

SOFTWARE PROJECT MANAGEMENT

软件项目管理

覃征 杨利英 高勇民 贺升平 韩毅 编著

Qin Zheng Yang Liying Gao Yongmin He Shengpin Han Yi

清華園



清华大学出版社

软件项目管理

覃征 杨利英 高勇民 贺升平 韩毅 编著

清华大学出版社
北京

内 容 简 介

本书系统地讲述了软件项目管理的基本概念、基本原理及基本方法。全书共分9章,围绕软件项目的进行过程对其中的管理内容展开论述。第1章是导论,介绍了软件项目管理的背景学科和相关概念;第2章从需求工程的角度阐述了软件项目中的需求管理;第3章在详细分析几个成本模型的基础上对软件项目估算进行了描述,并介绍了进度安排的相关内容;第4章是对软件项目中配置管理的论述;第5章讨论如何对软件项目中出现的各种风险进行管理;第6章讨论如何保证得到高质量的软件产品;第7章介绍了软件项目中的资源管理,包括人力资源管理和软、硬件资源管理;第8章讲述了国内外软件产业和软件市场的现状及策略;第9章是案例分析,将本书介绍的基本原理用于一个实际的软件项目。

本书可以作为高等院校软件项目管理课程的教科书,也可以作为从事软件项目管理、软件工程、软件系统分析与设计、软件开发及应用等工作的研究人员和其他相关人员的参考书。

版权所有,翻印必究。举报电话:010-62782989 13901104297 13801310933

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

软件项目管理/覃征等编著. —北京:清华大学出版社,2004.7

ISBN 7-302-08284-7

I. 软… II. 杨… III. 软件开发—项目管理—高等学校—教材 IV. TP311.52

中国版本图书馆CIP数据核字(2003)第019138号

出 版 者:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

组稿编辑:丁 岭

文稿编辑:许振伍

封面设计:王 永

印 刷 者:清华大学印刷厂

装 订 者:三河市金元装订厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:17.75 字数:440千字

版 次:2004年7月第1版 2004年7月第1次印刷

书 号:ISBN 7-302-08284-7/TP·5973

印 数:1~5000

定 价:29.00元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103或(010)62795704

前 言

软件及信息产业的发展将带动传统产业的发展与改造，促进国民经济持续、健康的发展，增强国际竞争力，因而软件产业快速、良性的发展具有重要意义。我国政府明确提出：力争在 2010 年使我国软件产业的研究开发和生产能力达到或接近国际先进水平。

20 世纪 60 年代，软件和硬件之间的不平衡发展导致的软件危机是软件产业面临的第一个难题。直到现在，类似的问题仍然存在。美国政府统计署的数据显示：全球最大的软件消费商——美国军方，每年要花费数十亿美元购买软件，而在其所购买的软件中，可直接使用的只占 2%，另外 3% 需要做一些修改，其余 95% 都成了垃圾。在技术相对成熟的现在，管理在软件产业中的地位变得越来越重要，也是导致许多软件失败的主要因素。针对当年的软件危机，学者们提出了用工业界工程化的思想，即软件工程去应对。同样，针对软件产业中的管理问题，可以采用工程中项目的思想，由此促使了软件项目管理学科的出现。

软件项目管理是软件工程和项目管理的交叉学科，是项目管理的原理和方法在软件工程领域的应用。与一般的工程项目相比，软件项目有其特殊性，主要体现在软件产品的抽象性上，因此软件项目管理的难度要比一般的工程项目管理的难度大。

本书系统地讲述了软件项目管理的基本概念、基本原理及基本方法，围绕软件项目的开发过程，从软件项目需求管理、软件项目估算与进度管理、软件项目配置管理、软件项目风险管理、软件项目质量管理、软件项目资源管理六个方面对软件项目中的管理问题进行了探讨，旨在为相关人员提供一些基础参考，促进我国软件产业的快速发展。

在本书的编写过程中，得到了清华大学信息学院、软件学院、计算机科学与技术系、西安交通大学计算机科学与技术系、电子商务研究所、计算机软件研究所许多教师的支持和指教，同时也得到了清华大学出版社的大力支持，我们在此表示衷心的感谢。本书的编写参考了大量的文献和网站，我们对所有这些文献著作者和网站版权所有表示真诚的谢意。

