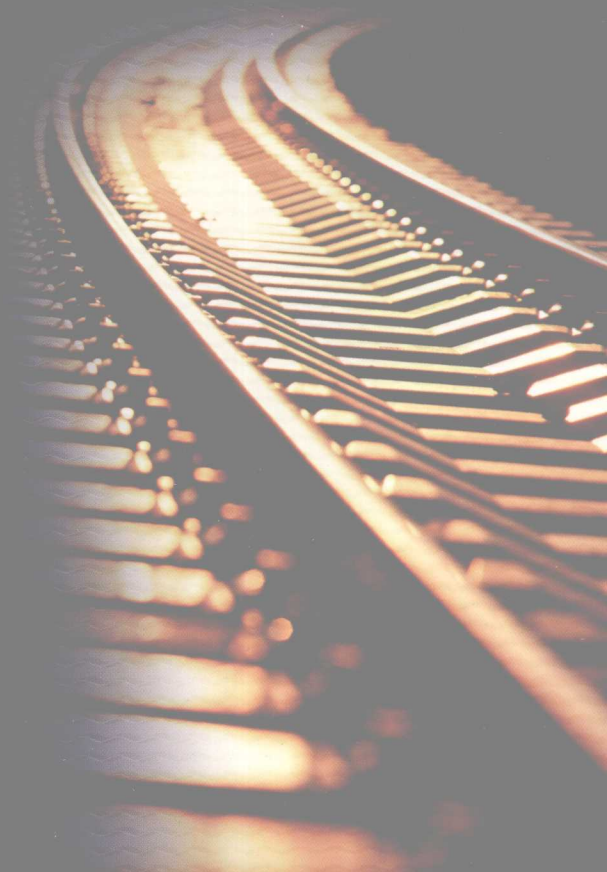
高等院校规划教材
软件工程系列

软件工程实践教程

刘冰 赖涵 瞿中 王化晶 编著



机械工业出版社
CHINA MACHINE PRESS

高等院校规划教材·软件工程系列

软件工程实践教程

刘 冰　赖 涵　瞿 中　王化晶　编著

机 械 工 业 出 版 社

本书从实用的角度出发，根据教育部高教司审定的《中国计算机科学与技术学科教程2002》中对软件工程的要求编写，并参照美国ACM和IEEE Computing Curricula 2001教程关于软件工程的描述，吸取了国内外软件工程的精华，详细介绍了软件工程、软件开发过程、软件计划、需求分析、总体设计、详细设计、编码、软件测试、软件维护、软件工程标准化和软件文档、软件工程质量、软件工程项目管理以及软件工程开发实例。各章均配有习题，以指导读者深入地进行学习，部分章后附有经典例题讲解和实验内容，帮助读者掌握相关知识。

本书既可作为高等学校计算机专业课程的教材或教学参考书，也可作为通信、电子信息、自动化等相关专业的计算机课程教材，还可供软件工程师、软件项目管理者和应用软件开发人员阅读参考。

图书在版编目(CIP)数据

软件工程实践教程/刘冰等编著. —北京：机械工业出版社，2009.1
(高等院校规划教材·软件工程系列)
ISBN 978-7-111-25458-4

Ⅰ.软… Ⅱ.刘… Ⅲ.软件工程-高等学校-教材 Ⅳ.TP311.5

中国版本图书馆CIP数据核字(2008)第165967号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)
责任编辑：唐德凯
责任印制：邓 博
北京四季青印刷厂印刷（三河市魏各庄装订二厂装订）

2009年1月第1版·第1次印刷
184mm×260mm·19.75印张·490千字
0001-3000册
标准书号：ISBN 978-7-111-25458-4
定价：32.00元

凡购本书，如有缺页，倒页，脱页，由本社发行部调换
销售服务热线电话：(010)68326294 68993821
购书热线电话：(010)88379639 88379641 88379643
编辑热线电话：(010)88379753 88379739
封面无防伪标均为盗版

