流并选择“通知已接收”的方式结束工作流。
- 9) 工作流将自动根据审批状态更新该项目目标定义书的状态，可能的状态包括“待审核”（工作流已启动）、“待会签”（中心 PMO 审核通过），“会签通过”（部门会签通过且不需中心领导审批），“待审批”（部门会签通过且需中心领导审批），“审批通过”（中心领导审批通过）。

### 3. 正式计划会签申请

步骤如下：

- 1) 项目定义阶段结束前，项目组成员需要在项目定义阶段计划基础上创建完整的项目计划；项目计划完成以后，请将项目状态由“项目计划”更改为“功能设计”。
- 2) 项目已处于“项目计划”阶段的计划，会签需要根据项目计划与 POD 的符合程度，经不同流程审批，选择“沟通→计划会签流程”，启动计划会签申请。
- 3) 采用与初次计划会签申请相同的方式响应工作流，选择计划会签类型。
- 4) 若项目经理选择“系统测试时间与 POD 相比超过 5 个工作日”，计划会签需经如下流程：
  - 各部门会签；
  - 中心 PMO 审核；
  - 中心分管领导审批。
- 5) 若项目经理选择“计划与 POD 中里程碑计划相比超过 5 个工作日”，计划会签需经如下流程：
  - 各部门会签；
  - 中心 PMO 审核；
  - 中心分管领导审批。
- 6) 若项目经理选择“项目计划与 POD 相差 5 个工作日内”，计划会签需经如下流程：
  - 各部门（不含总工办）会签；
  - 中心 PMO 审批。
- 7) 当流程任一步骤有人员否决该会签申请时，流程即否决关闭，项目经理将接收到“申请驳回通知”。
- 8) 项目经理可以通过项目右键菜单的“历史→过程”查看工作流审批历史，了解申请被驳回时部门 PMO 或中心 PMO、中心分管领导录入的注释。

- 9) 请响应“驳回”通知，选择“通知已接收”结束该工作流。
- 10) 会签通过以后，项目经理也将接收到类似的通知，请同样通过响应工作流并选择“通知已接收”的方式结束工作流。
- 11) 工作流将自动为通过计划会签的项目重新创建初始基线，将项目状态由“项目计划”转换为“功能设计”，并将新的项目计划发布给项目团队。

#### 16.2.4 项目范围管理

步骤如下：

- 1) 在项目初期，项目经理就要在 RPM 中创建项目依赖，此依赖保持与 POD 中描述一致，项目经理在“工作管理”视图中，选择需要创建依赖记录的项目，然后选择“范围管理”，进入“范围管理”视图，如图 16-20 所示。

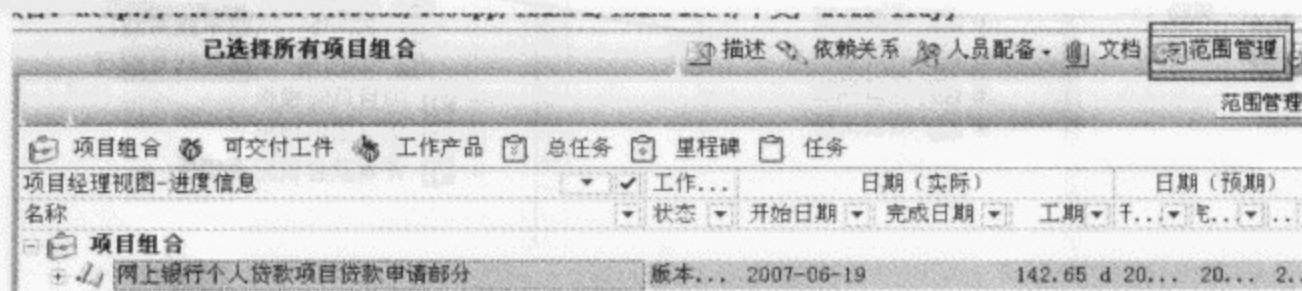


图 16-20 范围管理

- 2) 选择“范围管理”视图中的“范围过滤器”，进入“范围过滤器”选择视图，确保“所有范围元素→所有范围元素”过滤器已选中，然后重新点击“范围过滤器”，关闭过滤器视图，如图 16-21 所示。

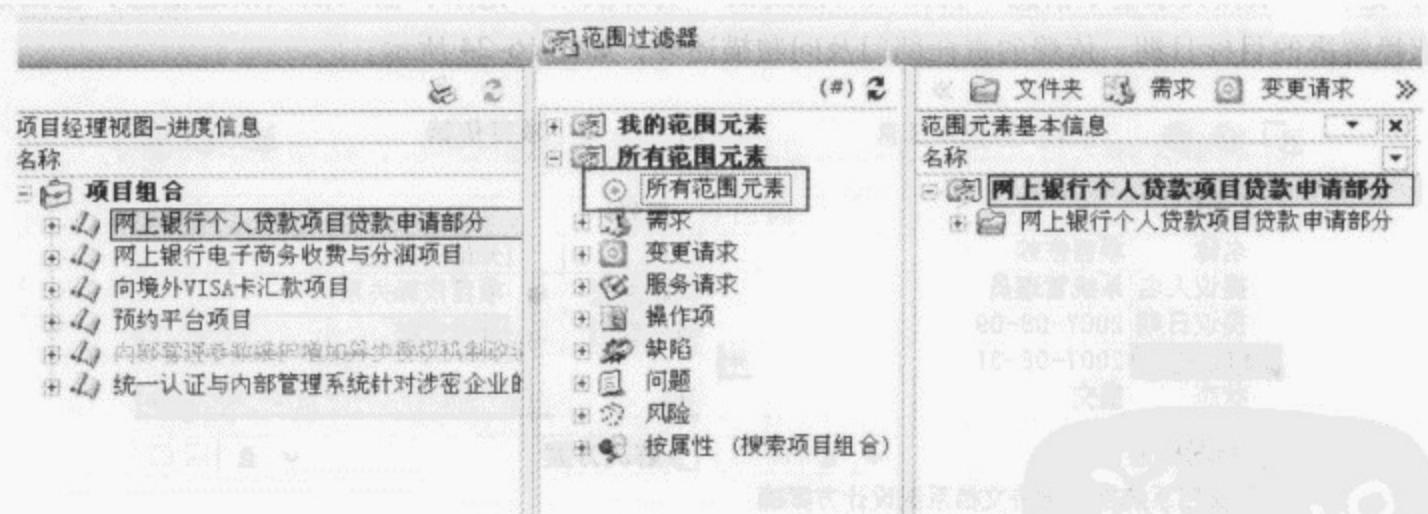


图 16-21 范围过滤器

- 3) 项目经理首先为项目依赖创建一个目录，以便更好的组织项目内的范围元素，如图 16-22 所示。

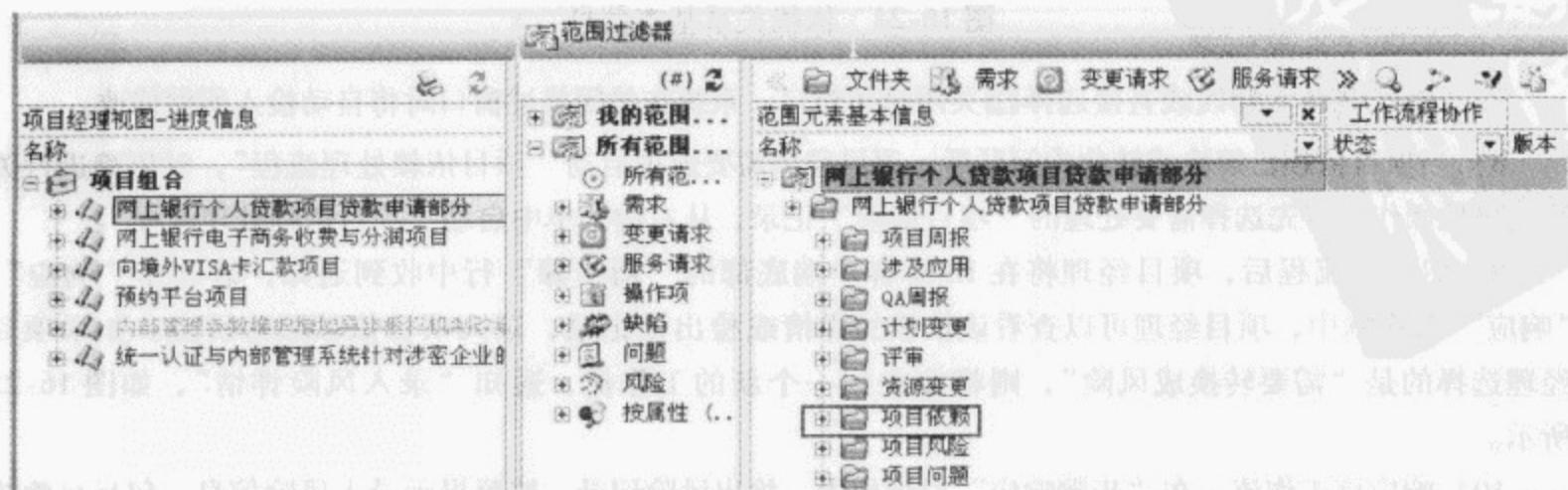


图 16-22 项目依赖

4) 选择“范围管理”视图右上角的“模板”，打开范围元素模板，进入“项目问题模板”目录，将“项目依赖”拖拽到项目根目录或新创建的文件夹下，如图 16-23 所示。

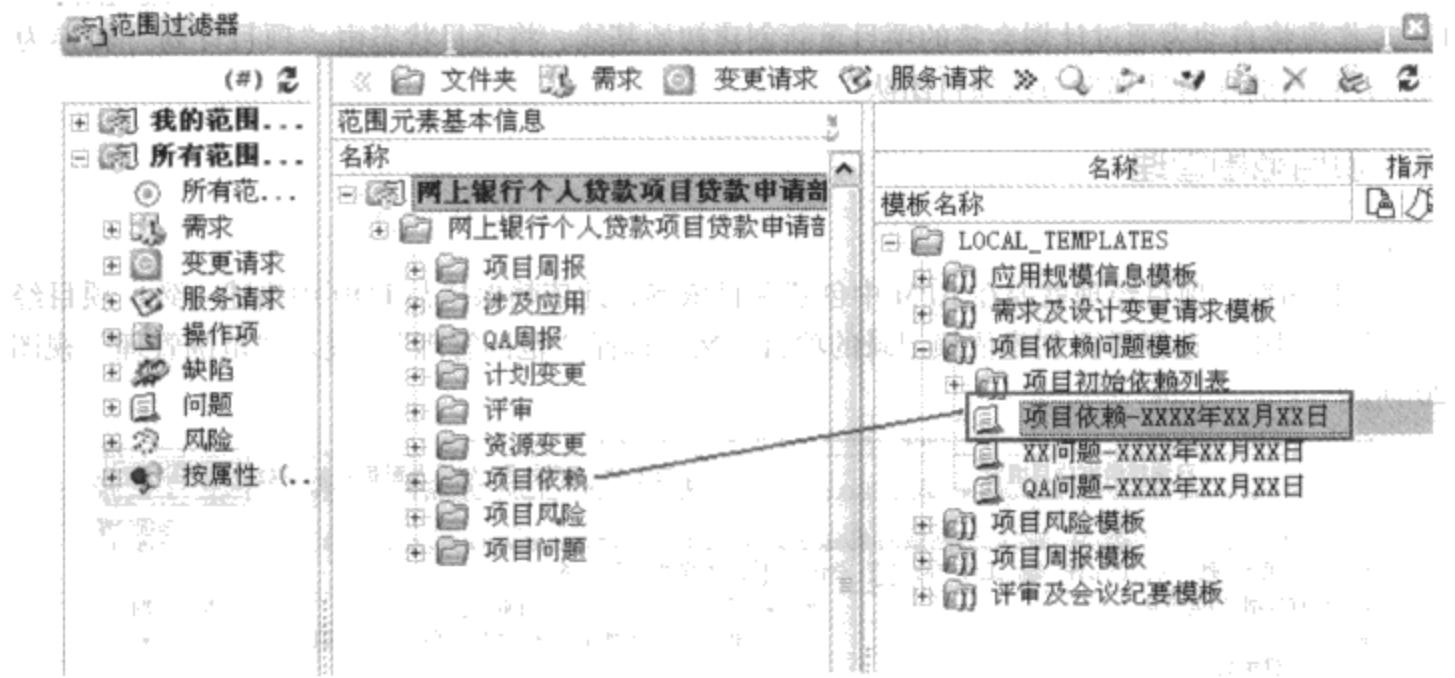


图 16-23 项目依赖创建

5) 关闭“模板”视图，双击新创建的“项目依赖”或选择“项目依赖”后按“描述”进入  视图。

6) 选择“依赖关系基本信息”视图以后检出该“项目依赖”记录，输入依赖描述信息，包括依赖名称、依赖解决的目标日期、依赖的责任部门及问题描述等，如图 16-24 所示。