本书由覃征教授确定研究内容和整体结构，第 1 章、第 2 章、第 3 章、第 4 章由杨利英编写，第 5 章、第 6 章由高勇民编写，第 7 章、第 8 章由贺升平编写，第 9 章由韩毅编写。

由于水平有限，加之软件产业的发展非常迅速，书中难免有疏漏和不妥之处，敬请读者批评指正。

编 者

目 录

第 1 章 导论	1
1.1 软件工程.....	1
1.1.1 软件工程定义.....	1
1.1.2 软件工程框架.....	1
1.1.3 软件工程模型.....	5
1.2 项目管理.....	7
1.2.1 项目与项目管理.....	7
1.2.2 项目管理知识体系.....	11
1.2.3 项目管理学科的发展.....	13
1.3 软件项目管理.....	14
1.3.1 软件项目产品的特点.....	14
1.3.2 软件项目失控的原因.....	15
1.3.3 软件项目管理的内容.....	19
1.4 小结.....	20
第 2 章 软件项目需求管理	21
2.1 软件需求.....	21
2.1.1 软件需求概念.....	21
2.1.2 软件需求类别.....	23
2.1.3 软件需求文档.....	26
2.1.4 软件需求度量.....	29
2.2 需求工程.....	31
2.2.1 产生与发展.....	31
2.2.2 研究内容.....	32
2.3 需求管理.....	34
2.3.1 需求管理的必要性.....	34
2.3.2 目标和原则.....	35
2.3.3 需求管理活动.....	37
2.3.4 需求变更管理.....	37
2.3.5 需求文档版本.....	40
2.3.6 需求状态.....	41
2.3.7 需求跟踪.....	42
2.4 需求管理质量保证.....	45
2.4.1 需求验证.....	45

2.4.2 需求评审	47
2.5 小结	48
第 3 章 软件项目估算与进度管理	49
3.1 软件项目估算	49
3.1.1 估算的意义	49
3.1.2 估算的时机	50
3.2 软件规模	51
3.2.1 工作分解结构	51
3.2.2 代码行	52
3.2.3 功能点	52
3.2.4 PERT 规模估计	54
3.2.5 软件生产率	55
3.3 软件项目成本估算	56
3.3.1 成本估算方法	56
3.3.2 成本估算模型	60
3.3.3 COCOMO 模型	62
3.3.4 Putnam 模型	72
3.3.5 成本估算步骤	74
3.3.6 成本模型的评价	77
3.4 软件项目进度管理	78
3.4.1 制定项目计划	78
3.4.2 分阶段交付	80
3.4.3 进度安排	81
3.4.4 已获值分析	86
3.5 小结	90
第 4 章 软件项目配置管理	91
4.1 配置管理概念	91
4.1.1 基本概念	91
4.1.2 软件配置管理	94
4.2 配置管理组织和职责	95
4.2.1 CMM 二级体系	95
4.2.2 SCM 的职责	97
4.2.3 SCM 文件体系与过程活动	98
4.3 配置管理功能	99
4.3.1 配置标识	99
4.3.2 配置控制	101
4.3.3 配置状态报告	104

4.3.4	配置审核	106
4.4	配置管理计划	109
4.4.1	计划 CM	109
4.4.2	CM 计划的大纲	109
4.5	基于构件的配置管理	111
4.5.1	软件复用	111
4.5.2	软件构件技术	113
4.5.3	基于构件的版本管理	115
4.5.4	基于构件的配置管理	117
4.6	小结	119
第 5 章	软件项目风险管理	120
5.1	概述	120
5.1.1	风险	120
5.1.2	风险管理	120
5.1.3	软件风险	121
5.1.4	软件风险管理	122
5.1.5	风险管理同项目的关系	123
5.1.6	风险管理的意义	124
5.1.7	软件风险管理的主要内容	124
5.2	风险管理计划	125
5.2.1	风险管理计划的内容	125
5.2.2	确定风险管理目标	126
5.2.3	制订风险管理策略	126
5.2.4	定义风险管理过程	128
5.2.5	定义风险管理验证	128
5.2.6	建立风险管理机制	128
5.3	风险识别	130
5.3.1	风险识别活动的依据	130
5.3.2	风险识别活动的成果	130
5.3.3	风险识别过程	131
5.3.4	风险识别的工具与技术	131
5.4	风险分析	134
5.4.1	风险分析的依据	134
5.4.2	风险分析的成果	135
5.4.3	风险分析过程	135
5.4.4	风险分析技巧与工具	137
5.4.5	软件项目的主要风险	138
5.5	风险计划	140