出版说明

计算机技术的发展极大地促进了现代科学技术的发展，明显地加快了社会发展的进程。因此，各国都非常重视计算机教育。

近年来，随着我国信息化建设的全面推进和高等教育的蓬勃发展，高等院校的计算机教育模式也在不断改革，计算机学科的课程体系和教学内容趋于更加科学和合理，计算机教材建设逐渐成熟。在“十五”期间，机械工业出版社组织出版了大量计算机教材，包括“21世纪高等院校计算机教材系列”、“21世纪重点大学规划教材”、“高等院校计算机科学与技术‘十五’规划教材”、“21世纪高等院校应用型规划教材”等，均取得了可喜成果，其中多个品种的教材被评为国家级、省部级的精品教材。

为了进一步满足计算机教育的需求，机械工业出版社策划开发了“高等院校规划教材”。这套教材是在总结我社以往计算机教材出版经验的基础上策划的，同时借鉴了其他出版社同类教材的优点，对我社已有的计算机教材资源进行整合，旨在大幅提高教材质量。我们邀请多所高校的计算机专家、教师及教务部门针对此次计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了“高等院校规划教材”的体系架构与编写原则，以保证本套教材与各高等院校的办学层次、学科设置和人才培养模式等相匹配，满足其计算机教学的需要。

本套教材包括计算机科学与技术、软件工程、网络工程、信息管理与信息系统、计算机应用技术以及计算机基础教育等系列。其中，计算机科学与技术系列、软件工程系列、网络工程系列和信息管理与信息系统系列是针对高校相应专业方向的课程设置而组织编写的，体系完整，讲解透彻；计算机应用技术系列是针对计算机应用类课程而组织编写的，着重培养学生利用计算机技术解决实际问题的能力；计算机基础教育系列是为大学公共基础课层面的计算机基础教学而设计的，采用通俗易懂的方法讲解计算机的基础理论、常用技术及应用。

本套教材的内容源自致力于教学与科研一线的骨干教师与资深专家的实践经验和研究成果，融合了先进的教学理念，涵盖了计算机领域的核心理论和最新的应用技术，真正在教材体系、内容和方法上做到了创新。同时本套教材根据实际需要配有电子教案、实验指导或多媒体光盘等教学资源，实现了教材的“立体化”建设。本套教材将随着计算机技术的进步和计算机应用领域的扩展而及时改版，并及时吸纳新兴课程和特色课程的教材。我们将努力把这套教材打造成为国家级或省部级精品教材，为高等院校的计算机教育提供更好的服务。

对于本套教材的组织出版工作，希望计算机教育界的专家和老师能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前 言

当今，软件工程已成为计算机科学的一个分支。但随着软件产业不断发展的需求，传统的计算机学科逐步上升到计算科学。2001 年，IEEE 发布的计算学科教学规划把计算学科划分为计算机科学、计算机工程、软件工程、信息系统、信息技术和其他有待发展的学科等子学科，标志着软件工程这个名词作为与计算机理论相对应的各种软件实践技术的总称已经得到世界的公认。

20 世纪 90 年代以来，软件重用和软件构件技术成为研究热点，面向对象方法和技术成为软件开发的主流技术。软件工程知识为开发高品质的软件产品提供了理论和科学支撑，强调采用工程化的方式开发软件。这些知识支持以精确的方式描述软件工程产品，为产品及其相互关系的建模和推理提供了基础，并为可预测的设计过程提供了依据。

在编写本书时我们不仅重视理论知识的介绍，并且将实践与理论相结合，在教学内容上也作了较大调整。为了培养学生规范化的软件开发方法与技能，重点强调结构化开发方法中软件生命周期各阶段的活动和产品，同时也加大了对面向对象的系统分析与设计方法的介绍，并且计划将此部分内容与“面向对象分析与设计”、“UML 统一建模语言”、“软件测试”等课程教学内容相结合，使学生能够更深刻地理解这种先进的软件开发思想。

本书的编者都是长期在高校从事软件教学的教师，有丰富的教学经验和科研开发能力，并参阅了大量国内外有关软件工程的教材和资料，编写而成。其中基础理论部分由刘冰、王化晶编写，实验部分由赖涵编写，瞿中负责统稿及内容审定。

施佳、王波、代永亮、董玉莲、田秉玺等参与了文字录入和图表制作工作，本书的顺利出版，得到了领导和同事给予的大力支持和帮助，也得到了计算机教育界许多同仁的关心，在此一并致谢。