图 16-24 依赖关系基本信息

7) 选择  检入修改或直接选择  关闭描述窗口，系统在关闭描述窗口时将自动检入所做修改。

8) 当项目依赖已解决或转化为问题后，项目经理需要通过启动“项目依赖处理流程”，对依赖进行关闭或转化操作。首先选择需要处理的“项目依赖”记录，从右键菜单中启动“项目依赖处理流程”。

9) 启动该流程后，项目经理将在 RPM 客户端底部的“新步骤”行中收到通知，请选择“响应”，“响应”对话框中，项目经理可以查看该依赖的详情或检出该记录，录入项目依赖跟踪历史。如果项目经理选择的是“需要转换成风险”，则将接收到一个新的工作流，通知“录入风险详情”，如图 16-25 所示。

10) 响应该工作流，在“步骤响应”对话框中，检出风险记录，按照提示录入风险信息，包括风险描述、风险分类、风险类别及风险矩阵，然后选择“详情已录入”，提交步骤响应。

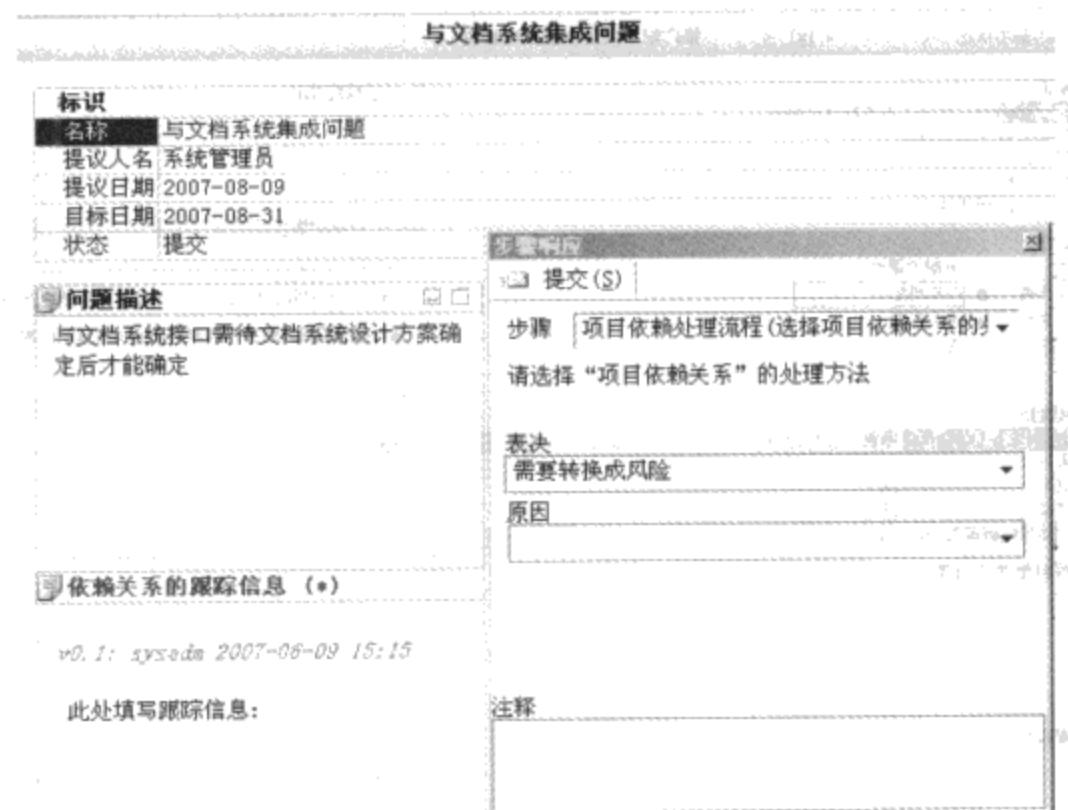


图 16-25 录入风险详情

11) 在“范围元素”视图中，可以看到原项目依赖记录的状态已自动更改为“已发生”或“已转换”（根据项目经理“选择依赖关系处理方法”时的选择确定），对于已转换为风险的依赖，RPM 已自动为该依赖创建了一个新的风险。

### 16.2.5 项目风险管理

步骤如下：

- 1) 项目的风险除了可以通过依赖转换外，还可以直接在 RPM 中创建，在“工作管理”视图中，选择需要创建风险记录的项目后选择“范围管理”，进入“范围管理”视图。
- 2) 选择“范围管理”视图中的“范围过滤器”，进入“范围过滤器”选择视图，确保“所有范围元素→所有范围元素”过滤器已选中后重新点击“范围过滤器”，关闭过滤器视图。
- 3) 项目经理先要为项目风险创建一个目录，以便更好的组织项目内的范围元素，选择“范围管理”视图右上角的“模板”，打开范围元素模板，进入“项目风险模板”目录，如图 16-26 所示，将“项目风险”拖拽到项目根目录或新创建的文件夹下。
- 4) 关闭“模板”视图，双击新创建的“项目风险”或选择“项目风险”，然后按“描述”进入描述视图，检出“项目风险”记录，输入风险描述信息，包括风险名称、目标解决日期、风险分类、类别、风险矩阵、建议对策等，选择检入修改或直接关闭描述窗口，系统在关闭描述窗口时将自动检入所做修改。

- 5) 当项目风险已发生、已规避或已转化为问题后，项目经理需要通过启动“项目风险处理流程”对风险进行关闭或转化操作。先选择需要处理的项目风险，从右键菜单中启动“项目风险处理流程”，启动该流程后，项目经理将在 RPM 客户端底部的“新步骤”行中收到通知，请选择“描述”，查看风险详情，并对风险处理流程进行响应，如图 16-27 所示。

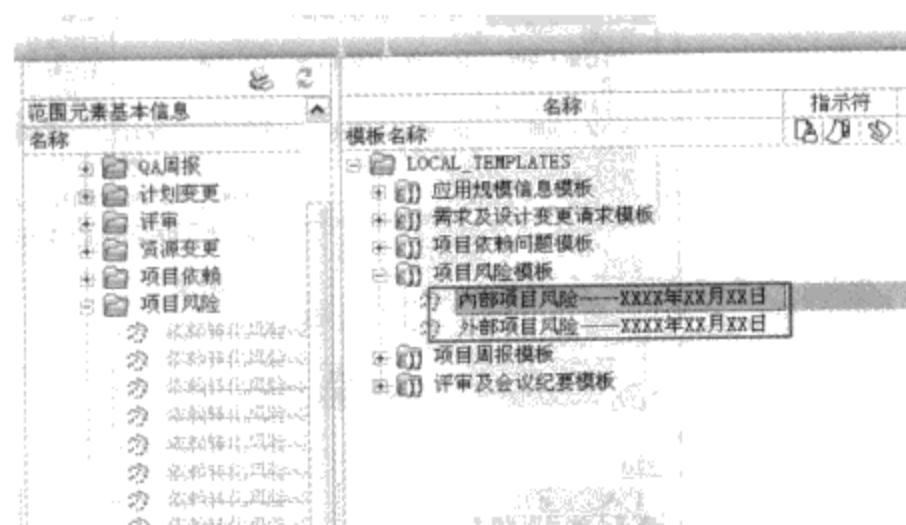


图 16-26 项目风险模板

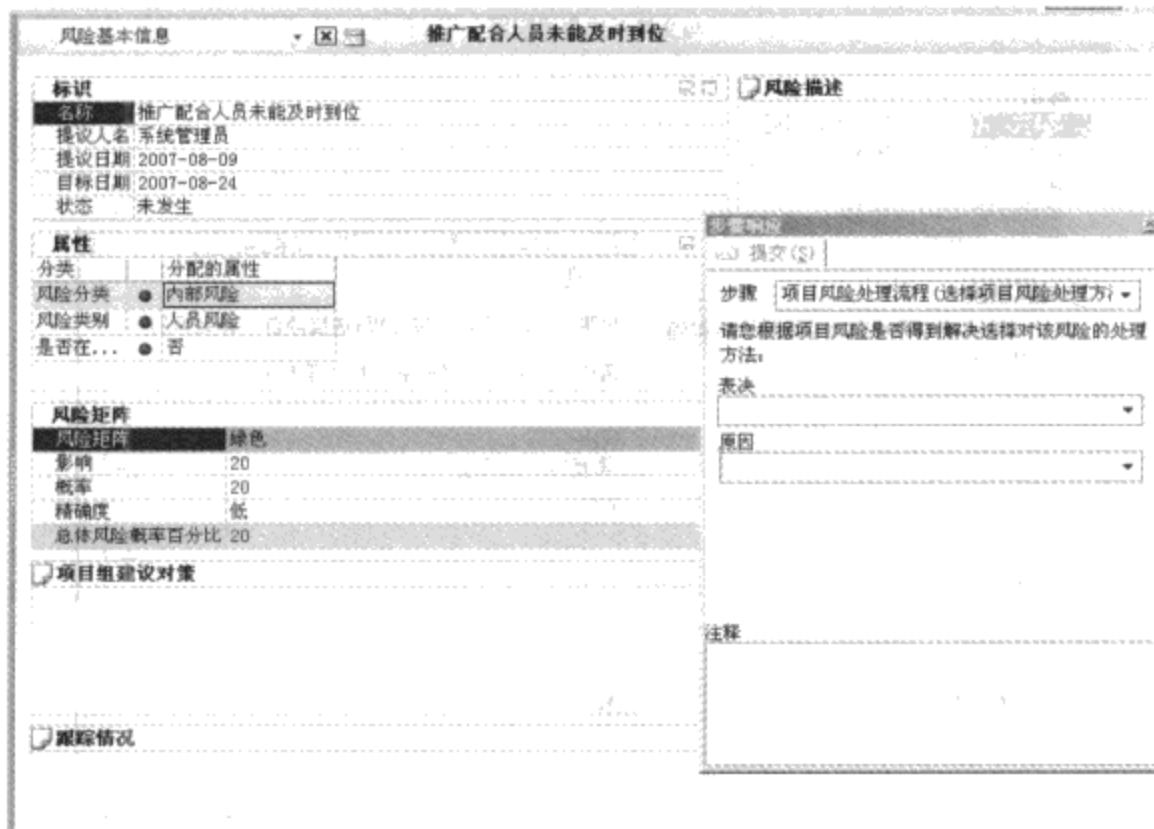


图 16-27 风险处理流程

- 6) 如果风险已发生，但未对项目造成主要问题，请在“表决”列表中选择“已经发生”；如果风险已规避，则选择“已经规避”；如果风险已发生且转化成项目问题，请选择“需要转换成问题”；如果您误启动了上述流程，则选择“取消操作”。
- 7) 如果项目经理选择的是“需要转换成问题”，则将接收到一个新的工作流，通知“录入问题详情”。
- 8) 响应该工作流，在“步骤响应”对话框中，检出问题记录，按照提示录入问题信息，包括问题优先级、等级、发生日期、目标解决日期、责任部门、问题分类、发生问题时项目所处阶段、问题描述等后选择“详情已录入”，提交步骤响应；如果问题分类为“协调”，则项目经理还要录入“协调事项属性”，如图 16-28 所示。

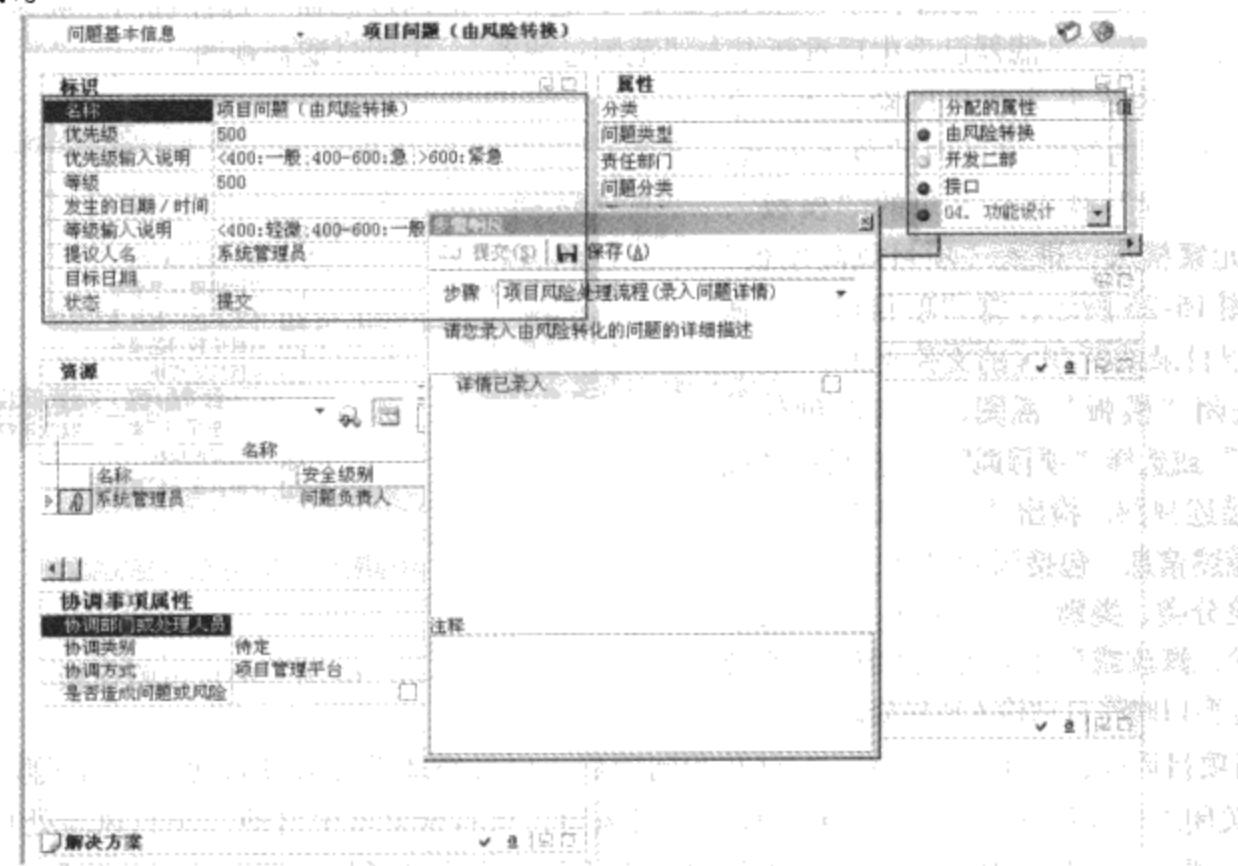


图 16-28 风险转换成问题

9) 在“范围元素”视图中，可以看到原项目依赖记录的状态已自动更改为“已规避”、“已发生”或“已转成问题”（根据项目经理“选择项目风险处理方法”时的选择确定），对于已转换为问题的风险，系统已自动为该风险创建了一个新的问题。