5.5.1	风险计划的依据.....	140
5.5.2	风险计划的成果.....	140
5.5.3	风险计划过程.....	140
5.6	风险跟踪.....	143
5.6.1	风险跟踪的依据.....	144
5.6.2	风险跟踪的成果.....	144
5.6.3	风险跟踪过程.....	144
5.6.4	风险跟踪技巧.....	146
5.7	风险应对.....	147
5.7.1	风险应对的依据.....	147
5.7.2	风险应对的成果.....	147
5.7.3	风险应对过程.....	148
5.7.4	风险管理回报.....	148
5.8	风险管理验证.....	149
5.8.1	评审风险管理计划.....	149
5.8.2	审计执行者及其典型产物.....	149
5.8.3	生成审计报告.....	150
5.8.4	跟踪行动项目.....	150
5.9	小结.....	150
第 6 章	软件项目质量管理.....	151
6.1	质量管理的概念.....	151
6.1.1	软件质量.....	151
6.1.2	软件产品质量与过程质量.....	152
6.1.3	软件质量保证 (SQA).....	153
6.2	软件评审.....	154
6.2.1	评审的概念.....	154
6.2.2	评审的实施.....	154
6.2.3	取得评审成功的关键.....	155
6.3	软件测试.....	156
6.3.1	软件测试的概念.....	156
6.3.2	软件测试类型.....	157
6.3.3	测试的原则.....	161
6.3.4	测试计划.....	162
6.3.5	测试用例的开发.....	163
6.3.6	测试的执行与报告.....	164
6.4	软件缺陷跟踪.....	165
6.5	软件缺陷预防.....	166
6.5.1	问题的提出.....	166

6.5.2	缺陷预防的原则.....	166
6.5.3	缺陷预防的步骤.....	167
6.6	ISO9000: 2000 质量认证体系.....	169
6.6.1	ISO9000 的概念.....	169
6.6.2	ISO9000 标准的八项质量管理原则.....	170
6.6.3	ISO9000 认证的条件和程序.....	172
6.6.4	ISO9001: 2000 标准的内容.....	173
6.7	软件能力成熟度模型 CMM.....	184
6.7.1	CMM 的提出.....	184
6.7.2	CMM 的演化过程.....	185
6.7.3	CMM 的基本概念.....	186
6.7.4	软件过程的内容和特性.....	187
6.7.5	软件过程成熟度的五个等级.....	187
6.7.6	成熟度等级的特征.....	188
6.7.7	CMM 的内部结构.....	193
6.7.8	软件过程改进.....	197
6.7.9	CMM 与 ISO9001 的比较.....	200
6.7.10	CMM 与 PSP 和 TSP.....	201
6.8	小结.....	202
第 7 章	软件项目资源管理.....	204
7.1	人力资源管理.....	204
7.1.1	研究任务.....	204
7.1.2	人员的组织与分工.....	204
7.1.3	人力计划的基本规则.....	207
7.1.4	人力资源计划的平衡.....	208
7.2	软件资源管理.....	210
7.2.1	软件资源的可复用性.....	210
7.2.2	可复用软件资源的管理.....	213
7.2.3	CASE 工具及其管理.....	215
7.3	硬件资源管理.....	217
7.3.1	硬件设备的经济管理.....	217
7.3.2	硬件设备的技术管理.....	218
7.4	小结.....	220
第 8 章	软件市场与软件产业.....	221
8.1	软件市场.....	221
8.1.1	软件商品.....	221
8.1.2	软件的成本.....	222