与本教材配套的电子教案和习题答案可从机械工业出版社网站上下载，网址为 www.cmpedu.com；或与作者联系，作者的邮箱为 cobly1837@sina.com。

目前，软件工程发展迅速，国内外有关软件工程技术与设计方面的资料很多，新理论、新技术层出不穷，由于编者水平有限，书中难免存在不妥和错误之处，恳请读者批评指正。

感谢您阅读本书。请将您的宝贵建议和意见发送至：jsjfw@mail.machineinfo.gov.cn。

作 者

目　录

第1章 基础知识

本章要点

- 软件
- 软件工程
- 软件工程学科
- Visio 绘图工具的介绍

1.1 概述

1.1.1 基本概念

1. 软件

“软件”这个词汇于20世纪60年代被首次提出。一个完整的计算机系统由软件和硬件组成,它们相互依存,缺一不可。IEEE 给软件的定义:软件是计算机程序、规程以及运行计算机系统可能需要的相关文档和数据。其中:

1）计算机程序是计算机设备可以接受的一系列指令和说明,为计算机的运行提供所需的功能和性能。

2）数据是事实、概念或指令的结构化表示,能够被计算机设备接收、理解或处理。

3）文档是描述程序研制过程、方法及使用的图文材料。

从软件的内容来说,软件更像是一种嵌入式的数字化知识,其形成是一个通过交互对话和抽象理解而不断演化的过程。

软件是一种特殊的产品,它具有如下特点。

1）复杂性:软件比任何其他人类制造的结构更复杂,甚至硬件的复杂性和软件相比也是微不足道的。软件本质上的复杂性使软件产品难以理解,影响软件过程的有序性和软件产品的可靠性,并使维护过程变得十分困难。

2）一致性:软件必须遵从人为的习惯并适应已有的技术和系统,软件需要随接口的不同而改变,随时间的推移而变化,而这些变化是不同的人设计的结果。许多复杂性来自保持与其他接口的一致,对软件的任何再设计,都无法简化这些复杂特性。

3）可变性:软件产品扎根于文化的母体中,如各种应用、用户、自然及社会规律、计算机硬件等,这些因素持续不断地发生着变化,而这些变化使软件随之变化。人们总是认为软件是很容易修改的,通常忽视了修改带来的副作用,即引入新的错误,造成故障率的升高。

4）不可见性:软件是客观世界和计算机之间的一种逻辑实体,不具有物理的形体特征。软件这种无法可视化的固有特性,剥夺了一些具有强大功能的概念工具的构造思路,不仅限制了个人的设计过程,也严重地阻碍了相互之间的交流。由于软件的不可见性,定义“需要做什么”成为软件开发的根本问题。

根据软件服务对象的范围不同,软件可以划分为通用软件和定制软件两种类型。

1）通用软件：通用软件是由软件开发组织开发、面向市场用户公开销售的独立运行系统，如操作系统、数据库系统、字处理软件、绘图软件包和项目管理工具等均属于这种类型。

2）定制软件：定制软件是由某个特定客户委托，软件开发组织在合同的约束下开发的软件，如企业 ERP 系统、卫星控制系统和空中交通指挥系统等都属于这种类型。

2. 软件工程

1968 年 10 月，北大西洋公约组织（North Atlantic Treaty Organisation，NATO）科学委员会在德国的加尔密斯（Garmisch）开会讨论软件可靠性与软件危机的问题，Fritz Bauer 首次提出了“软件工程”的概念。至今已经过去 40 年了，“软件工程”术语被广泛应用于工业、政府和学术界，但是人们对这个术语的含义依然存在着争论和不同观点。IEEE 给软件工程下的定义如下：