### 16.2.6 项目问题管理

步骤如下：

- 1) 在“工作管理”视图中，选择需要创建问题记录的项目后选择“范围管理”，进入“范围管理”视图。
- 2) 选择“范围管理”视图中的“范围过滤器”，进入“范围过滤器”选择视图，确保“所有范围元素→所有范围元素”过滤器已选中，然后重新点击“范围过滤器”，关闭过滤器视图。
- 3) 项目经理可以先为项目问题创建一个目录以便更好的组织项目内的范围元素（可选）。
- 4) 选择“范围管理”视图右上角的“模板”，打开范围元素模板。
- 5) 进入“项目问题模板”目录，将“项目问题”或“QA问题”拖拽到项目根目录或新创建的文件夹下，如图 16-29 所示。

文件夹		名称	指示符	快速状态	资源
	模板名称			工期 / 工作量	分配人 负责人
...	LOCAL_TEMPLATES				
...	应用规模信息模板				sysadm; sysadm
...	计划变更请求模板				sysadm; sysadm
...	需求及设计变更请求模板				sysadm; sysadm
...	项目问题模板				sysadm; sysadm
...	项目依赖			0.04 d	系统管理员; 系统管理员
...	项目问题			0.04 d	系统管理员; 系统管理员
...	QA问题			0.04 d	系统管理员; 系统管理员
...	项目问题(由风险转...) 提交			0.04 d	系统管理员; 系统管理员
...	项目风险模板				sysadm; sysadm
...	项目周报模板				sysadm; sysadm
...	评审需求模板				sysadm; sysadm

图 16-29 项目问题模板

- 6) 关闭“模板”视图，双击新创建的问题或选择该问题后按“描述”进入模板视图。
- 7) 选择“项目问题基本信息”视图，检出问题，录入问题信息，包括优先级、等级、目标解决日期、责任部门、问题分类、项目阶段及问题描述等。
- 8) 选择检入修改或直接关闭描述窗口，系统在关闭描述窗口时将自动检入所做修改。
- 9) 项目问题解决以后，问题责任人需要通过启动“项目问题关闭流程”对风险进行关闭或转化操作。
- 10) 首先选择需要处理的项目问题，从右键菜单中启动“项目问题关闭流程”。
- 11) 启动该流程后，问题责任人将在 RPM 客户端底部的“新步骤”行中收到通知，请选择“描述”，查看问题详情并对问题关闭流程进行响应。
- 12) 系统将在问题解决方案确认已录入后直接将问题状态更改为“解决问题”。

### 16.2.7 项目计划修改

项目计划修改步骤如下：

- 1) 如果修改项目计划，则需要从 RPM 中导出项目计划，在待导出项目计划的右键菜单中选择“导入/导出→导出”，按照导出向导的提示，完成导出操作，导出结束后，保存文档，此时 RPM 中的原项目计划将被锁定，请通过该项目的右键菜单将项目解锁。
- 2) 项目经理如需修改各任务的起止时间，需要通过修改资源的分配时间实现。找到需要修改起止时间的任务，进入“人员配备→搜索/分配”视图。
  - 如果任务还没有开始，可以直接修改任务的开始时间和任务的 EETC（预计未完工工作量，如果需要延长任务时间，请增加任务的 EETC）。
  - 如果任务已经开始，则只能修改任务的结束时间和 EETC，注意开始时间不能更改，因为该任务已经开始了。

3) 对于没有分配过资源的任务，项目经理可以直接修改该任务的开始结束时间、工期后给该任务分配资源。找到需要修改开始、结束时间或工期的任务，打开其描述视图，检出该记录，在“项目经理视图”布局中，找到“进度计划日期”并做相应的修改，确认修改无误后检入该记录。如图 16-30 所示。

