21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

软件工程（第4版）学习辅导与习题解析

Learning Guidance and Analysis of Exercises for Software Engineering (Version 4)

张海藩 吕云翔 编著

- 简明扼要地复习软件工程的重点内容
- 习题丰富多样且解析兼顾重点和难点
- 课程设计指导利于理论与实践的结合
- 模拟试题有助于自测和检验学习效果

名家系列

人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

软件工程（第4版）学习辅导与习题解析

Learning Guidance and Analysis of Exercises for Software Engineering (Version 4)

张海藩 吕云翔 编著

名家系列

人民邮电出版社

北京

图书在版编目（CIP）数据

软件工程（第4版）学习辅导与习题解析 / 张海藩，吕云翔编著. -- 北京 : 人民邮电出版社，2013.12
21世纪高等学校计算机规划教材
ISBN 978-7-115-33074-1

Ⅰ. ①软… Ⅱ. ①张… ②吕… Ⅲ. ①软件工程－高等学校－教学参考资料 Ⅳ. ①TP311.5

中国版本图书馆CIP数据核字(2013)第230217号

内 容 提 要

本书是《软件工程（第4版）》的配套教材。全书共分为16章。每章由3部分组成：第1部分简明扼要地复习每一章的重点内容；第2部分给出原教材每一章后的习题；第3部分是习题解析，不是简单地给出答案，而是仔细分析题目，讲解解题思路，从而有助于读者举一反三，学会用软件工程方法学分析问题和解决问题。

本书还提供了3个附录，分别给出了综合应用题解析、课程设计指导和模拟试题与参考答案。

本书既可以与《软件工程（第4版）》配合使用，也可以供学习软件工程的读者单独使用（包括参加计算机等级考试或相关专业自学考试），以加深对所学内容的理解并检验学习效果。

◆ 编　　著　张海藩　吕云翔
责任编辑　武恩玉
责任印制　彭志环　杨林杰

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路11号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
大厂聚鑫印刷有限责任公司印刷

◆ 开本：787×1092　1/16
印张：14　　2013年12月第1版
字数：347千字　　2013年12月河北第1次印刷

定价：35.00元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京崇工商广字第0021号

前　言

软件工程是应用计算机科学技术、数学、管理学的原理，运用工程科学的理论、方法和技术，研究和指导软件开发和演化的一门交叉学科。随着科技的发展，软件工程已成为计算机科学及其相关专业的一门重要的必修课。其教学目的在于使学生掌握软件工程的基本概念和原则，培养学生使用工程化的方法高效地开发高质量软件的能力，以及进行项目管理的能力。

《软件工程》已经出版了3个版本，颇受读者欢迎。为配合《软件工程（第4版）》的出版，作者编写了这本教材的配套教材《软件工程（第4版）学习辅导与习题解析》。

本书共分为16章。第1章、第2章讲述软件工程与软件过程；第3章、第4章、第5章讲述传统方法学，包括结构化分析、设计与实现；第6章、第7章、第8章、第9章、第10章讲述面向对象方法学，包括面向对象的概念、模型、分析、设计、实现，同时介绍了统一建模语言UML；第11章、第12章、第13章、第14章讲述软件项目管理，包括软件项目的计划、组织和控制，软件维护与软件文档；第15章、第16章讲述软件工程的高级课题，包括形式化方法和软件重用。

每章由3部分组成：第1部分简明扼要地复习每一章的重点内容；第2部分给出原教材每一章后的习题；第3部分是习题解析，不是简单地给出答案，而是仔细分析题目，讲解解题思路，从而有助于读者举一反三，学会用软件工程方法学分析问题和解决问题。

附录A是综合应用题解析。众所周知，软件工程习题的难点不在于填空题、选择题、判断正误题、名词解释和简答题，而在于应用题。也就是说，能够做好应用题才能够真正理解软件工程的一些概念、原理。附录A提供的应用题具有一定的深度和广度，可以扩大读者的视野，这在一般的书上是看不到的。