8.1.3	软件的定价	222
8.1.4	软件市场的垄断现象.....	222
8.1.5	软件市场的发展现状.....	223
8.2	软件产业	224
8.2.1	软件产业的特点.....	224
8.2.2	软件产业在国民经济中的地位.....	225
8.2.3	国外软件产业发展策略.....	226
8.2.4	对我国软件产业发展策略的认识.....	231
8.3	小结	234
第9章	案例分析	235
9.1	案例项目描述	235
9.1.1	项目名称	235
9.1.2	研究背景	235
9.1.3	项目目标	236
9.2	项目计划总览	236
9.2.1	确定项目范围和目标.....	236
9.2.2	确定项目结构.....	237
9.2.3	分析项目特征.....	239
9.2.4	确定项目产品和活动.....	239
9.2.5	估计每个活动的工作量.....	242
9.2.6	确定活动风险.....	243
9.2.7	分配资源	243
9.2.8	检查、公布、执行计划.....	244
9.2.9	更细层次上的计划.....	244
9.3	项目开发模型	244
9.4	工作量估算	246
9.4.1	估算方法选择.....	246
9.4.2	估算技巧与估算修正.....	246
9.5	资源管理	247
附录	249
附录 A	可行性分析报告	249
附录 B	需求分析报告.....	252
附录 C	项目开发计划.....	253
附录 D	概要设计说明书	255
附录 E	详细设计说明书.....	257
附录 F	用户操作手册	258
附录 G	测试计划	260

附录 H 测试分析报告	262
附录 I 程序维护手册	263
附录 J 总结性报告	268
参考文献	269
参考网站	272

第 1 章 导 论

1.1 软件工程

1.1.1 软件工程定义

20 世纪 60 年代以来, 随着计算机的广泛应用, 计算机软件的数量以惊人的速度剧增, 软件的规模越来越庞大, 软件生产率、软件质量远远满足不了实际需要, 成为社会发展和经济发展的制约因素。与软件相关的问题突出表现为开发进度延误、费用剧增、质量低下、错误频出、灵活性差、生产率低下等等。同时, 由于微电子技术的进步, 计算机硬件成本每五年下降 2~3 个数量级, 且质量稳步提高。计算机软硬件之间的这种不平衡发展使得软件相关问题的解决尤为迫切。科学家们把软件开发和维护过程中遇到的一系列问题统称为“软件危机”。导致软件危机的因素很多, 除了软件本身固有的特征外, 还与软件开发模式、软件设计方法、软件开发支持以及软件开发管理等有关。

为解决软件开发和维护过程中遇到的一系列问题, 在 1968 年北大西洋公约组织的工作会议上, 许多程序员、计算机科学家与工业界人士聚集在一起共商对策。通过借鉴传统工业的成功做法, 他们主张用工程化的方法开发软件来解决软件危机, 并提出“软件工程”这一术语。由此, 指导计算机软件开发和维护的一门工程科学——软件工程学逐步形成。

软件是与一个系统, 特别是一个计算机系统有关的程序、过程和有关文档的完整集合。工程是科学和数学的应用, 通过这一应用, 使得自然界的物质和能源的特性通过各种结构、机器、产品、系统和过程成为对人类有用的东西。在软件和工程这两个定义的基础上, 软件工程可定义如下:

软件工程是一类求解软件的工程。它应用计算机科学、数学以及管理科学等原理、借鉴传统工程的原则、方法, 创建软件以达到提高质量、降低成本的目的, 使计算机设备的能力借助于软件成为对人类有用的东西。其中, 计算机科学、数学用于构造模型和算法, 工程科学用于制定规范、设计模式、评估成本及确定权衡, 管理科学用于计划、资源、质量、成本等管理。

1.1.2 软件工程框架

软件工程作为一种工程, 同其他工程项目一样, 有自己的目标、活动和原则。软件工程的框架如图 1.1 所示。

1. 软件工程项目目标

软件项目的目标是“生产正确、可用及具有经济效益的产品”。正确性指软件产品达到

预期功能的程度。可用性指软件基本结构、实现、文档为用户可用的程度。具有经济效益指软件开发、运行的整个开销满足用户要求的程度。这些目标的实现还需要解决很多理论和实践的问题，它们形成了对过程、过程模型及工程方法选取的约束。