软件工程是：

1）将系统性的、规范化的、可定量的方法应用于软件的开发、运行和维护，即将工程化应用到软件上。

2）对 1）中所述方法的研究。

软件工程涉及计算机科学、工程科学、管理科学和数学等领域，是一门综合性的交叉学科。

软件工程的目标如下：

1）支付较低的开发成本。

2）达到要求的软件功能。

3）获取较好的软件性能。

4）开发的软件易于移植。

5）需要较低的维护费用。

6）能按时完成开发任务，及时交付使用。

7）开发的软件可靠性高。

3. 软件工程的发展

软件工程的发展共经历了 4 个阶段。

（1）第一阶段：控制机器（1956 ~ 1967）

在计算机发展的早期阶段，计算机的主要用途是用于快速计算，出现了以 Ada、Fortran 等编程语言为标志的算法技术。这个阶段可以说是软件工程的史前时代，程序设计被认为是一种任人发挥创造才能的活动，不需要系统化的方法和开发管理，程序的质量完全依赖于程序员个人的技巧。在 20 世纪 60 年代末期，出现了软件产品和“软件作坊”的概念，设计人员开发程序不再像早期那样只为自己的研究工作需要，而是为了用户能更好地使用计算机，人们已经开始认识到软件开发不仅仅是编码。

（2）第二阶段：控制过程（1968 ~ 1982）

计算机应用开始涉及各种以非数值计算为特征的商业事务领域，交互技术、多用户操作系统、数据库系统等随之发展起来，出现了以 Pascal、C ++、Java 等编程语言和关系数据库管理系统为标志的结构化软件技术。软件危机已经爆发，人们开始使用工程化的方法和原则来解决软件开发中的问题。软件工程成为一个研究领域，软件的工作范围从只考虑程序的编写扩展到从定义、编码、测试到使用、维护等整个软件生命周期，瀑布模型被广泛使用。

（3）第三阶段：控制复杂性（1983 ~ 1995）

微处理器的出现与应用使计算机真正成为大众化的产品，而软件系统的规模、复杂性以及

在关键领域的广泛应用，促进了软件开发过程的管理及工程化开发。在这一时期，计算机辅助软件工程（Computer Aided Software Engineering，CASE）及其相应的集成工具大量出现，软件开发技术中的度量问题受到重视，出现了著名的软件工作量估计结构性成本估算模型（Constructive Cost Model，COCOMO）、软件过程改进模型（Capability Maturity Model，CMM）等。20 世纪 80 年代后期，以 Smalltalk、C ++ 等为代表的面向对象技术重新崛起，传统的结构化技术受到了严峻的考验。

（4）第四阶段：开放式的软件工程（1996 ~ 至今）

Internet 技术的迅速发展使软件系统从封闭走向开放，Web 应用成为人们在 Internet 上最主要的应用模式，异构环境下分布式软件的开发成为一种主流需求，软件复用和构件技术成为技术热点，需求分析、软件过程、软件体系结构等方面的研究也取得了有影响的成果。

1.1.2 软件危机

软件危机是指在计算机软件的开发和维护过程中遇到的一系列严重问题，如软件费用、软件可靠性、软件维护、软件生产率、软件重用等。软件危机在 20 世纪 60 年代末全面爆发，虽然软件开发的新工具和新方法层出不穷，但是至今软件危机依然没有消除。

软件危机主要表现在以下几个方面：

1）软件开发的成本和进度难以准确估计，延迟交付，甚至取消项目的现象屡见不鲜。

2）软件存在着错误多、性能低、不可靠、不安全等质量问题。

3）软件维护极其困难，而且很难适应不断变化的用户需求和使用环境。

软件危机产生的原因有以下几个方面：

1）软件的规模越来越大，结构越来越复杂。

2）软件开发管理困难而复杂。

3）软件开发费用不断增加。

4）软件开发技术落后。

5）生产方式落后。

6）开发工具落后，生产提高缓慢。

1.2 软件生存周期和软件过程

1.2.1 软件生存周期

软件生存周期是指一个软件从提出需求开始直到该软件报废为止的整个时期。通常，软件生存周期包括可行性分析和项目开发计划、需求分析、概要设计、详细设计、编码、测试、维护等活动，可以将这些活动以适当方式分配到不同阶段去完成。

（1）可行性分析和项目开发计划

这个阶段的任务是明确“要解决的问题是什么”，“解决问题的办法和费用”，“解决问题所需的资源和时间”。要回答这些问题，就要进行问题定义、可行性分析、制定项目开发计划。