附录B是课程设计指导。指导中主要讲述了如何进行项目选题、组建团队、团队工作方式和项目进度安排。

附录C是模拟试题，共给出了3份试卷，每份试卷都有相应的参考答案。读者可以用这些试题自我测试，检验学习效果。

感谢在成书过程中所有给予过帮助的教师和学生。

由于软件工程是一门新兴学科，软件工程的教学方法本身还在探索之中，加之我们的水平和能力有限，书中难免有疏漏之处，恳请各位同仁和广大读者给予批评指正，也希望各位能将实践过程中的经验和心得与我们交流（yunxianglu@hotmail.com）。

编　者

2013年6月

目　录

第1章 软件工程概述

软件工程的研究领域包括软件的开发方法、软件的生命周期以及软件的工程实践等。

1.1 软件危机与软件工程的起源

1.1.1 软件危机的出现

随着计算机应用的日益普及，软件的数量也开始急剧膨胀。在程序运行时发现的错误必须设法改正；用户有了新的需求时必须相应地修改程序；硬件或操作系统更新时，通常需要修改程序以适应新的环境。更严重的是，许多程序的个体化特性使得它们最终成为不可维护的。“软件危机”就这样开始出现了。

1.1.2 软件危机介绍

软件危机是指在计算机软件的开发和维护过程中所遇到的一系列严重问题。软件危机包含下述两方面的问题：如何开发软件，以满足对软件日益增长的需求；如何维护数量不断膨胀的已有软件。

具体来说，软件危机主要有以下一些典型的表现。

- 对软件开发成本和进度的估计常常很不准确。
- 用户对“已完成的”软件系统不满意的现象经常发生。
- 软件产品的质量往往靠不住。
- 软件常常是不可维护的。
- 软件通常没有适当的文档资料。
- 软件成本在计算机系统总成本中所占的比例逐年上升。
- 软件开发生产率提高的速度，既跟不上硬件的发展速度，也远远跟不上计算机应用迅速普及深入的趋势。

1.1.3 产生软件危机的原因

由于软件缺乏“可见性”，因此管理和控制软件开发过程相当困难。软件维护通常意味着改正或修改原来的设计，这就在客观上使得软件较难维护。

软件的一个显著特点是规模庞大，而且程序复杂性将随着程序规模的增加而呈指数上升。目前，相当多的软件专业人员对软件开发和维护还有不少糊涂观念，在实践过程中或多或少地采用

了错误的方法和技术，这可能是使软件问题发展成软件危机的主要原因。

与软件开发和维护有关的许多错误认识和做法的形成，可以归于在计算机系统发展的早期阶段软件开发的个体化特点。错误的认识和做法主要表现为忽视软件需求分析的重要性、认为软件开发就是编写程序并设法使之运行、轻视软件维护等。

事实上，对用户要求没有完整准确的认识就匆忙着手编写程序是许多软件开发工程失败的主要原因之一。

编写程序只是软件开发过程中的一个阶段，而且在典型的软件开发工程中，编写程序所需的工作量只占软件开发全部工作量的10%～20%。

还必须认识到程序只是完整的软件产品的一个组成部分，在软件生命周期的每个阶段都要得出最终产品的一个或几个组成部分（这些组成部分通常以文档资料的形式存在）。也就是说，一个软件产品必须由一个完整的配置组成，软件配置主要包括程序、文档、数据等。

严重的问题是，在软件开发的不同阶段进行修改需要付出的代价是不尽相同的。

维护是极端艰巨复杂的工作，需要花费很大代价。软件工程学的一个重要目标就是提高软件的可维护性，减少软件维护的代价。

1.1.4 消除软件危机的途径

为了消除软件危机，首先应该对计算机软件有一个正确的认识。软件是程序、数据及相关文档的完整集合。其中，程序是能够完成预定功能和性能的可执行的指令序列；数据是使程序能够适当地处理信息的数据结构；文档是开发、使用和维护程序所需要的图文资料。

软件开发不是某种个体劳动的神秘技巧，而应该是一种组织良好、管理严密、各类人员协同配合、共同完成的工程项目。充分吸取和借鉴人类长期以来从事各种工程项目所积累的行之有效的原理、概念、技术和方法是必须的。

应该开发和使用更好的软件工具。总之，为了消除软件危机，既要有技术措施（方法和工具），又要有必要的组织管理措施。软件工程正是从管理和技术两方面研究如何更好地开发和维护计算机软件的一门新兴学科。