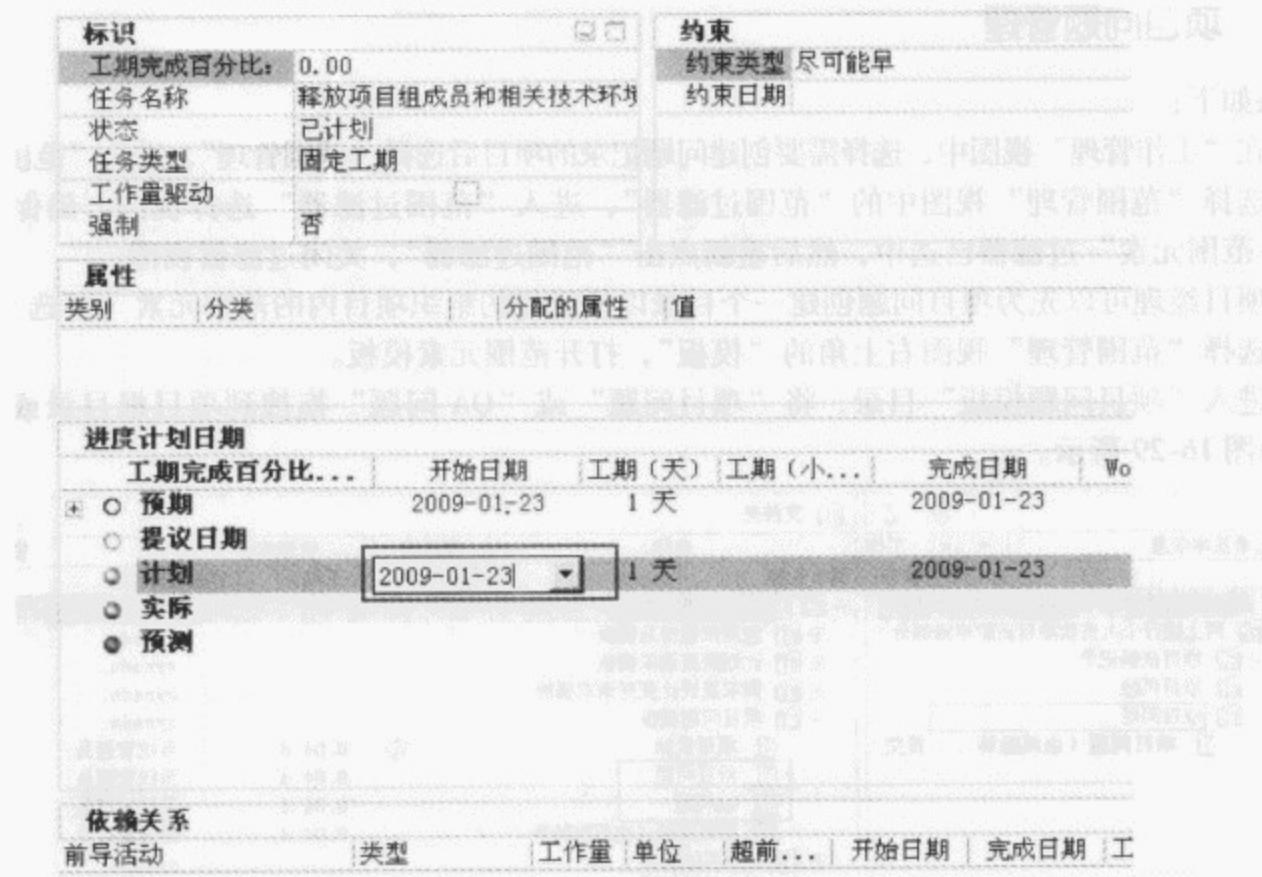


图 16-30 修改项目计划

### 16.2.8 计划变更管理

完整的计划变更流程如图 16-31 所示。

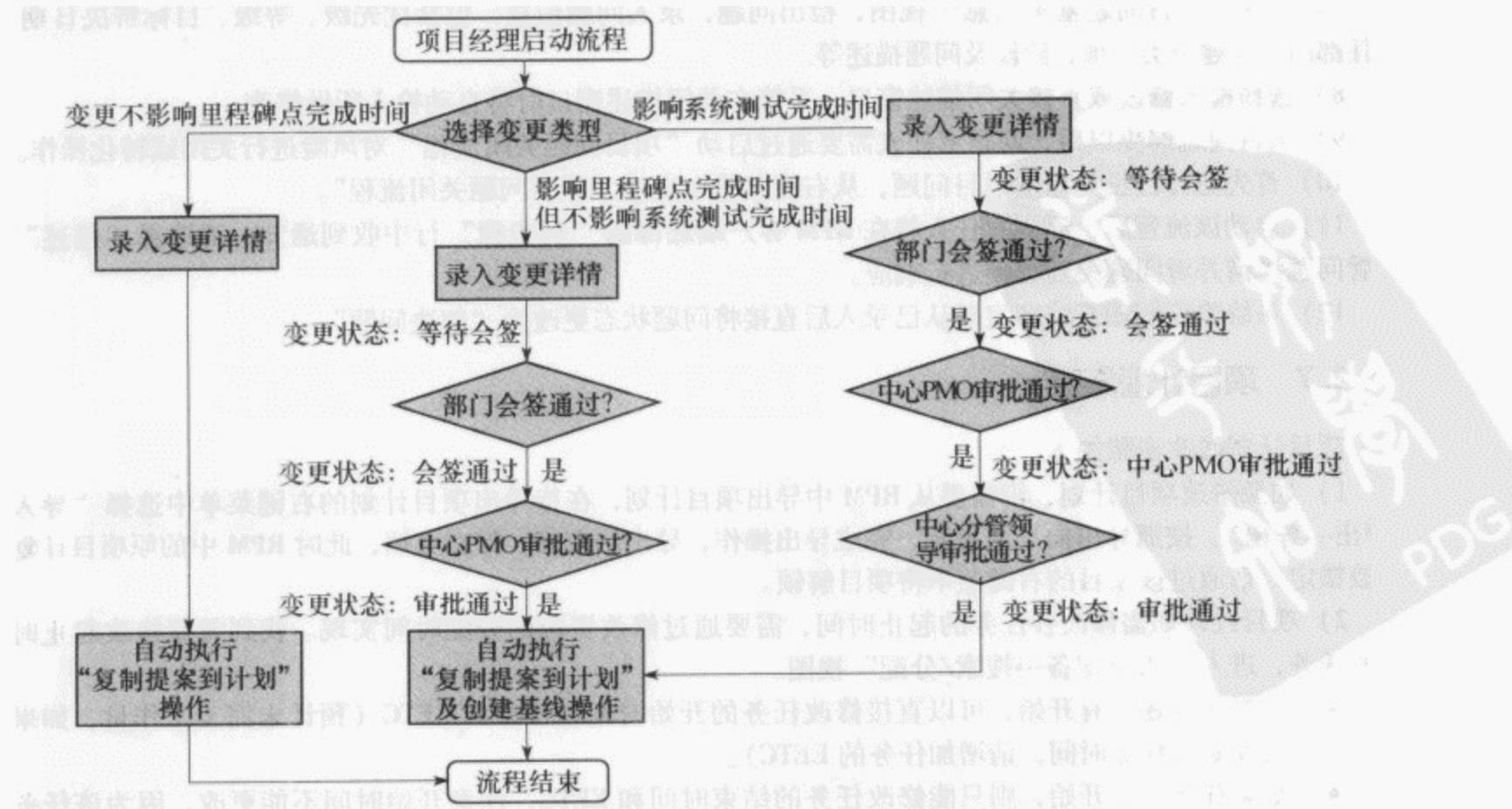


图 16-31 计划变更流程

步骤如下：

- 1) 要启动计划变更申请流程，请在项目计划的右键菜单中选择“计划变更申请”。
- 2) 启动该流程后，项目经理将在 RPM 客户端底部的“新步骤”行中收到通知，请响应该步骤，选择计划变更的类别：
  - 如果项目经理选择“只修改资源分配，不涉及里程碑点计划”，计划变更申请不需特殊审批即可生效。
  - 如果项目经理选择“涉及里程碑点计划的变化，但不影响项目结束时间”，计划变更申请需经如下流程：各部门会签→中心 PMO 审批。
  - 如果项目经理选择“影响系统测试结束时间”，计划变更申请需经如下流程：各部门会签→中心 PMO 审核→中心分管领导审批。
- 3) 选择完变更类型后，还需要输入变更详情。同样在客户端底部的“新步骤”中，可以看到一条新的工作流步骤“计划变更请求（请录入计划变更详情）”。
- 4) 选择“描述”，响应该工作流步骤，录入变更详情（对于不影响里程碑点的计划变更，项目经理可以不输入任何变更详情；对于其他变更，需要录入变更详情及变更对项目计划的影响）。
- 5) 最后，项目经理需要确认“计划变更详情已录入”并提交该步骤响应。
- 6) 系统将自动通过工作流把审批申请递交给相应部门的 PMO、中心 PMO 和中心分管领导；当流程任一步骤有人员否决该会签申请时，流程即否决关闭，项目经理将接收到“申请驳回通知”。
- 7) 项目经理可以通过项目右键菜单的“历史→过程”查看工作流是审批历史，了解申请被驳回时部门 PMO 或中心 PMO、中心分管领导录入的注释；请响应“驳回”通知，选择“通知已接收”结束该工作流。
- 8) 当工作流审批通过以后，项目经理及专业经理都将收到“计划变更申请（计划变更申请通过）”的通知；项目经理和专业经理需要选择“通知已接收”结束该工作流。
- 9) 进入项目的“范围管理”视图，可以看到系统已自动为刚才的变更创建了一条变更请求记录，该记录的状态与变更审批状态相同，可以通过查看这些变更记录了解项目计划变更历史，项目经理也可以选择为这些变更创建一个文件夹，把相应记录都拖拽到这个目录下，以更好的组织项目内的变更记录。
- 10) 要重新对项目计划进行计算，只需要选择要调整的项目计划，在项目的右键菜单中选择“计算”或“计算/平衡”即可。
- 11) 根据计算的结果，项目经理可以选择“复制计算结果”或者根据计算结果，手工调整受影响的任务计划。

### 16.2.9 需求/设计变更管理

步骤如下：

- 1) 在“工作管理”视图中找到要提交变更请求的项目，进入“范围元素”视图，在“范围元素”视图中，选择“模板”。
- 2) 项目经理可以先为需求或设计变更创建目录，以便更好的组织项目内的范围元素（可选）。
- 3) 根据要提交的变更申请类型（需求或设计），从“需求及设计变更请求模板”目录下拖拽合适的变更请求模板到范围元素视图根目录或新创建的文件夹下，如图 16-32 所示。
- 4) 关闭“模板”视图，双击新创建的变更请求，录入该变更请求的基本信息。
- 5) 在新创建的变更的右键菜单中，选择“沟通→变更审批申请”。
- 6) 如果项目经理或配置管理员不慎为一个不处于“提交”状态的需求/设计变更启动了“变更审批申请”，系统将给变更请求创建者发送“变更审批申请错误提示”，请响应该工作流，选择“通知已接收”。

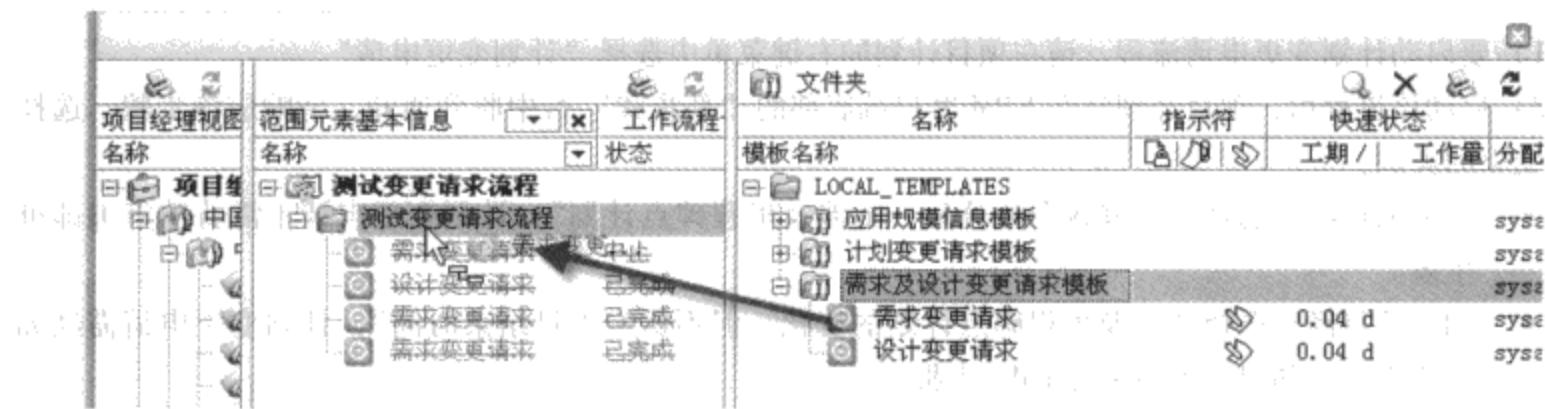


图 16-32 需求及设计变更请求模板

7) 审批通过以后, 系统将自动把变更的状态转换为“实施中”并为所有干系人发送“需求/设计变更申请已通过”的信息, 相关人员需要选择“通知已接收”关闭该工作流。如果审批被驳回, 系统将自动把变更的状态转换为“中止”, 同时干系人也将收到通知, 变更请求责任人需要创建新的变更请求后重新在新变更请求上提交审批申请。

8) 如果变更请求处于“提交”状态且变更创建者启动了“变更审批申请”, 项目经理、配置管理员和项目相关的专业经理(包括应用架构师、开发经理、开发组长、质保经理、测试经理、支持经理、测试组长)将在RPM客户端底部的“新步骤”行中收到通知。

9) 请选择“描述”对该申请进行审批。

10) 流程中任何一位审批人员拒绝该审批, 变更审批申请都将否决退出。

11) 在待关闭变更的右键菜单中, 选择“沟通→变更关闭申请”。

12) 如果项目经理或配置管理员不慎为一个不处于“实施中”状态的需求/设计变更启动了“变更关闭申请”, 系统将给变更请求创建者发送“变更关闭申请错误提示”, 请响应该工作流, 选择“通知已接收”。

13) 审批通过以后, 系统将把变更请求的状态转换为“已完成”并为变更请求责任人发送“需求/设计变更关闭申请已通过”的信息, 相关人员需要选择“通知已接收”关闭该工作流; 假如审批被拒绝, 变更请求责任人也将收到通知, 变更请求责任人需要修改变更信息后重新在同一个变更上提交“变更请求关闭申请”。

14) 如果该变更请求处于“实施中”状态, 项目经理在变更申请创建人启动变更关闭请求后即可接到新工作流申请, 请选择“描述”对该申请进行审批。

15) 如果项目经理同意该申请, 则变更请求状态将自动更改为“已完成”, 否则变更状态仍为“实施中”。

## 16.2.10 项目阶段交付物管理

步骤如下:

- 1) 选择与阶段标志物提交活动相关的里程碑点, 进入“文档”视图。
- 2) 为该里程碑点创建一个目录, 目录名称必须符合待提交的文档类型(如为“提交项目办审核POD”创建“项目目标定义书”目录)。
- 3) 在新创建的目录下(注意: 必须创建在该目录下, 而不能与该目录平行)创建一个新的URL。
- 4) 录入阶段交付物的名称后回车确定。
- 5) 在弹出的详情输入对话框中录入该交付物的URL。
- 6) 点击“确定”, 提交URL。
- 7) 在与阶段标志物提交有关的里程碑点上启动“阶段标志物归档申请”工作流。
- 8) 如果项目经理或配置管理员不慎为一个已经处于“关闭”状态的里程碑点启动了“阶段标志物归档申请”, 系统将给项目经理发送“归档申请错误提示”。
- 9) 审批通过以后, 系统将自动把里程碑点的状态转换为“关闭”并为项目专业经理发送“阶段交付

物已提交”的通知，相关人员需要选择“通知已接收”关闭该工作流；通过相应该工作流，项目相关人员可以查看到标志物的保存路径。

- 10) “阶段标志物归档申请”启动后，项目经理将接收到该工作流。
- 11) 项目经理需要检查工作流启动者录入的标志物保存路径是否正确，如果不正确，在响应工作流之前可以对该信息进行修改。
- 12) 如果确认阶段标志物已提交且路径正确，项目经理需要确认“阶段标志物已提交”并提交工作流步骤响应，工作流将自动关闭该里程碑。

### 16.2.11 监控阶段标志物提交情况

步骤如下：

- 1) 选择需要检查其阶段交付物交付情况的项目或项目群组，运行“项目组合→OLAP 数据透视表→工作管理→任务分析”。
- 2) 选择需要查询的里程碑点的日期范围，考虑提议日期还是计划日期，以基线还是预期日期做参考（报表将统计任务是否延迟完成，而判断任务是否延迟的标准可以是里程碑点当前的预期日期或最新基线的日期）。
- 3) 选择“阶段交付物检查”布局，“任务分析”数据透视表将列出所有计划在指定时间范围内完成的里程碑点，QA 可以按任意列对数据进行过滤和排序。
- 4) 如果质控办 QA 需要进一步查询需要提交项目交付物的里程碑点的详情，可以选择“WBS 计划→按任务类型→里程碑”过滤器，检查出所有与自己有关的项目中需要提交阶段交付物的任务，如图 16-33 所示。

名称	状态	负责人	分配人	开始日期	完成日期	工期
提交项目办审核POD	已计划	系统管...	系统管...	2007-07-31	2007-08-01	1.00
总体方案及POD上报总行	已计划	系统管...	系统管...	2007-08-01	2007-08-02	1.00
项目计划审批	已计划	系统管...	系统管...	2007-08-31	2007-09-03	0.50
业务需求分析说明书修改	已计划	系统管...	系统管...	2007-09-25	2007-09-27	2.00
非功能性需求说明书修改	已计划	系统管...	系统管...	2007-09-30	2007-10-09	2.00
软件需求说明书的修改	已计划	系统管...	系统管...	2007-10-10	2007-10-12	2.00
系统规格书的修改	已计划	系统管...	系统管...	2007-11-06	2007-11-08	2.00
程序规格书的修改	已计划	系统管...	系统管...	2007-11-02	2007-11-06	2.00
代码检查与修改	已计划	系统管...	系统管...	2007-11-19	2007-11-21	2.00
集成测试报告	已计划	系统管...	系统管...	2007-12-26	2007-12-28	2.00
测试报告定稿	已计划	系统管...	系统管...	2008-02-19	2008-02-21	2.00
提交项目办审核POD	已计划	中心PMO	中心PMO	2007-08-22	2007-08-22	1.00
总体方案及POD上报总行	已计划	中心PMO	中心PMO	2007-08-23	2007-08-23	1.00
项目计划评审	已计划	中心PMO	中心PMO	2007-09-21	2007-09-21	1.00
软件需求说明书的修订	已计划	中心PMO	中心PMO	2007-09-30	2007-09-30	1.00
系统规格书的修改	已计划	中心PMO	中心PMO	2007-10-25	2007-10-26	2.00
程序规格书的修改	已计划	中心PMO	中心PMO	2007-10-17	2007-10-19	2.00
代码检查与修改	已计划	中心PMO	中心PMO	2007-11-02	2007-11-05	2.00
集成测试报告	已计划	中心PMO	中心PMO	2007-11-20	2007-11-21	2.00
测试报告定稿	已计划	中心PMO	中心PMO	2008-01-10	2008-01-11	2.00
第一阶段件由函	未决	系统管...	系统管...	2007-07-31	2007-08-01	1.00

图 16-33 检查阶段交付物

- 5) 通过使用“项目管理人员视图-工作流状态”布局中的“状态”过滤器，可以过滤掉所有已关闭的里程碑点，剩余的里程碑就是仍未提交阶段交付物的里程碑，再进一步核对基线、计划的任务开始、结束时间，QA 可以做出该项目的阶段里程碑点是否延迟提交的判断并直接通过 RPM 的沟通功能，给相应项目的负责人发送通知，要求对方提交相应文档。

- 6) 在运行查询的过程中，如果与当前用户有关的项目数量过多，QA 还可以结合使用客户端左上角提交的项目过滤器过滤出某个或某几个项目，针对这些项目逐一检查交付物的提交情况，如图 16-34 所示。