（2）需求分析

这个阶段的任务是准确地确定软件系统必须做什么，确定软件系统具备哪些功能，写出软

件需求规格说明书。

(3) 概要设计

这个阶段的任务是把软件需求规格说明书中确定的各项功能转换成需要的体系结构。

(4) 详细设计

这个阶段的任务是为每个模块完成的功能进行具体描述,把功能描述转变为精确的、结构化的过程描述。

(5) 编码

这个阶段的任务是把每个模块的控制结构转换成计算机可接受的程序代码。

(6) 测试

这个阶段的任务是运行程序,从而发现程序中的错误。

(7) 维护

这个阶段的任务是诊断和修改软件,以识别和纠正软件中存在的错误,去掉软件性能上的缺陷,排除实施过程中的错误操作等。

1.2.2 软件开发过程模型

1. 软件开发过程

软件开发过程是指软件工程人员为了获得软件产品在软件工具支持下实施的一系列软件工程活动。

软件开发过程应该明确定义以下元素:

1) 过程中所执行的活动及其顺序关系。

2) 每一个活动的内容和步骤。

3) 团队成员的工作和职责。

软件开发过程一共包括 7 个过程,即获取过程、供应过程、开发过程、操作过程、维护过程、管理过程和支持过程。

2. 软件开发过程模型

目前,常见的软件开发过程模型包括瀑布模型、快速原型模型、增量模型、喷泉模型、螺旋模型、形式化方法模型、基于构件的开发模型和基于知识的模型等。

(1) 瀑布模型

瀑布模型将软件开发过程划分为需求定义与分析、软件设计、软件实现、软件测试和运行维护等一系列基本活动,并且规定这些活动自上而下、相互衔接的固定次序。瀑布模型如图 1-1 所示。该模型支持结构化的设计方法,但它是一种理想的线性开发模式,缺乏灵活性,无法解决软件需求不明确或不准确的问题。

瀑布模型的优点如下:

1) 严格规范软件开发过程,克服了非结构化的编码和修改过程的缺点。

2) 强调文档的作用,要求每个阶段都要仔细验证。

瀑布模型的缺点如下:

1) 各个阶段的划分固定,需要文档记录各阶段的内容,极大地增加了工作量。

2) 由于瀑布模型是线性的,用户只有等到整个过程的末期才能见到开发成果,中间提出的变更要求很难响应。

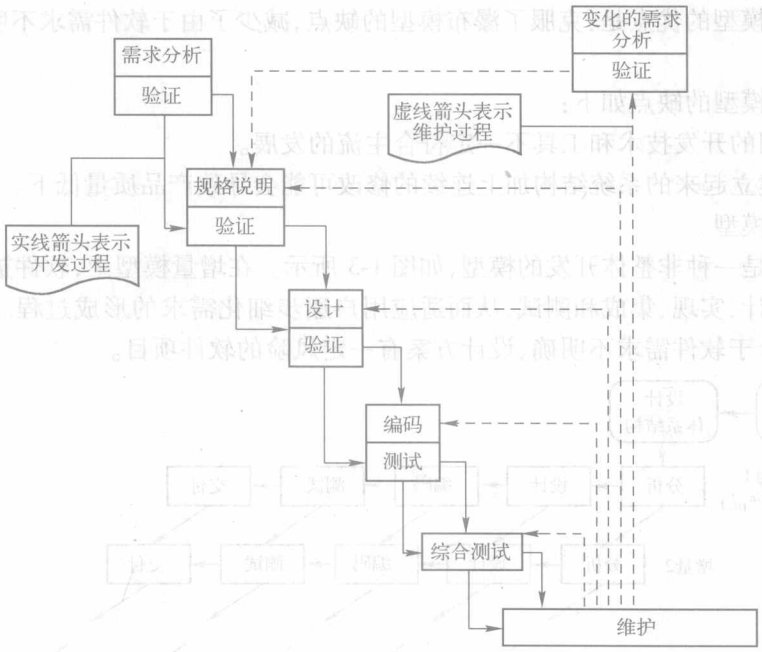


图 1-1　瀑布模型

3）早期的错误可能要等到开发后期的测试阶段才能发现，进而带来严重的后果。

（2）快速原型模型

快速原型模型需要迅速建造一个可以运行的软件原型，以便理解和澄清问题，使开发人员与用户达成共识，最终在确定的客户需求基础上开发客户满意的软件产品。快速原型模型如图 1-2 所示。