1.2 软 件 工 程

1.2.1 什么是软件工程

概括地说，软件工程是指导计算机软件开发和维护的工程学科。采用工程的概念、原理、技术和方法来开发与维护软件，把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来，经济地开发出高质量的软件并有效地维护它，这就是软件工程。

1.2.2 软件工程的基本原理

著名的软件工程专家Barry W. Boehm提出了软件工程的7条基本原理。

- 用分阶段的生命周期计划严格管理。
- 坚持进行阶段评审。
- 实行严格的产品控制。

- 采用现代程序设计技术。
- 结果应能清楚地审查。
- 开发小组的人员应该少而精。
- 承认不断改进软件工程实践的必要性。

1.3　软件工程包含的领域

IEEE 在 2004 年发布的《软件工程知识体系指南》中将软件工程知识体系划分为以下 10 个知识领域。

- 软件需求。
- 软件设计。
- 软件构建。
- 软件测试。
- 软件维护。
- 软件配置管理。
- 软件工程管理。
- 软件工程过程。
- 软件工程工具和方法。
- 软件质量。

习　　题

一、判断题

1. 软件就是程序，编写软件就是编写程序。（　　）
2. 软件危机的主要表现是软件需求增加，软件价格上升。（　　）
3. 软件工程学科出现的主要原因是软件危机的出现。（　　）
4. 与计算机科学的理论研究不同，软件工程是一门原理性学科。（　　）

二、选择题

1. 在下列选项中，(　　)不是软件的特征。
   A. 系统性与复制性　　B. 可靠性与一致性
   C. 抽象性与智能性　　D. 有形性与可控性
2. 软件危机的主要原因是(　　)。
   A. 软件工具落后　　B. 软件生产能力不足
   C. 对软件的认识不够　　D. 软件本身的特点及开发方法
3. 下列说法中正确的是(　　)。
   A. 20 世纪 50 年代提出了软件工程概念
   B. 20 世纪 60 年代提出了软件工程概念

C. 20世纪70年代出现了客户机/服务器技术

D. 20世纪80年代软件工程学科达到成熟

4. (　　)是将系统化的、规范的、可定量的方法应用于软件的开发、运行和维护的过程，它包括方法、工具和过程三个要素。

A. 软件生命周期　　B. 软件测试　　C. 软件工程　　D. 软件过程

5. 在下列选项中，(　　)不属于软件工程学科所要研究的基本内容。

A. 软件工程材料　　B. 软件工程目标

C. 软件工程原理　　D. 软件工程过程

6. 软件工程的三要素是(　　)。

A. 技术、方法和工具　　B. 方法、对象和类

C. 方法、工具和过程　　D. 过程、模型和方法

7. 用来辅助软件开发、运行、维护、管理、支持等过程中的活动的软件称为软件开发工具，通常也称为（　）工具。

A. CAD　　B. CAI　　C. CAM　　D. CASE

三、简答题

1. 与计算机硬件相比，计算机软件有哪些特点？
2. 软件就是程序吗？如何定义软件？
3. 什么是软件危机？什么原因导致了软件危机？
4. 为什么说软件工程的发展可以在一定程度上解决软件危机的各种弊端？
5. 请简述软件工程研究的内容。
6. 请简述软件工程的三要素。
7. 请简述软件工程的目标、过程和原则。
8. 请简述软件工程的基本原则。
9. 请简述现代软件工程与传统软件工程显著的区别和改进。

习题解析

一、判断题

1. ×　　2. ×　　3. √　　4. ×

解析：

2. 软件危机的主要表现是：开发人员开发的软件产品不能完全满足用户的需求；软件产品的质量难以得到保证；软件产品的开发周期、开发经费和维护费用很难被准确地估计，从而给项目的管理带来很多麻烦；已有的软件产品往往不能灵活地适应环境的改变，软件产品的可维护性、可扩展性和可复用性往往不能满足市场的要求。

4. 软件工程是一门工程性学科。

二、选择题

1. D　　2. D　　3. B　　4. C　　5. A　　6. C　　7. D

解析：

2. 软件危机出现的原因可以概括为以下几点：软件开发是一项复杂的工程，需要用科学的工

程化的思想来组织和指导软件开发的各个阶段；没有完善的质量保证体系；软件文档的重要性没有得到软件开发人员和用户的足够重视；从事软件开发的专业人员对这个产业的认识不充分，缺乏经验；软件独有的特点也给软件的开发和维护带来困难。