2. 软件工程活动

软件工程的**活动**是指生产一个最终满足需求且达到工程目标的软件产品所需要的步骤,主要包括问题定义、可行性研究、需求分析、设计、实现、确认、支持等。

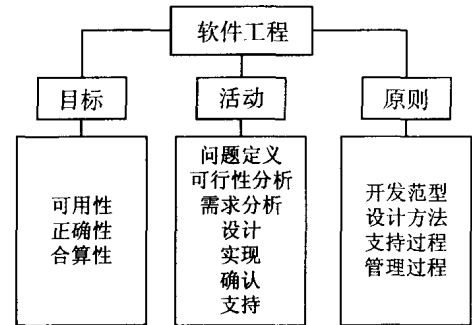


图 1.1 软件工程框架

(1) 问题定义

问题定义阶段的任务是明确要解决的问题。如果在不知道问题是什么的情况下就试图解决它，则最终得出的结果很可能是毫无意义的。尽管确切定义问题的必要性是很显然的，但在实践中它却是最容易被忽视的一个步骤。在问题定义阶段，系统分析员通过对系统的实际用户和使用部门负责人的访问调查，提出关于问题性质、工程目标和规模的书面报告，并在用户和使用部门负责人会议上认真讨论这份书面报告，澄清含糊不清的地方，改正理解不正确的地方，最后得到一份双方都满意的文档。

(2) 可行性研究

可行性研究阶段的任务是要回答这样一个问题：对于问题定义阶段所确定的问题是否有可行的解决办法。为解决该问题，系统分析员需要进行一次大大压缩和简化了的系统分析和设计的过程，也就是在较抽象的高层次上进行的分析和设计过程。可行性研究应该比较简短，因为它不是具体地去解决问题，而是研究问题的范围，探索这个问题是否值得去解决以及是否有可行的解决办法。

问题定义阶段提出的对工程目标和规模的报告通常比较含糊，而可行性研究阶段应该导出系统的高层逻辑模型，在此基础上更准确、更具体地确定工程规模和目标，从而分析员可以更准确地估计系统的成本和效益。对建议的系统进行仔细的成本和效益分析是这个阶段的主要任务之一。

可行性研究的结果是决定工程继续与否的重要依据。一般来说，只有投资可能取得较大效益的那些软件项目才有必要继续进行下去。可行性研究之后的那些阶段将需要投入更多的人力和物力，适时中止不值得投资的软件项目，可以避免更大的浪费。

(3) 需求分析

需求分析阶段的任务是确定为解决该问题，目标系统必须具备哪些功能。用户了解他们所面对的问题，知道必须做什么，但是通常不能完整准确地表达出他们的要求，更不知道怎样利用计算机解决他们的问题。软件开发人员知道怎样使用软件实现人们的要求，但是对特定用户的具体要求并不完全清楚。因此系统分析员在需求分析阶段必须和用户密切配合，充分交流信息，以得出经过用户确认的系统逻辑模型。用于描述系统逻辑模型的方法有数据流图、数据字典和简要的算法等。

需求分析阶段确定的系统逻辑模型是以后设计和实现目标系统的基础，因此必须准确完整地体现用户的要求。系统分析员通常都是计算机软件专家、技术专家，他们一般都喜欢很快着手进行具体设计，然而一旦分析员开始谈论程序设计的细节时，就会脱离用户，使他们不能继续提出自己的要求和建议。软件工程使用的结构分析设计的方法为每个阶段都规定了特定的结束标准，需求分析阶段必须提供系统逻辑模型，经过用户确认之后才能进入下一个阶段，这就可以有效地防止和克服急于着手进行具体设计的倾向。

(4) 设计

设计活动包含两个子活动，即总体设计和详细设计。

① 总体设计

总体设计阶段是从概括的层面上探讨如何解决问题。在该阶段，对同一问题通常考虑若干种解决方案，如，目标系统的一些主要功能是用计算机自动完成还是用人工完成；如果使用计算机，那么是使用批处理方式还是人机交互方式；信息存储使用传统的文件系统还是数据库等。一般来说，总体设计阶段权衡的方案包括：