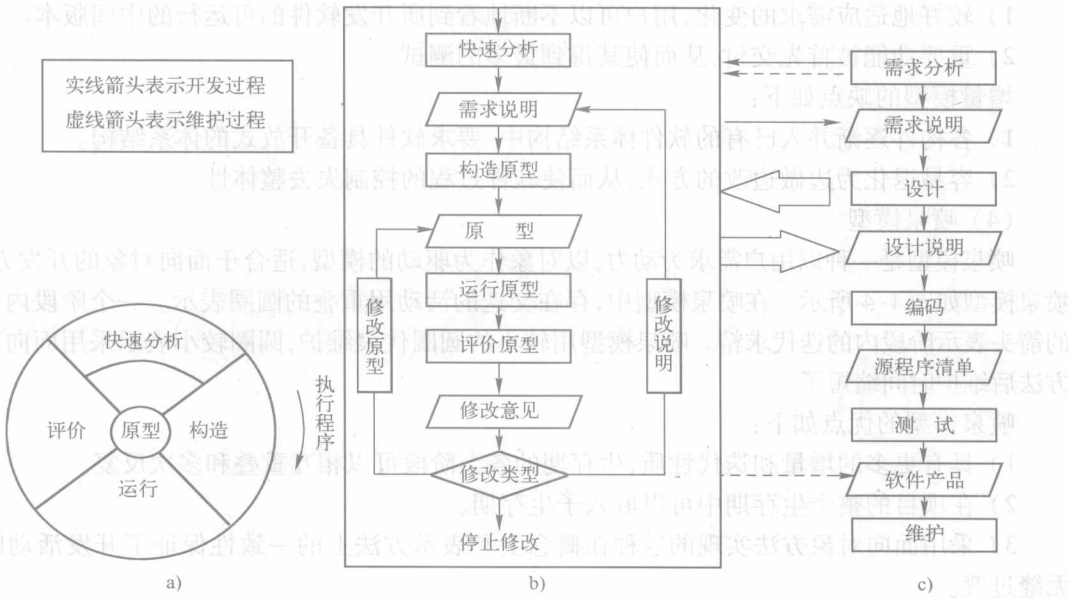


图 1-2　快速原型模型

a）原型表示　b）原型使用　c）开发过程

快速原型模型的优点是：克服了瀑布模型的缺点，减少了由于软件需求不明确所带来的开发风险。

快速原型模型的缺点如下：

1）所选用的开发技术和工具不一定符合主流的发展。

2）快速建立起来的系统结构加上连续的修改可能会导致产品质量低下。

（3）增量模型

增量模型是一种非整体开发的模型，如图 1-3 所示。在增量模型中，软件被作为一系列的增量构件来设计、实现、集成和测试，从而适应用户逐步细化需求的形成过程。该模型有较大的灵活性，适合于软件需求不明确、设计方案有一定风险的软件项目。