3. 20 世纪 60 年代出现了客户机 / 服务器技术，不是 70 年代。

三、简答题

1. 软件的特点如下。

- 无法直接观察计算机软件的物理形态，只能通过观察它的实际运行情况来了解它的功能、特性和质量等。
- 人们在分析、设计、开发、测试软件产品，以及在软件开发项目的管理过程中，渗透了大量的脑力劳动。
- 不存在像硬件一样的磨损和老化现象，但存在着缺陷维护和技术更新的问题。
- 软件的开发和运行必须依赖于特定的计算机系统环境。
- 具有可复用性。

2. 人们经过长期的实践已经逐步认识到，软件≠程序，程序只是软件的关键要素。普遍能被接受的观点是：软件 = 程序+数据+文档。

3. 软件危机的现象如下。

- 经费超出预算，项目一再拖延。
- 不重视需求，开发的软件不能满足用户的要求，项目成功率低。
- 没有规范的软件工程方法，软件可维护性差、软件质量差、可靠性差。
- 开发工具落后，手工方式，开发效率低。

所有导致软件危机的原因，都与软件本身的产品特点相关。

- 软件是一个复杂的逻辑产品。如果没有解决复杂问题的有效方法，以及软件产品的结构、质量、可维护性得不到保障，开发与维护费用就会持续升高。
- 软件产品不能实现大规模复用，这导致了软硬件生产效率的不同。
- 软件生产是脑力劳动，它看不见、摸不着，开发成本、开发周期等都无法做到准确估算，生产过程不易控制。
- 软件成本主要是由研发成本构成；而硬件的生产成本主要是材料和制造成本，分摊的研发成本很少，即软件研发过程与硬件制造过程相比要复杂得多。

4. 软件工程的提出是为了解决软件危机所带来的各种弊端。具体地讲，软件工程的目标主要包括以下几点。

- 使软件开发的成本能够控制在预计的合理范围内。
- 使软件产品的各项功能和性能能够满足用户需求。
- 提高软件产品的质量。
- 提高软件产品的可靠性。
- 使生产出来的软件产品易于移植、维护、升级和使用。
- 使软件产品的开发周期能够控制在预计的合理时间范围内。

5. 软件工程研究的内容包括软件开发方法、软件开发模型、软件支持过程和软件管理过程。

- 软件开发方法的内容涵盖市场调研、正式立项、需求分析、项目策划、概要设计、详细设计、编程、测试、试运行、产品发布、用户培训、产品复制、销售、实施、系统维护、版本升级等。

- 常用的软件开发模型有瀑布模型、迭代模型、增量模型和原型模型等。
- 软件支持过程由所支持的CASE工具组成，常用的CASE工具有Power Designer和Rational Rose等。

6. 软件工程的三要素是从计算机科学的观点来看软件工程，即从软件开发的技术层面来看有过程、方法和工具这三要素。

- “过程”是软件产品加工所经历的一系列有组织的活动，保证能够合理、高质量和及时地开发出软件。
- “方法”为软件开发提供“如何做”的技术。它涵盖了项目计划、需求分析、系统设计、程序实现、测试与维护等一系列活动的做法，如经常说的面向结构、面向对象、面向组件等开发方法，以及项目管理中的估算、度量、计划等管理方法。软件工程以介绍方法为主。
- “工具”可为过程和方法提供自动的或半自动的支持。这些工具既包括软件，也包括硬件。软件工具包括编程、建模、管理等开发工具。通过网络环境把这些软件工具集成起来搭建一个能够支持团队开发的平台，称为计算机辅助软件工程，即CASE。CASE集成了软件、硬件和一个存放开发过程信息的软件工程数据库，形成了一个软件工程环境。

7. 目标、过程和原则是一切工程的三维框架，这里是以工程的观点来看待软件开发。

- 软件工程的目标：降低成本、及时交付高质量的软件产品（高质量、高效率、高效益）。
- 实现目标的过程即完成产品加工的过程，包括：基本过程、支持过程和组织过程。
- 进行过程应遵守的原则：原则就是过程中的轨道约束，包括：选取适宜的开发范型、采用合适的设计方法、提供高质量的工程支持、重视开发过程的管理。