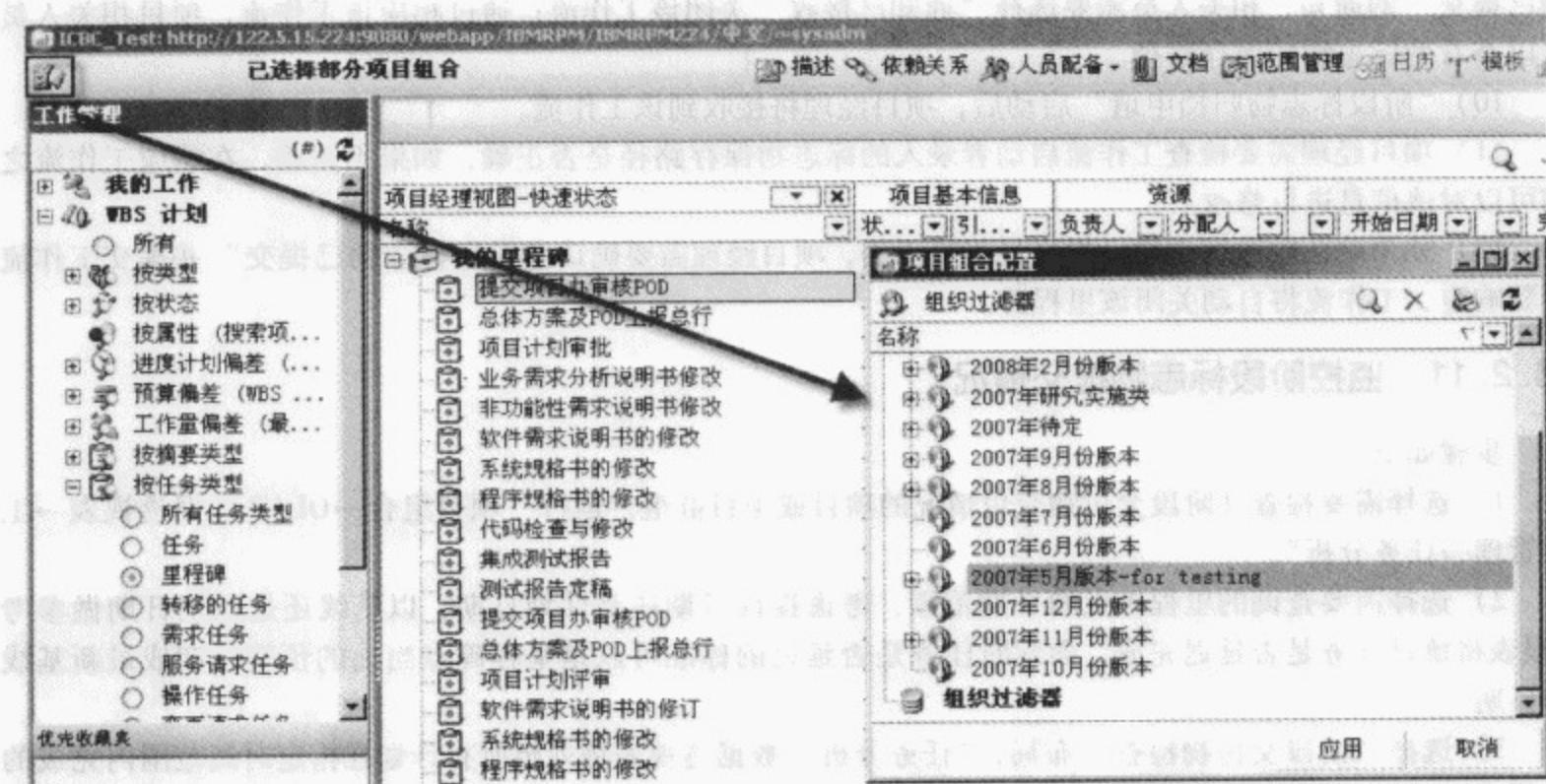


图 16-34 检查提交情况

### 16.2.12 项目评审管理

步骤如下：

- 1) 选择需要提交评审申请的任务，在其右键菜单中选择“沟通→评审申请”。
- 2) 评审申请流程启动后，项目经理在“沟通”视图或“工作管理”视图的底部可以看到相关的工作流步骤，响应该步骤，选择评审类型。
- 3) 根据要启动的评审类型，项目经理可以选择“同行评审”或“正式评审”。如果误启动了流程，可以使用“取消操作”选项直接关闭工作流，如图 16-35 所示。

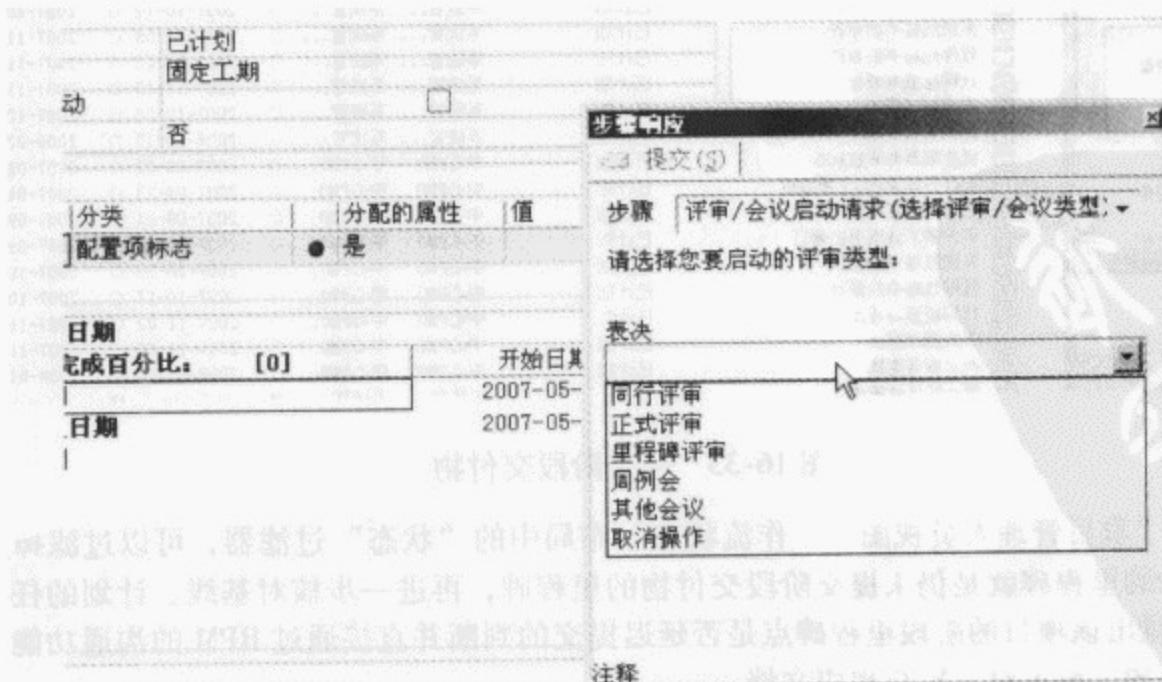


图 16-35 启动的评审类型

- 4) 如果选择了“同行评审”或“正式评审”，项目经理将收到另一个新的工作流步骤响应请求。项目经理必须检出该评审需求后选择本次评审的对象，在“资源”信息端口中录入所有需要参与评审的人，使他们成为“需求参与者”，输入关于评审对象的说明，以便评审参与者接收到评审通知后可以根据该信息找到需要评审的对象。

- 5) 确认所需信息都已录入后即可提交步骤响应。
- 6) 项目经理在响应评审申请时录入的所有“需求参与者”，在评审信息录入完毕后都将接收到评审启动的通知。
- 7) 项目经理和项目组成员都可以在与该评审有关的任务的“范围管理”视图中查看该评审需求的情况，如图 16-36 所示。

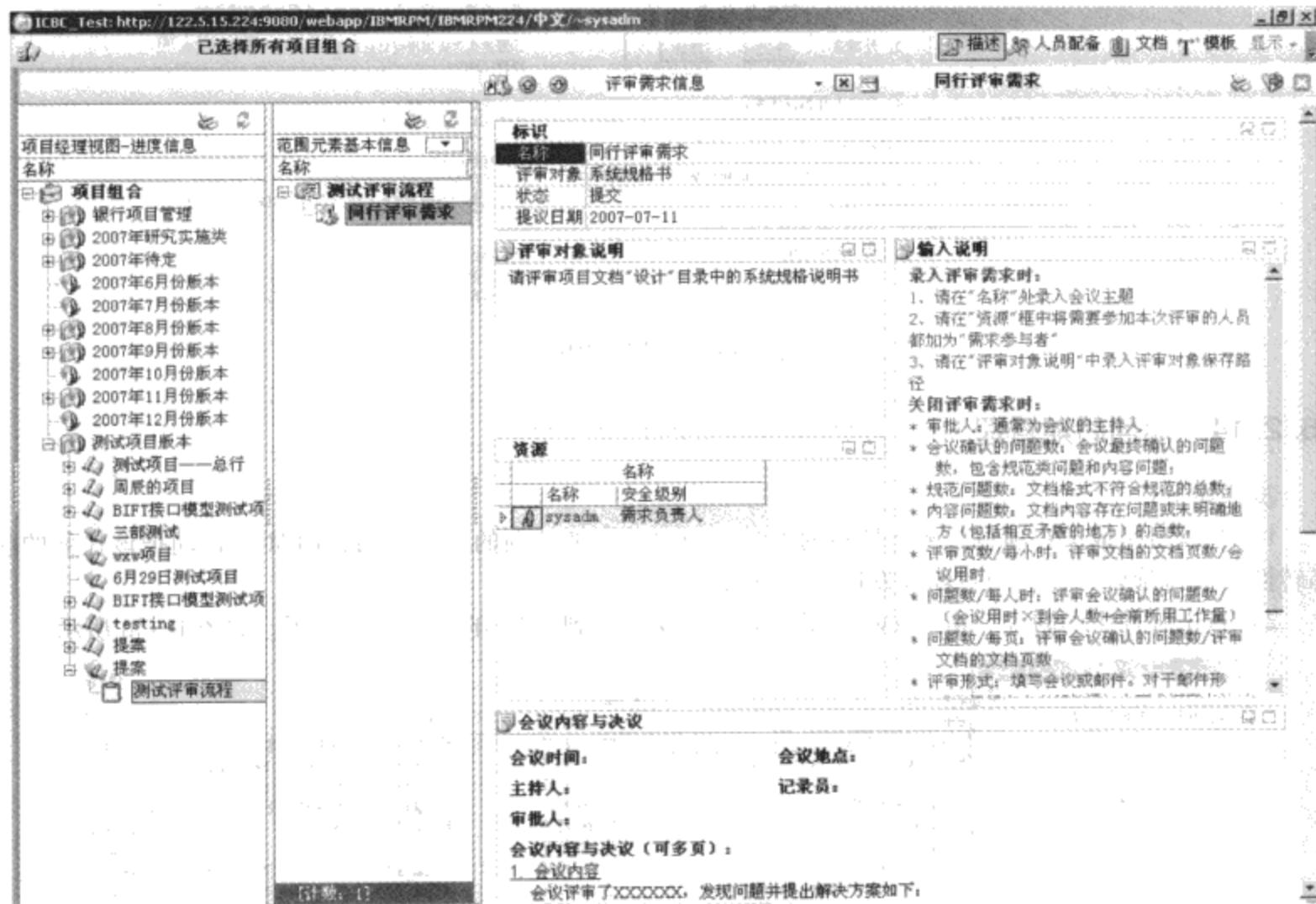


图 16-36 查看评审需求

- 8) 评审结束后或在评审会议期间，项目经理或配置管理员可以针对上述需求启动“关闭评审请求”流程。
- 9) 如果项目经理或配置管理员误对已关闭的评审需求记录启动了“关闭评审请求”，系统将给需求提交者发送“评审记录已关闭”的错误提示，请响应该提示，选择“通知已接受”关闭工作流。
- 10) 对于正确启动了关闭请求的记录，需求的提交者将接收到“关闭评审请求（录入评审结果）”工作流步骤响应请求，提示用户录入评审详情；请检出需求记录，录入评审情况并提交响应结果。
- 11) 所有需求参与者都将接收到评审关闭通知，请响应该通知，选择“通知已接收”。
- 12) 该评审需求的状态将被自动更改为“完成”。

### 16.2.13 项目周报管理

步骤如下：

- 1) 进入待报告项目的“范围元素”视图，选择“模板”，从“项目周报模板”目录下选择周报模板，将它拖拽到项目的范围元素目录，见图 16-37。
- 2) 关闭模板视图，打开新创建的操作的描述视图。缺省情况下，该元素的“提议日期”为创建该记录的日期，如果需要为上周或下周提交项目情况报告，请将该日期修改为上周或下周的时间，按照模板的提示，录入其他项目信息后检入该记录。