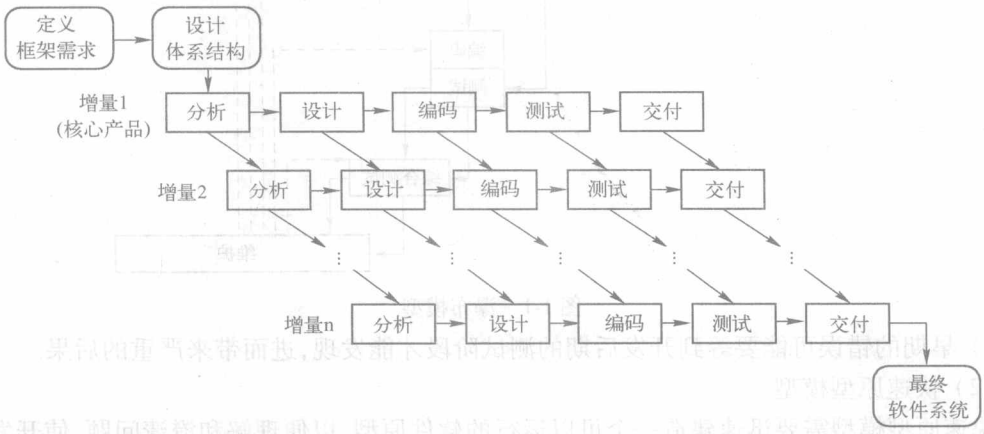


图 1-3　增量模型

增量模型的优点如下：

1）较好地适应需求的变化，用户可以不断地看到所开发软件的可运行的中间版本。

2）重要功能被首先交付，从而使其得到最多的测试。

增量模型的缺点如下：

1）各构件逐渐并入已有的软件体系结构中，要求软件具备开放式的体系结构。

2）容易退化为边做边改的方式，从而使软件过程的控制失去整体性。

（4）喷泉模型

喷泉模型是一种以用户需求为动力，以对象作为驱动的模型，适合于面向对象的开发方法。喷泉模型如图 1-4 所示。在喷泉模型中，存在交迭的活动用重叠的圆圈表示。一个阶段内向下的箭头表示阶段内的迭代求精。喷泉模型用较小的圆圈代表维护，圆圈较小表示采用面向对象方法后维护时间缩短了。

喷泉模型的优点如下：

1）具有更多的增量和迭代性质，生存期的各个阶段可以相互重叠和多次反复。

2）在项目的整个生存期中可以嵌入子生存期。

3）采用面向对象方法实现的这种在概念上和表示方法上的一致性保证了开发活动间的无缝过渡。

喷泉模型的缺点如下：

1）面向对象方法要求经常对开发活动进行迭代，这就有可能造成在使用喷泉模型的开发

过程过于无序。

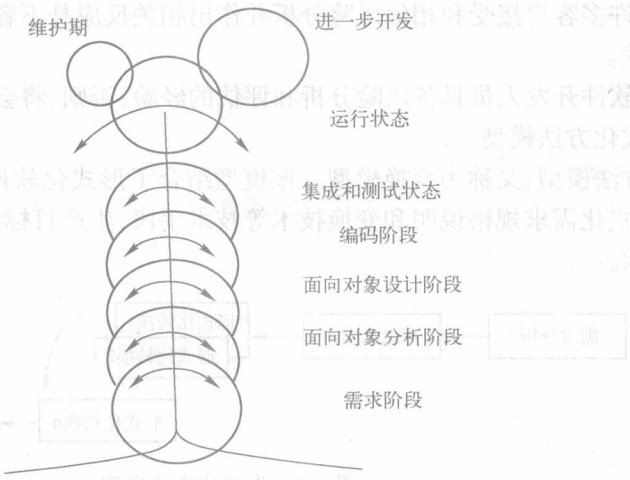


图 1-4　喷泉模型

（5）螺旋模型

螺旋模型将瀑布模型和快速原型模型结合起来，将软件过程划分为若干个开发回线，每一个回线表示开发过程的一个阶段。例如，最中心的第一个回线可能与系统可行性有关，第二个回线可能与需求定义有关，第三个回线可能与软件设计有关，……如此反复，形成了螺旋上升的过程。