8. 软件工程的基本原则如下。

- 将软件的生命周期划分为多个阶段，对各个阶段实行严格的项目管理。
- 坚持阶段评审制度，以确保软件产品的质量。
- 实施严格的产品控制，以适应软件规格的变更。在软件开发的过程中，用户需求很可能不断地发生变化。
- 采用现代的程序设计技术。
- 开发出来的软件产品应该能够清楚地被审查。
- 合理地安排软件开发小组的人员，并且开发小组的人员要少而精。
- 不断地改进软件工程实践。

9. 传统的软件工程是基于结构化的软件开发方法，而现代软件工程是以面向对象技术为标志。

- 不仅在编程技术上有很大的改进，而且在分析、设计等整个开发过程中，采用面向对象的思维方式，更加完整、自然地反映客观世界。
- 采用架构技术，开发效率、产品质量得到了极大提高。
- 更注重团队开发和管理，融入更多、更新的管理理念和手段，如RUP模型、XP模型、过程改进、能力成熟度模型、配置管理等。

第2章 软件过程

软件过程定义了运用方法的顺序、应该交付的文档资料、为保证软件质量和协调变化所需要采取的管理措施，以及标志软件开发各个阶段任务完成的里程碑。

2.1 软件生命周期的基本任务

概括地说，软件生命周期由软件定义、软件开发和运行维护等3个时期组成，每个时期又可进一步划分成若干个阶段。

软件定义时期的任务是确定软件开发工程必须完成的总目标；确定工程的可行性；导出实现工程目标应该采用的策略及系统必须完成的功能；估计完成该项工程需要的资源和成本，并且制订工程进度表。这个时期的工作通常又称为系统分析，由系统分析员负责完成。软件定义时期通常进一步划分为3个阶段，即问题定义、可行性研究和需求分析。

软件开发时期具体设计和实现在前一个时期定义的软件，它通常由下述4个阶段组成：概要设计、详细设计、编码和单元测试、综合测试。其中前两个阶段又称为系统设计，后两个阶段又称为系统实现。

运行维护时期的主要任务是使软件持久地满足用户的需要。具体地说，当软件在使用过程中发现错误时应该加以改正；当环境改变时应该修改软件以适应新的环境；当用户有新要求时应该及时改进软件以满足用户的新需要。通常对维护时期不再进一步划分阶段，但是每一次维护活动本质上都是一次压缩和简化了的定义和开发过程。

2.2 瀑布模型

在20世纪80年代之前，瀑布模型一直是唯一被广泛采用的生命周期模型。现在，它仍然是软件工程中应用最广泛的过程模型。

实际的瀑布模型是带“反馈环”的，如图2.1所示（图中实线箭头表示开发过程，虚线箭头表示维护过程）。

瀑布模型有许多优点：可强迫开发人员采用规范的方法（如结构化技术）；严格地规定了每个阶段必须提交的文档；要求每个阶段交出的所有产品都必须经过质量保证小组的仔细验证。

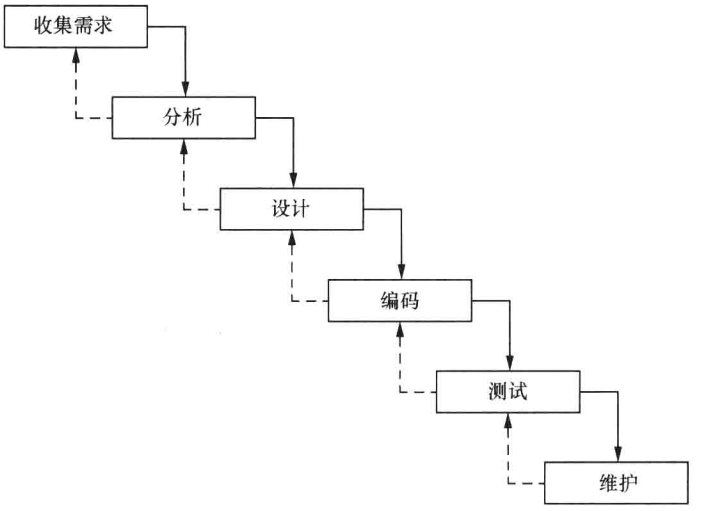

图 2.1　加入迭代过程的瀑布模型

“瀑布模型是由文档驱动的”这个事实也是它的一个主要缺点。在可运行的软件产品交付给用户之前，用户只能通过文档来了解产品是什么样的。要求用户不经过实践就提出完整准确的需求，在许多情况下都是不切实际的。总之，由于瀑布模型几乎完全依赖于书面的规格说明，很可能导致最终开发出的软件产品不能真正满足用户的需要。