范围元素基本信息  
名称: 网上银行个人贷款项目贷款  
描述人名: sysadmin  
提交日期: 2007-06-18

**标识**  
名称: 项目情况报告  
提建议人: sysadmin  
提交日期: 2007-06-18

**本周工作**  
1. 本周计划  
2. 本周需求  
3. 外部请求  
今天 [清除]

**填写说明**

- 功能设计阶段
  - ①描述功能设计的进度情况。
  - ②介绍功能评审情况，包括：评审是否按计划进行，相关人是否全部出席，未确定的需求问题占比是否较多，需确认问题是是否已经全部落实解决的计划和责任人等。
  - ③阐述是否需要根据功能设计(功能点预算)情况调整项目计划？
- 系统设计阶段
  - ①简述系统设计的进度及评审情况。
- 程序设计阶段
  - ①简述程序设计的进度及评审情况。
  - ②介绍集成测试方案和系统测试方案的编写及评审情况。
- 编码阶段
  - ①简述编码的进度情况。
  - ②介绍集成测试环境的搭建和确认情况。

**存在问题**

- SPI<0.7的原因分析和措施：（CMMI评估项目专用）
- 接口问题说明：

图 16-37 项目周报

### 16.2.14 项目后期管理

步骤如下：

- 当项目组成员在项目任务上填写的工时达到项目经理计划的工时以后，RPM 将自动把该任务的状态转换为“待确认”并给项目经理发送邮件通知。
- 点击邮件中的附件“workflow.pmo”，可以进入到 RPM 的“沟通”视图，如图 16-38 所示。

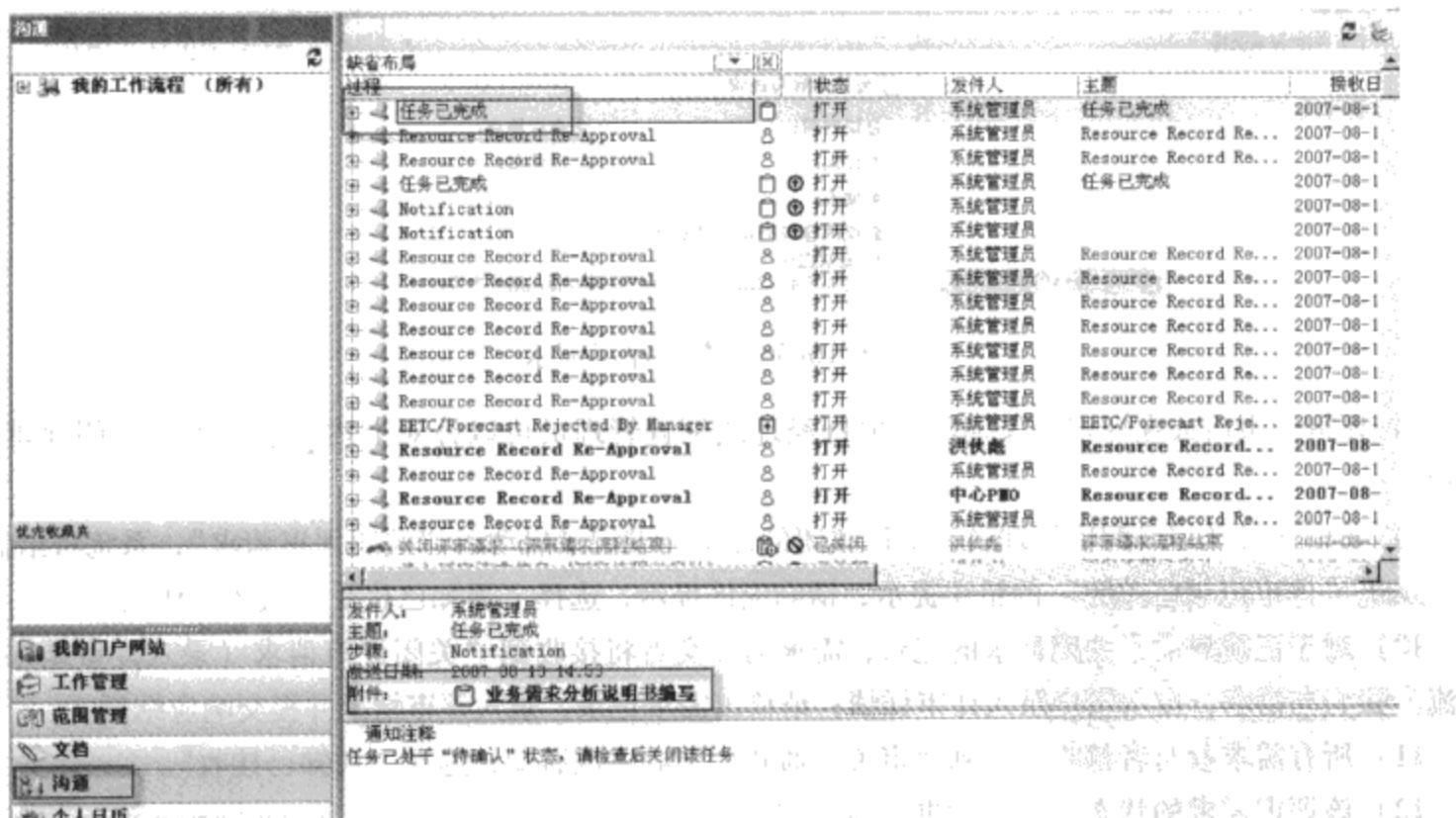


图 16-38 沟通视图看待确认任务

- 选择上述沟通记录，选择“树”，系统将打开任务所处的 WBS，如图 16-39 所示。

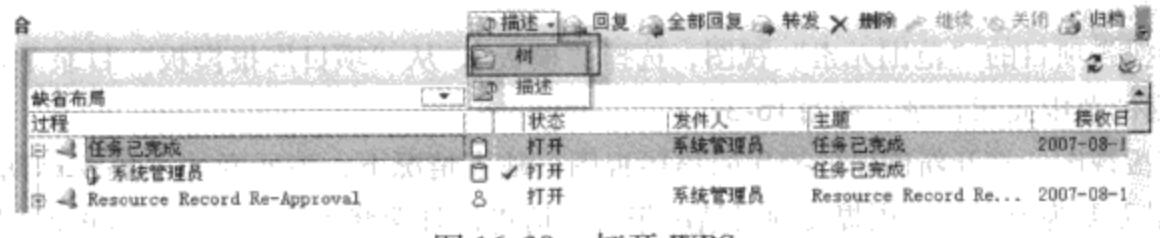


图 16-39 打开 WBS

- 4) 在任务的右键菜单中选择“关闭任务”，关闭该任务，此后项目组成员将无法再在该任务上填报工时。
- 5) 当某个工作包内的所有任务都完成以后，系统将给项目经理发送“工作包已完成”通知，点击邮件中的附件“workflow.pmo”，可以直接进入到RPM的“沟通”视图，选择上述沟通记录，选择“树”，系统将打开工作包所在的WBS。
- 6) 在工作包的右键菜单中选择“更改状态→关闭”，关闭该工作包。
- 7) 如果项目经理关闭的是一个“可交付工件”，则表示该可交付工件所代表的某个项目阶段已完成，此时项目经理还需要手工更改项目状态。
- 8) 在项目的右键菜单中选择“更改状态”，将项目状态设置为新阶段所处的状态。

## 16.2.15 项目评估

步骤如下：

- 1) 进入“工作管理”视图，选择“我的团队工作→所有工作包”，找到需要评估的项目组成员，选择“描述”，检出该资源记录，最大化“计分卡”信息端口，将“项目组成员考核表”添加到该资源的记录中，如图16-40所示。

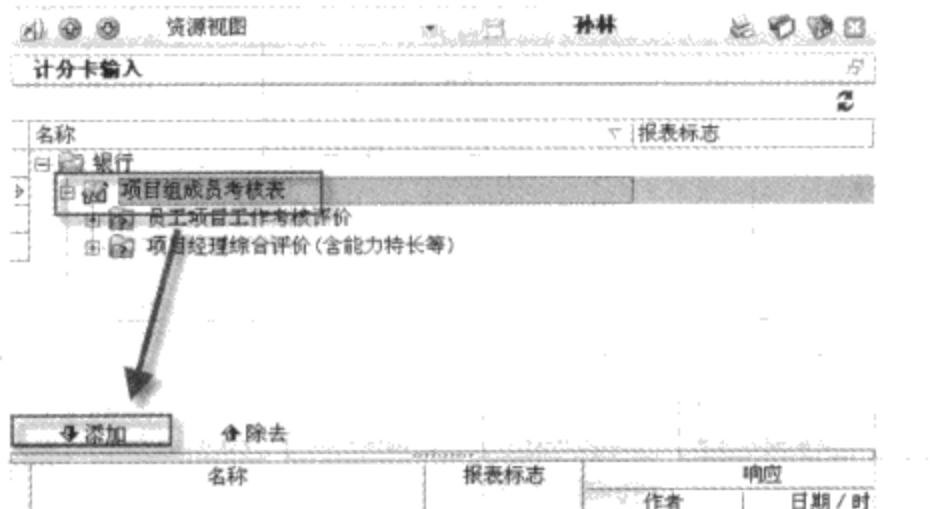


图16-40 添加项目组成员考核表

- 2) 根据该资源在项目中的表现情况选择各个问题的响应，如图16-41所示。

名称	报表标志	响应						
项目组成员考核表								
发布								
项目工作考核评价								
重命名								
项目完成进度	<input checked="" type="radio"/>	<table border="1"> <tr><td>所有任务按时完成 (15-21)</td></tr> <tr><td>系统管理员 2007-08-</td></tr> <tr><td>良 (40-50)</td></tr> <tr><td>系统管理员 2007-08-</td></tr> <tr><td>良 (10-15)</td></tr> <tr><td>无</td></tr> </table>	所有任务按时完成 (15-21)	系统管理员 2007-08-	良 (40-50)	系统管理员 2007-08-	良 (10-15)	无
所有任务按时完成 (15-21)								
系统管理员 2007-08-								
良 (40-50)								
系统管理员 2007-08-								
良 (10-15)								
无								
项目完成质量	<input checked="" type="radio"/>							
项目文档的完善和规范程度	<input checked="" type="radio"/>							
其他加/减分项	<input checked="" type="radio"/>	<table border="1"> <tr><td>所有任务按时完成 (15-21)</td></tr> <tr><td>系统管理员 2007-08-</td></tr> <tr><td>良 (40-50)</td></tr> <tr><td>系统管理员 2007-08-</td></tr> <tr><td>良 (10-15)</td></tr> <tr><td>无</td></tr> </table>	所有任务按时完成 (15-21)	系统管理员 2007-08-	良 (40-50)	系统管理员 2007-08-	良 (10-15)	无
所有任务按时完成 (15-21)								
系统管理员 2007-08-								
良 (40-50)								
系统管理员 2007-08-								
良 (10-15)								
无								
项目经理综合评价 (含能力特长等)								
对该员工的能力特长的综合评价	<input checked="" type="radio"/>	<table border="1"> <tr><td>所有任务按时完成 (15-21)</td></tr> <tr><td>系统管理员 2007-08-</td></tr> <tr><td>良 (40-50)</td></tr> <tr><td>系统管理员 2007-08-</td></tr> <tr><td>良 (10-15)</td></tr> <tr><td>无</td></tr> </table>	所有任务按时完成 (15-21)	系统管理员 2007-08-	良 (40-50)	系统管理员 2007-08-	良 (10-15)	无
所有任务按时完成 (15-21)								
系统管理员 2007-08-								
良 (40-50)								
系统管理员 2007-08-								
良 (10-15)								
无								

图16-41 员工考核数据

- 在“响应”列选择对该员工的评价。
  - 在相应注释/修正操作处填上对该员工的评分结果（具体分数）。
  - 在“项目经理综合评价”中填写对该员工的评价信息（文本）。
- 3) 确认录入数据无误后，选中“项目组成员考核表”列，在右键菜单中选择“发布”，发布该计分卡。
  - 4) 项目考核得分由项目组的审计QA维护，选择要录入项目考核得分的项目，打开其“描述”，检出项目描述后选择“质控办QA视图-项目度量”。

- 5) 在“项目考核得分情况”信息端口中录入项目考核得分。
- 6) 选择要录入项目考核得分的项目，打开其“描述”，检出项目描述后选择“质控办 QA 视图 - 项目度量”。
- 7) 最大化“计分卡”信息端口。
- 8) 根据项目所处的阶段和类型，选择需要为项目添加的计分卡。
- 9) 根据该项目的表现情况选择各个问题的响应。
  - 在“响应”列选择对该项目工作的评价。
  - 在相应注释/修正操作处填写评价的理由。
  - 如果某个选项对当前项目不适用，请选择“裁减”或“计划裁减”。
- 10) 确认录入数据无误后，选中计分卡名称列，在右键菜单中选择“发布”，发布该计分卡。
- 11) 选择要查看其项目评分情况的项目或项目群，运行“项目组合→计分卡图”，双击计分卡可以看到评分明细，如图 16-42、图 16-43 所示。