螺旋模型适合于大型软件的开发，它吸收了软件工程“演化”的概念。螺旋模型如图 1-5 所示。

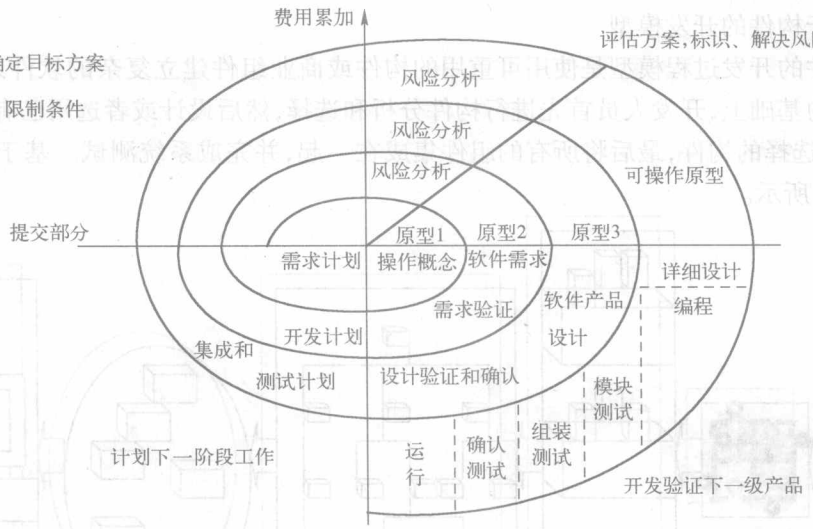


图 1-5　螺旋模型

螺旋模型的优点如下：

1）以风险驱动开发过程，强调可选方案和约束条件，从而支持软件的重用。

2）关注早期错误的消除，将软件质量作为特殊目标融入产品开发之中。

螺旋模型的缺点如下：

1）要求许多客户接受和相信风险分析并作出相关反应是不容易的，往往适应于内部的大规模软件开发。

2）需要软件开发人员具备风险分析和评估的经验；否则，将会带来更大的风险。

（6）形式化方法模型

形式化方法模型，又称为变换模型。该模型结合了形式化软件开发方法和程序自动生成技术，采用形式化需求规格说明和变换技术等技术手段，生产目标程序系统。形式化方法模型如图 1-6 所示。

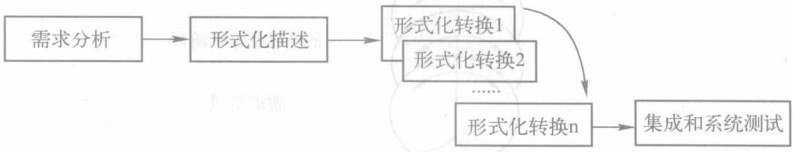


图 1-6　形式化方法模型

形式化方法模型的优点是：由于数学方法具有严密性和准确性，形式化方法开发过程所交付的软件系统具有较少的缺陷和较高的安全性。

形式化方法模型的缺点如下：

1）开发人员需要具备一定技能并经过特殊训练后才能掌握形式化开发方法。

2）现实应用的系统大多数是交互性强的软件，但是这些系统难以用形式化方法进行描述。

3）形式化描述和转换是一项费时费力的工作，采用这种方法开发系统在成本和质量等方面并不占有优势。

（7）基于构件的开发模型

基于构件的开发过程模型是使用可重用的构件或商业组件建立复杂的软件系统，即在确定需求描述的基础上，开发人员首先进行构件分析和选择，然后设计或者选用已有的体系结构框架，复用所选择的构件，最后将所有的组件集成在一起，并完成系统测试。基于构件的开发模型如图 1-7 所示。

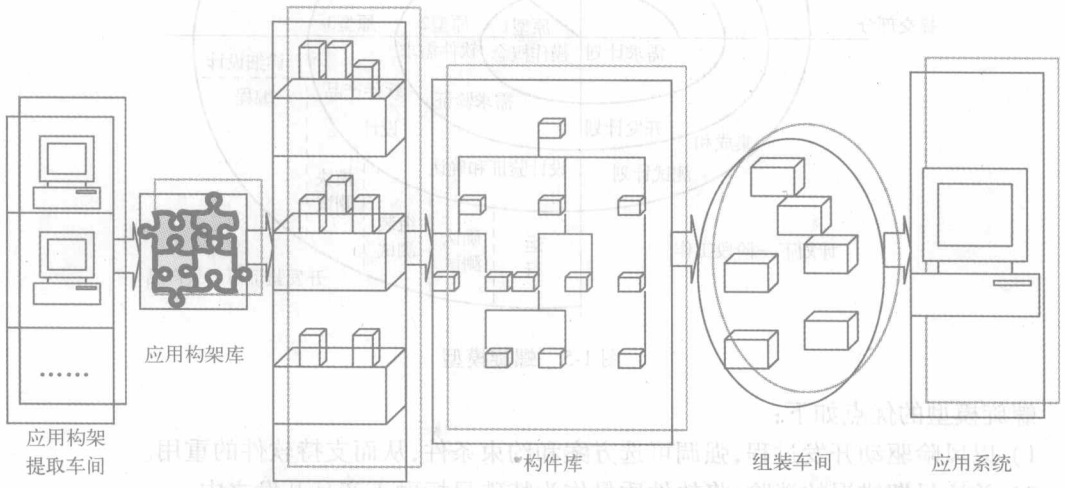


图 1-7　基于构件的开发模型