2.3　快速原型模型

快速原型是快速建立起来的可以在计算机上运行的程序，它所能完成的功能往往是最终产品能完成的功能的一个子集。图 2.2 描述了快速原型模型（图中实线箭头表示开发过程，虚线箭头表示维护过程）。

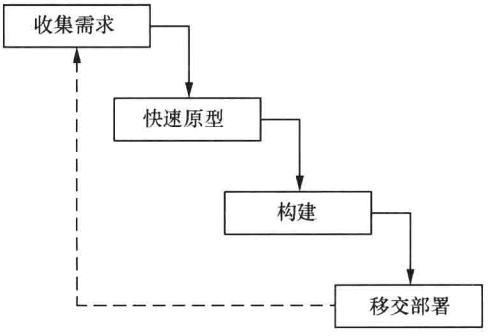

图 2.2　快速原型模型

快速原型模型是不带反馈环的，这正是这种过程模型的主要优点：软件产品的开发基本上是按线性顺序进行的。它的优点是有助于保证用户的真实需要得到满足。

2.4　增 量 模 型

增量模型也称为渐增模型，如图 2.3 所示。使用增量模型开发软件时，把软件产品作为一系列的增量构件来设计、编码、集成和测试。每个构件由多个相互作用的模块构成，并且能够完成特定的功能。

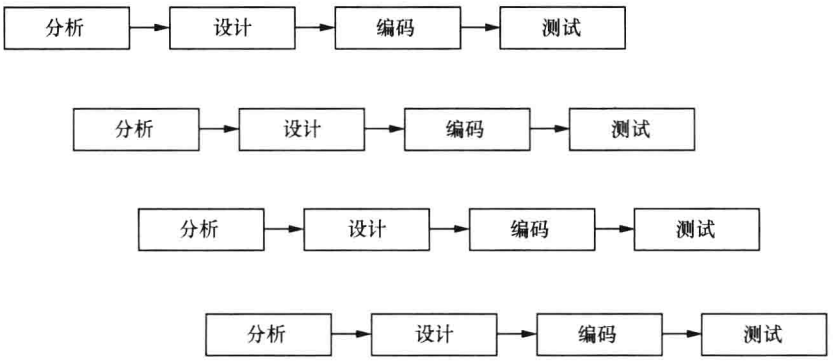

图 2.3　增量模型

能在较短时间内向用户提交可完成一些有用的工作的产品，是增量模型的一个优点。增量模型的另一个优点是，逐步增加产品功能可以使用户有较充裕的时间学习和适应新产品，从而减少一个全新的软件可能给客户组织带来的冲击。

使用增量模型的困难是，在把每个新的增量构件集成到现有软件体系结构中时，必须不破坏原来已经开发出的产品。此外，必须把软件的体系结构设计得便于按这种方式进行扩充，向现有产品中加入新构件的过程必须简单、方便。也就是说，软件体系结构必须是开放的。从长远观点看，具有开放结构的软件拥有真正的优势，这种软件的可维护性明显好于封闭结构的软件。尽管采用增量模型比采用瀑布模型和快速原型模型需要更精心的设计，但在设计阶段多付出的劳动将在维护阶段获得回报。如果一个设计非常灵活而且足够开放、足以支持增量模型，那么这样的设计将允许在不破坏产品的情况下进行维护。

2.5　螺 旋 模 型

在软件开发过程中必须及时识别和分析风险，并且采取适当措施以消除或减少风险的危害。

构建原型是一种能使某些类型的风险降至最低的方法。降低交付给用户的产品不能满足用户需要的风险。

螺旋模型的基本思想是，使用原型及其他方法来尽量降低风险。理解这种模型的一个简便方法，是把它看做在每个阶段之前都增加了风险分析过程的快速原型模型，如图 2.4 所示。