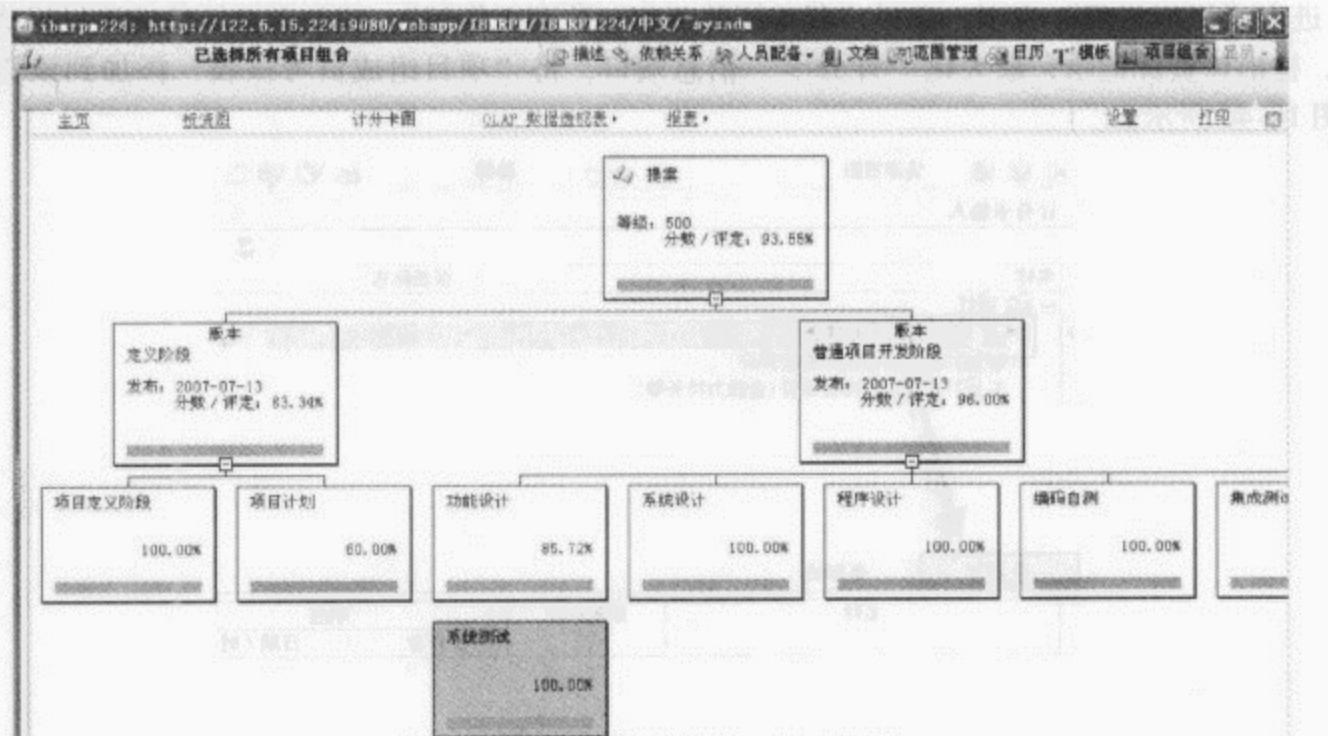
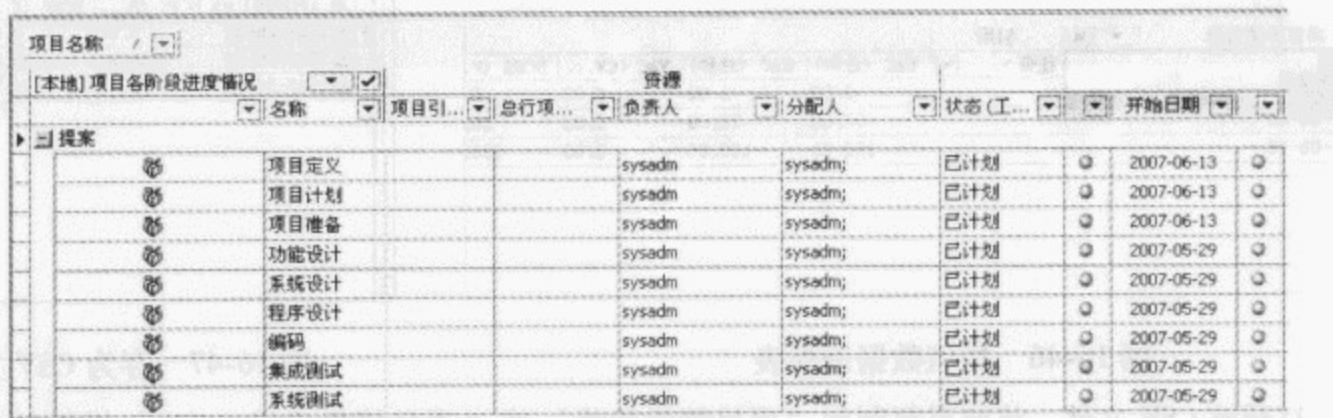


图 16-42 查看评分明细

名称	响应	报表标志	费率	描述	注释 / 修正操作	作者
- yid 普通项目开发阶段			96.00 %			sysadm
- 功能设计			85.72 %			sysadm
① 《业务需求分析说明书》... 不符合	✓		0.01 %			sysadm
② 《业务需求分析说明书》... 未见不符合	✓		100.00 %			sysadm
③ 《业务需求分析说明书》... 不符合	✓		0.01 %			sysadm
④ 《业务需求分析说明书》... 未见不符合	✓		100.00 %			sysadm
⑤ 《非功能性需求说明书》... 未见不符合	✓		100.00 %			sysadm
⑥ 《非功能性需求说明书》... 未见不符合	✓		100.00 %			sysadm
⑦ 《非功能性需求说明书》... 未见不符合	✓		100.00 %			sysadm
⑧ 《非功能性需求说明书》... 未见不符合	✓		100.00 %			sysadm
⑨ 《软件需求说明书》同行... 未见不符合	✓		100.00 %			sysadm
⑩ 《软件需求说明书》是否... 未见不符合	✓		100.00 %			sysadm
⑪ 里程碑是否符合规范... 未见不符合	✓		100.00 %			sysadm
⑫ 配置管理是否符合规范 (... 未见不符合	✓		100.00 %			sysadm
⑬ 变更管理是否符合规范 (... 未见不符合	✓		100.00 %			sysadm
⑭ 周例会、周报是否符合规范	未见不符合	✓	100.00 %			sysadm
- 系统设计			100.00 %			sysadm
- 程序设计			100.00 %			sysadm
- 编码自测			100.00 %			sysadm
- 集成测试			100.00 %			sysadm
- 系统测试			100.00 %			sysadm

图 16-43 看到评分明细

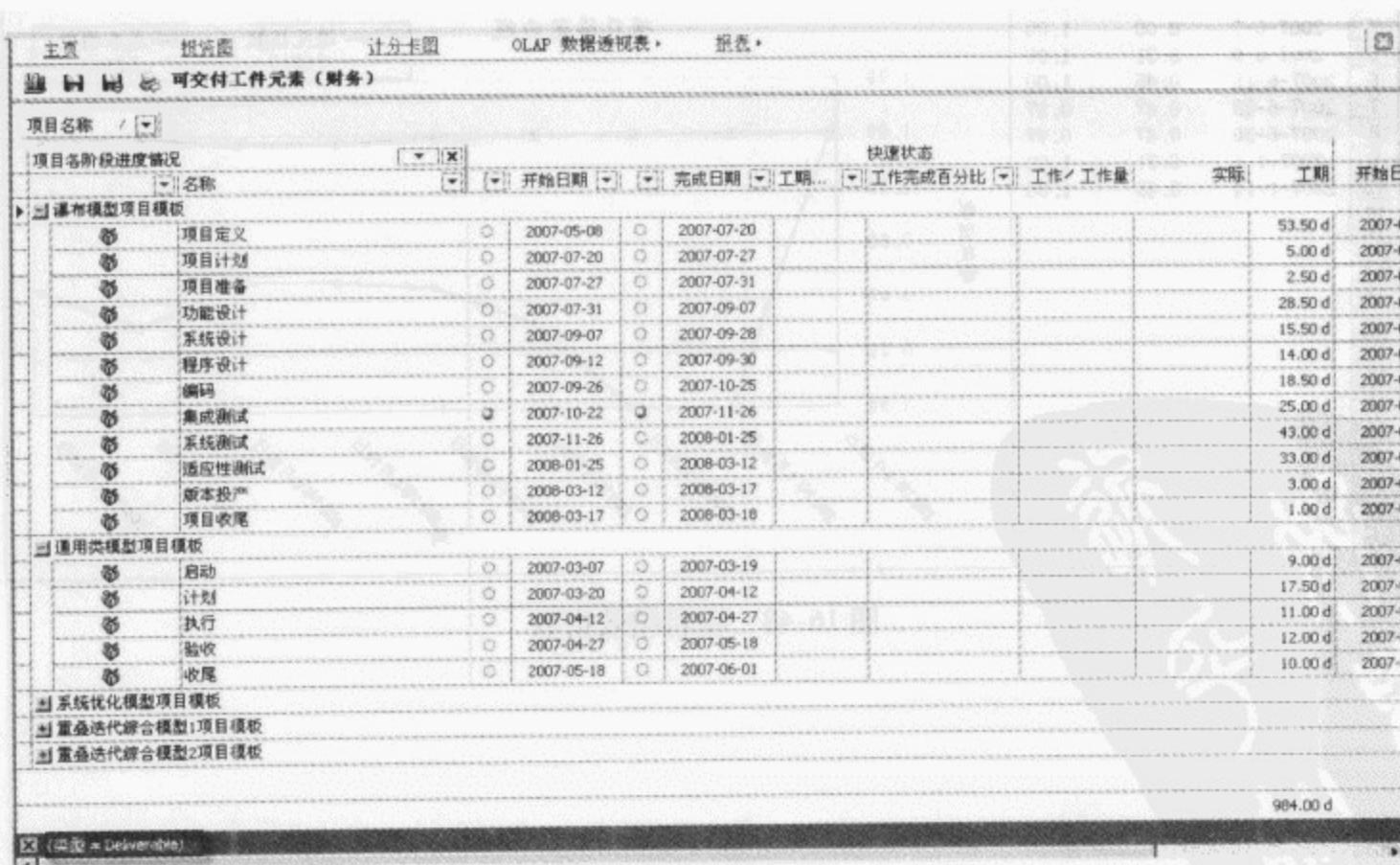
- 12) 选择要统计的数据对象，然后选择“项目组合→OLAP 数据透视表”或“数据透视表/报表”。  
 13) 如要获取项目各阶段完成情况。选择“OLAP 数据透视表→定制→项目周报→阶段完成情况”，将列出选定项目中可交付件的完成情况，如图 16-44 所示。



项目名称 /		资源						
[本地] 项目各阶段进度情况		名称	项目引...	总行项...	负责人	分配人	状态 (工...	开始日期
<b>▶ 提交</b>								
①	项目定义		sysadm	sysadm;	已计划	●	2007-06-13	●
②	项目计划		sysadm	sysadm;	已计划	●	2007-06-13	●
③	项目准备		sysadm	sysadm;	已计划	●	2007-06-13	●
④	功能设计		sysadm	sysadm;	已计划	●	2007-05-29	●
⑤	系统设计		sysadm	sysadm;	已计划	●	2007-05-29	●
⑥	程序设计		sysadm	sysadm;	已计划	●	2007-05-29	●
⑦	编码		sysadm	sysadm;	已计划	●	2007-05-29	●
⑧	集成测试		sysadm	sysadm;	已计划	●	2007-05-29	●
⑨	系统测试		sysadm	sysadm;	已计划	●	2007-05-29	●

图 16-44 阶段完成情况

- 14) 对于项目数据，还需要找到要运行的数据透视表的类别，在“布局”下拉列表中可以看到当前类别的数据透视表中所有的布局，选择您要运行的数据透视表即可。  
 15) 与项目信息有关的数据透视表都包含在“OLAP 数据透视表→定制→项目信息统计”目录中，分别展示。  
 16) 可以通过“定制→项目信息统计→项目基本信息”和“定制→项目信息统计→项目进度情况”来定制数据表信息。  
 17) 本项目的一部分数据透视表如图 16-45、图 16-46、图 16-47、图 16-48 所示。