- 低成本的解决方案，即系统只能完成最必要的工作，不能多做一点额外的工作。
- 中等成本的解决方案，即系统不仅能够很好地完成预定任务，而且可能还具有用户没有具体指定的某些功能和特点。这些附加的功能和特点是系统分析员根据自己的知识和经验所确定的，且在实践中将被证明是很有价值的。
- 高成本的完美方案，即系统具有用户可能希望的所有功能和特点。

系统分析员应该使用系统流程图或其他工具描述每种可能的系统，估计每种方案的成本和效益，还应该在充分权衡各种方案的基础上，推荐一个较好的系统，并且制定实现所推荐系统的详细计划。如果用户接受分析员推荐的系统，则可以着手完成本阶段的另一项主要工作，即详细设计。

② 详细设计

总体设计阶段以比较抽象概括的方式提出了解决问题的办法。详细设计阶段的任务则是把解决方案具体化，也就是确定具体实现这个系统的方法。但是，该阶段的任务仍然不是编写程序，而是设计出程序的详细规格说明。这种规格说明的作用类似于其他工程领域中工程师经常使用的工程蓝图，它们应该包含必要的细节，程序员可以根据它们写出实际的程序代码。通常用 HIPO 图（层次图加输入/处理/输出图）或 PDL 语言（过程设计语言）描述详细设计的结果。

(5) 实现

实现阶段的关键任务是问题的具体解决，即写出正确的、容易理解和维护的程序模块。程序员应该根据目标系统的性质和实际环境，选取一种适当的高级程序设计语言（必要时用汇编语言），把详细设计的结果翻译成用选定的语言书写的程序。实现过程往往伴随着单元测试，即详细测试编写出的每一个模块。

(6) 确认

确认活动是综合测试的过程，关键任务是通过各种类型的测试使软件达到预定的要求。最基本的测试是集成测试和验收测试。集成测试是根据设计的软件结构，把经过单元测试检验的模块按选定的某种策略装配起来，在装配过程中对程序进行必要的测试。验收测试则是按照在需求分析阶段确定的规格说明书，由用户或在用户积极参与下对目标系统进行

验收。必要时还可以通过现场测试或平行运行等方法对目标系统进一步测试检验。

为了使用户能够积极参加验收测试，并且在系统投入生产性运行以后能够正确有效地使用这个系统，通常需要以正式的或非正式的方式对用户进行培训。

(7) 支持

支持活动是指软件的维护。维护阶段的关键任务是，通过各种必要的维护活动使系统持久地满足用户的需要。通常有四类维护活动：

- 改正性维护，即诊断和改正在使用过程中发现的软件错误。
- 适应性维护，即修改软件以适应环境的变化。
- 完善性维护，即根据用户的要求改进或扩充软件使它更完善。
- 预防性维护，即修改软件为将来的维护活动预先做准备。

虽然没有把维护阶段进一步划分成更小的阶段，但是实际上每一项维护活动都应该经过提出维护要求或报告问题、分析维护要求、提出维护方案、审批维护方案、确定维护计划、修改软件设计、修改程序、测试程序、复检验收等一系列步骤，因此实质上是经历了一次压缩和简化了的软件定义和开发的全过程。

3. 软件工程原则

围绕工程设计、工程支持及工程管理，下面列出了软件工程的四条基本原则。

(1) 选取适宜的开发模型

该原则与系统设计有关。在系统设计中，软件需求、硬件需求以及其他因素之间是相互制约、相互影响的，经常需要权衡。因此，必须认识需求定义的易变性，采用适宜的开发模型予以控制，以保证软件产品满足用户的要求。