项目名称 /		快速状态								
项目各阶段进度情况		名称	开始日期	完成日期	工期	工作完成百分比	工作／工作量	实际	工期	开始日
<b>▶ 演示模型项目模板</b>										
①	项目定义	○	2007-05-08	○	2007-07-20				53.50 d	2007-0
②	项目计划	○	2007-07-20	○	2007-07-27				5.00 d	2007-0
③	项目准备	○	2007-07-27	○	2007-07-31				2.50 d	2007-0
④	功能设计	○	2007-07-31	○	2007-09-07				28.50 d	2007-0
⑤	系统设计	○	2007-09-07	○	2007-09-28				15.50 d	2007-0
⑥	程序设计	○	2007-09-12	○	2007-09-30				14.00 d	2007-0
⑦	编码	○	2007-09-26	○	2007-10-25				18.50 d	2007-0
⑧	集成测试	○	2007-10-22	○	2007-11-26				25.00 d	2007-0
⑨	系统测试	○	2007-11-26	○	2008-01-25				43.00 d	2007-0
⑩	适应性测试	○	2008-01-25	○	2008-03-12				33.00 d	2007-0
⑪	版本投产	○	2008-03-12	○	2008-03-17				3.00 d	2007-0
⑫	项目收尾	○	2008-03-17	○	2008-03-18				1.00 d	2007-0
<b>▶ 通用类模型项目模板</b>										
⑬	启动	○	2007-03-07	○	2007-03-19				9.00 d	2007-0
⑭	计划	○	2007-03-20	○	2007-04-12				17.50 d	2007-0
⑮	执行	○	2007-04-12	○	2007-04-27				11.00 d	2007-0
⑯	验收	○	2007-04-27	○	2007-05-18				12.00 d	2007-0
⑰	收尾	○	2007-05-18	○	2007-06-01				10.00 d	2007-0
<b>▶ 系统优化模型项目模板</b>										
<b>▶ 重叠迭代综合模型1项目模板</b>										
<b>▶ 重叠迭代综合模型2项目模板</b>										
984.00 d										
(类型 = Deliverable)										

图 16-45 各阶段进度情况

从项目“管理文档”目录下取“项目绩效曲线”或“项目绩效曲线（宏）”模板，点击“项目组合→OLAP 数据透视表→定制→项目周报→任务度量”。

- 选择“项目 CPL/SPI 计算”数据透视表，如图 16-46 所示。
- 选择“另存为可见”，将结果另存为任意 CSV 文件，如图 16-47 所示。

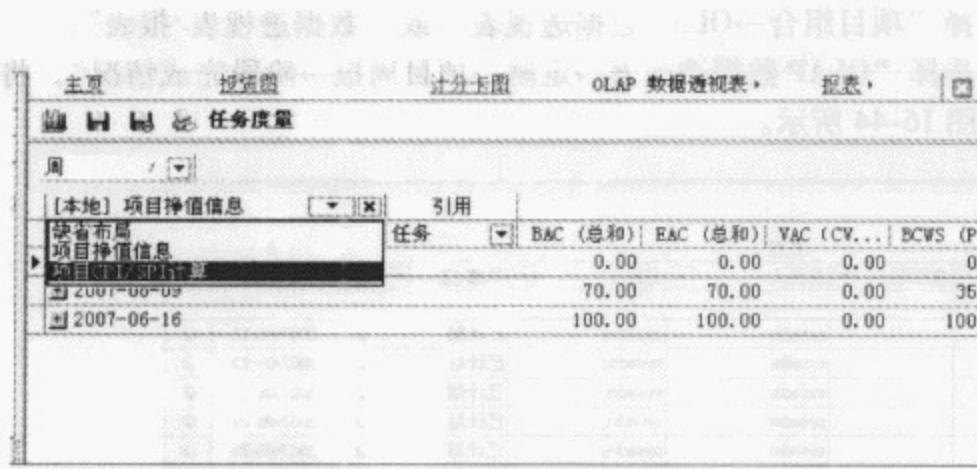


图 16-46 挣值数据透视表

主页		进度图		计分卡图	
周	任务度量	周	任务度量	周	任务度量
[本...]	挣值	周 (时间段)	BCWS (P...) BCWP (E...) ACWP (A...)		
2007-06-02	0.00	2007-06-09	35.00	35.00	35.00
2007-06-16	100.00	2007-06-23	79.00	79.00	79.00

图 16-47 存为 CSV 文件

- 打开该 CSV 文件，将结果复制到“项目绩效曲线”或“项目绩效曲线（宏）”模板的“EV 原始数据”页，如图 16-48 所示。
- 使用“项目绩效曲线（宏）”，用户只需要按下“计算绩效指数”按钮，就可以获得最新曲线。

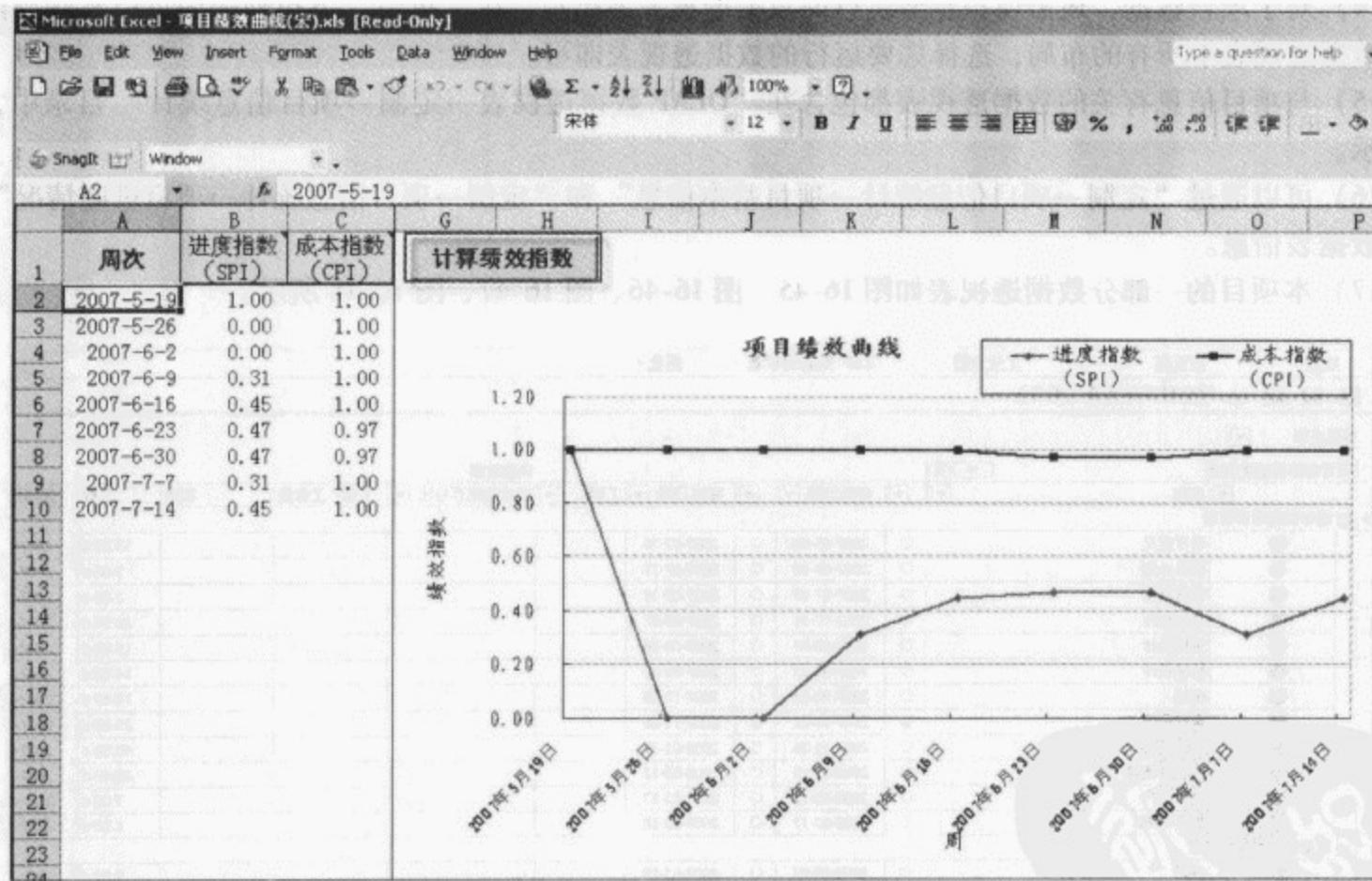


图 16-48 项目绩效指数

# 结 束 语

本书路线图就像一个导游指南，指导你如何进行软件项目管理，告诉你可能遇到的困难和问题等。同时也强调了要积极预防，避免风险和意外的发生，以及出现问题如何处理的方法等。项目管理过程，是一个复杂的大系统。从系统工程的观点看，项目管理普遍涉及到多个目标的要求，由于资源约束性、多重目标性，一个优秀的项目管理人员，应该努力学习项目管理相关知识、技能，注重实践中的分析问题能力，结合理论知识，在实践中锻炼提高，积极解决各种各样的问题，形成一套行之有效的管理方法，并再通过实践来检验它的科学性，使项目管理工作越做越好。项目管理是一门艺术，而艺术是无止境的。所以，要多多的掌握项目管理的理论，培养管理技能，这样就有利于今后的发展。最后希望大家都成为一个优秀的项目经理。

# 参 考 文 献

- [1] J D 弗雷姆. 组织机构中的项目管理 [M]. 北京: 世界图书出版公司, 2000.
- [2] Rajeev T Shandilya. 软件项目管理 [M]. 北京: 科学技术出版社, 2002.
- [3] J D 弗雷姆. 新项目管理 [M]. 北京: 世界图书出版公司, 2001.
- [4] 史蒂夫·麦克康奈尔. 微软项目求生法则 [M]. 北京: 机械工业出版社, 2000.
- [5] 史蒂夫·马魁尔. 微软研发致胜策略 [M]. 北京: 机械工业出版社, 2000.
- [6] 邱莞华, 等. 现代项目管理导论 [M]. 北京: 机械工业出版社, 2002.
- [7] 纪燕萍. 21世纪项目管理教程 [M]. 北京: 人民邮电出版社, 2002.
- [8] 孙涌, 等. 现代软件工程 [M]. 北京: 北京希望电子出版社, 2002.
- [9] Internal Function Point Users Group. IT Measurement [M]. 北京: 清华大学出版社, 2003.
- [10] 纪康宝. 软件开发项目可行性研究与经济评价手册 [M]. 长春: 吉林科学技术出版社, 2002.
- [11] Mark J Christensen. The Project Manager's Guide to Software Engineering's Best Practices [M]. 北京: 电子工业出版社, 2004.
- [12] John McGarry, 等. Practical Software Measurement [M]. 北京: 机械工业出版社, 2003.
- [13] 韩万江, 等. 软件开发项目管理 [M]. 北京: 机械工业出版社, 2003.
- [14] PMI. A Guide to the Project Management Body of Knowledge [S]. 2000.
- [15] 6σ工作室. PMP 成功之路 [M]. 北京: 机械工业出版社, 2003.
- [16] Garmus D, David H. The Software Measuring Process: A Practical Guide to Functional Measurements [M]. New Jersey: Yourdon Press, 1996.
- [17] Humphrey W. A Discipline for Software Engineering, SEI Series in Software Engineering [M]. Boston: Addison-Wesley, 1995.
- [18] John H Baumert. Software Measures and the Capability Maturity Model. CMU/SEI-92-TR-25 [EB]. 1992.
- [19] PMI. A Guide to the Project Management Body of Knowledge [EB]. 2000.
- [20] Ronald P Higuera, Yacov Y Haimes. Software Risk Management [EB]. CMU/SEI-96-TR-012. 1996.
- [21] Anne Mette, 等. Configuration Management Principles and Practice [M]. 北京: 清华大学出版社, 2003.
- [22] Newbold R C. Project Management in the Fast Lane—Applying the Theory of Constraints [M]. Boca Raton: St. Lucie Press, 1998.
- [23] Leach L P. Critical Chain Project Management Improves Project Performance [J]. Project Management Journal, 1999, 30 (2): 39–51.
- [24] Mabin V J, Balderstone S J. The World of the Theory of Constraints—A Review of the International Literature [M]. Boca Raton: St. Lucie Press, 1999.
- [25] Goldratt E M. Critical Chain [M]. Great Barrington: The North River Press, 1997.
- [26] Gutierrez G J, Kouvelis P. Parkinson's Law and Its Implication for Project management [J]. Management Science, 1991, 37 (8): 990–1001.
- [27] Cook S C. Apply Critical Chain to 31 Improve the Management of Uncertainty in Projects [D]. Boston: Massachusetts Institute of Technology, 1998.
- [28] 万伟, 蔡晨, 王长峰. 在单资源约束项目中的关键链管理 [J]. 中国管理科学, 2003, 11 (2): 70–75.
- [29] Herroelen W, Leus R, Demeulemeester E. Critical Chain Project Scheduling: Do Not Oversimplify [J]. Project Management Journal, 2002, 33: 48–60.
- [30] 杨雪松、胡旻. 基于关键链方法的多项目管理 [J]. 工业工程与管理, 2005, 2: 48–52.
- [31] 寿涌毅. 关键链项目管理方法综述 [J], 项目管理技术, 2006(9).

[ General Information ]

书名 = 软件项目管理案例教程

作者 = 韩万江 , 姜立新编著

页数 = 320

出版社 = 北京市 : 机械工业出版社

出版日期 = 2009 . 06

S S 号 = 12400514

D X 号 = 000006734706

U R L = h t t p : / / b o o k . s z d n e t . o r g . c n / b o o k D e t a i l . j s

p ? d x N u m b e r = 000006734706 & d = A 5 2 3 8 8 5 D 4 0 E 4 C 7 6 8 B

B 1 9 E 3 5 1 8 2 F E 5 F 5 A