(2) 采用合适的设计方法

在软件设计中，通常要考虑软件的模块化、抽象与信息隐蔽、局部化、一致性、适应性等特征。合适的设计方法有助于这些特征的实现，以达到软件工程的目标。

(3) 提供高质量的工程支持

“工欲善其事，必先利其器”。在软件工程中，软件工具与环境对软件过程的支持颇为重要。软件工程项目的质量与开销直接取决于对软件工程所提供的支撑质量和效用。

(4) 重视开发过程的管理

软件工程的管理，直接影响可用资源的有效利用、生产满足目标的软件产品、提高软件组织的生产能力等问题。因此，只有对软件过程进行有效的管理，才能实现有效的软件工程。

总之，软件工程框架告诉我们：软件工程目标是可用性、正确性和合算性；实施一个软件工程要选取适宜的开发模型、要采用合适的设计方法、要提供高质量的工程支撑、要实行开发过程的有效管理；软件工程活动主要包括问题定义、可行性研究、需求分析、设计、实现、确认、支持等活动，每一活动可根据特定的软件工程，采用合适的开发模型、设计方法、支持过程、过程管理。从中也可以看出，管理在软件工程中是个不可或缺的概念。

1.1.3 软件工程模型

如前文所述，软件工程的演化过程包括问题定义、可行性研究、需求分析、设计、实现、确认、支持等；所有这些活动都必须进行管理。软件项目管理贯穿于软件工程的演化过程之中，如图 1.2 所示。

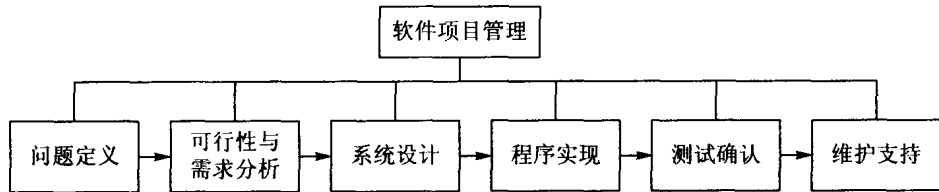


图 1.2 软件工程的演化过程

为更好地管理软件工程的演化过程，业界开发出了许多组织软件工程活动的方法，称为软件工程模型。软件工程模型用一定的流程将各个活动连接起来，并可用规范的方式操作全过程，如同工厂的生产线。常见的软件工程模型有线性模型、快速原型模型、螺旋模型、渐增式模型等。

线性模型是最早出现的软件工程模型，也称作瀑布模型，如图 1.3 所示。线性模型在软件开发过程中做出了重要贡献，但有其固有的问题。Frederick P. Brooks Jr.指出，线性模型有两点不足：

- 假设项目只经历一次过程，而且体系结构出色并易于使用，设计合理可靠，随着测试的进行，编码实现是可以修改和调整的。也就是说，线性模型假设所有错误发生在编码实现阶段，因此错误修复只在单元测试和系统测试中就可以完成。
- 假设整个系统一次性地被构建，在所有的设计、大部分编码、部分单元测试完成之后，才把各部分合并起来，进而开始后续工作。

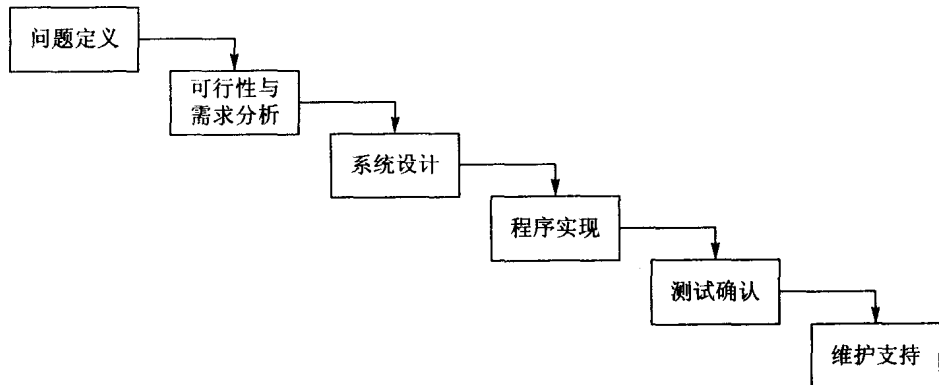


图 1.3 软件工程的线性模型

虽然线性模型有其不足，但“线性”是人们最容易掌握并能熟练应用的思想方法。当面对一个复杂的“非线性”问题时，人们总是千方百计地将其分解或转化为一系列简单的线性问题，然后逐个解决。一个软件系统的整体可能是复杂的，单个子程序却总是简单的，

可以用线性的方式来实现。

软件开发的其它模型或多或少都是在线性模型的基础上发展起来的。它们克服了线性模型的一些不足，在软件工程领域找到了自己的一席之地。如螺旋模型可以看成是连接的线性模型（如图 1.4 所示），而目前计算机工业中应用广泛的渐增式模型则可以看成是分段的线性模型（如图 1.5 所示）。

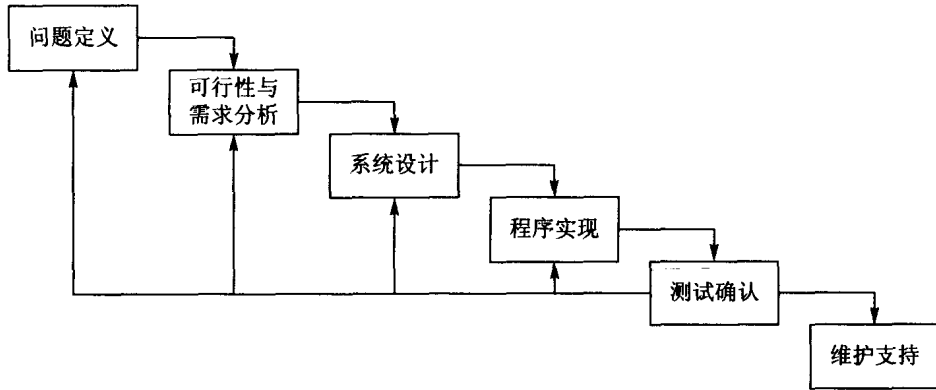


图 1.4 软件工程的螺旋模型

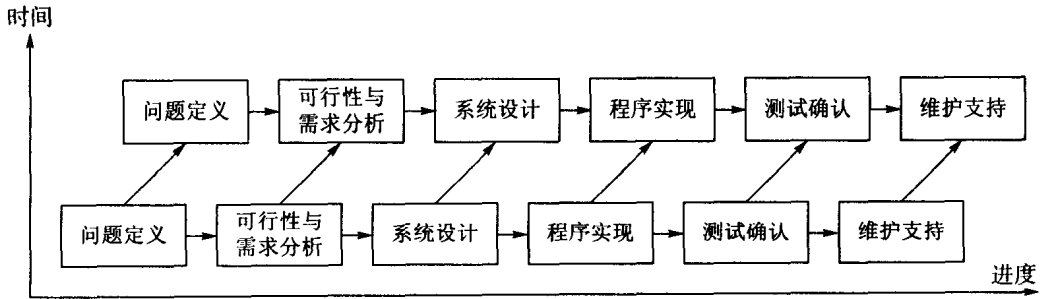


图 1.5 软件工程的渐增式模型

渐增式模型首先构建系统的基本轮询回路，为每个功能都提供了子函数调用，但只是空的子函数，如图 1.6 所示。此时的基本轮询回路不完成任何事情，只保证可以正常运行。接下来，逐次添加功能，逐步开发和增加相应模块。每个功能都基本上可以运行之后，逐个精化或重写模块——增量的开发整个系统。有时候可能会出现需要修改原有驱动回路、甚至是回路模块接口的情况，但在每个阶段，都有一个可运行的系统。因此，渐增式模型可以很早就开始用户测试，也可以采用按预算开发的策略，即在保证必要功能特性的基础上，确保不出现进度或者预算超支的情况。Microsoft 的“每晚重建 (Nightly Build)”方法实际是一种逻辑上的渐增式开发。

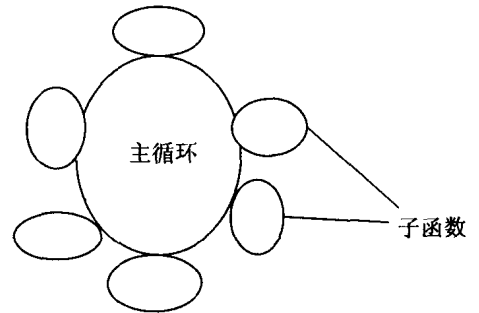


图 1.6 系统的基本轮询回路

这里不再介绍其他的软件工程模型了，明白