可下载教学资料

http://www.tup.tsinghua.edu.cn

高等学校教材

软件工程

软件案例分析

刘天时 等 编著　卫红春 主审

清华大学出版社

内容简介

本书围绕软件开发的一些案例由浅入深地讲述了软件开发过程中的一些设计方法(包括算法设计方法)和实例技巧;按照软件开发流程介绍了一个信息系统的开发过程,通过理论与应用相结合的方式,帮助和引导读者进一步掌握软件工程的基本概念、理论、方法和技术。结合具体案例分析讲解是本书的特点。

本书可作为高等院校本科计算机相关专业高年级和研究生教材,也可作为从事软件开发、管理、维护和应用的工程技术和管理人员的参考书。

图书在版编目(CIP)数据

软件案例分析/刘天时等编著. —北京:清华大学出版社,2008.5
(高等学校教材·软件工程)
ISBN 978-7-302-17248-2

Ⅰ. 软… Ⅱ. 刘… Ⅲ. 软件开发—案例—分析—高等学校—教材 Ⅳ. TP311.52

中国版本图书馆 CIP 数据核字(2008)第 040503 号

责任编辑:郑寅堃
责任校对:焦丽丽
责任印制:王秀菊
出版发行:清华大学出版社 **地 址**:北京清华大学学研大厦 A 座
http://www.tup.com.cn **邮 编**:100084
社 总 机:010-62770175 **邮 购**:010-62786544
投稿与读者服务:010-62776969,c-service@tup.tsinghua.edu.cn
质 量 反 馈:010-62772015,zhiliang@tup.tsinghua.edu.cn
印 刷 者:北京市清华园胶印厂
装 订 者:北京国马印刷厂
经 销:全国新华书店
开 本:185×260 **印 张**:14.75 **字 数**:359 千字
版 次:2008 年 5 月第 1 版 **印 次**:2008 年 5 月第1次印刷
印 数:1~4000
定 价:23.00 元

出版说明

改革开放以来，特别是党的十五大以来，我国教育事业取得了举世瞩目的辉煌成就，高等教育实现了历史性的跨越，已由精英教育阶段进入国际公认的大众化教育阶段。在质量不断提高的基础上，高等教育规模取得如此快速的发展，创造了世界教育发展史上的奇迹。当前，教育工作既面临着千载难逢的良好机遇，同时也面临着前所未有的严峻挑战。社会不断增长的高等教育需求同教育供给特别是优质教育供给不足的矛盾，是现阶段教育发展面临的基本矛盾。

教育部一直十分重视高等教育质量工作。2001 年 8 月，教育部下发了《关于加强高等学校本科教学工作，提高教学质量的若干意见》，提出了十二条加强本科教学工作提高教学质量的措施和意见。2003 年 6 月和 2004 年 2 月，教育部分别下发了《关于启动高等学校教学质量与教学改革工程精品课程建设工作的通知》和《教育部实施精品课程建设提高高校教学质量和人才培养质量》文件，指出“高等学校教学质量和教学改革工程”是教育部正在制定的《2003—2007 年教育振兴行动计划》的重要组成部分，精品课程建设是“质量工程”的重要内容之一。教育部计划用五年时间（2003—2007 年）建设 1500 门国家级精品课程，利用现代化的教育信息技术手段将精品课程的相关内容上网并免费开放，以实现优质教学资源共享，提高高等学校教学质量和人才培养质量。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作，提高教学质量的若干意见》精神，紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”，在有关专家、教授的倡议和有关部门的大力支持下，我们组织并成立了“清华大学出版社教材编审委员会”（以下简称“编委会”），旨在配合教育部制定精品课程教材的出版规划，讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师，其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求，“编委会”一致认为，精品课程的建设工作从开始就要坚持高标准、严要求，处于一个比较高的起点上；精品课程教材应该能够反映各高校教学改革与课程建设的需要，要有特色风格、有创新性（新体系、新内容、新手段、新思路，教材的内容体系有较高的科学创新、技术创新和理念创新的含量）、先进性（对原有的学科体系有实质性的改革和发展、顺应并符合新世纪教学发展的规律、代表并引领课程发展的趋势和方向）、示范性（教材所体现的课程体系具有较广泛的辐射性和示范性）和一定的前瞻

性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。首批推出的特色精品教材包括:

(1) 高等学校教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 高等学校教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 高等学校教材·电子信息——高等学校电子信息相关专业的教材。

(4) 高等学校教材·软件工程——高等学校软件工程相关专业的教材。

(5) 高等学校教材·信息管理与信息系统。

(6) 高等学校教材·财经管理与计算机应用。

清华大学出版社经过20年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业作出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

E-mail: dingl@tup.tsinghua.edu.cn

前言

本书以面向对象开发环境为基础，围绕软件开发的典型案例讲述了软件开发过程中的一些设计方法和实用技巧。结合案例来进行讲解，不但有助于读者理解相关的知识，而且在实际的应用中能起到参考作用。本书可作为计算机相关专业本科高年级和研究生教材，也可以作为从事软件开发和管理人员的参考书。

本书分为8个章节，第1章主要介绍软件工程的发展历程和研究现状以及数据库的发展过程；第2章为PowerBuilder入门，主要介绍PowerBuilder的一些基础知识；第3章是一些应用实例技巧，包括一对多表单设计、数据加锁方法、回滚与提示、游标模板、通知发布和常用外部函数；第4章讲述的是通用功能中的界面设计部分，包括界面风格设计、快捷键设置、进度指示器和打印机设置；第5章讲述通用功能中的数据操作部分，包括数据整理、跨库查询、数据导出与导入、大文本数据的管理、角色与授权和系统启动；第6章为算法设计，主要讲述汉诺塔游戏算法、数字拼图游戏算法、P2P网络通信算法和通用试题库组卷算法；第7章主要讲解树形可视图形界面，包括树形视图的数据库设计、数据检索和数据操作；第8章以医院管理信息系统为例，介绍软件开发的过程和方法；附录为一些实验项目，可作为实验教学内容，通过实验培养学生的实践、分析和解决问题的能力。

本书由刘天时承担主要的编写工作以及负责最后的校正和统稿工作，卫红春负责审阅本书全稿，给本书编写提出了许多中肯的意见。另外，其他参与本书编写的人员有：卫红春参与第1章和第8章，范莉莉参与第1章、第2章和第7章，马刚参与第3章和第8章，李皎参与第1章和第6章。许礼豪为本书的改错做了一定工作，李皎同学为本书后期的整理做了大量的工作。在此对本书的编写做过工作的所有老师和同学表示衷心的感谢。

由于作者水平所限，书中难免有疏漏和欠妥之处，恳请读者批评指正。本书为教师免费提供本书的实验代码，需要的老师可与本书编辑联系索取。本书编辑的电子邮箱地址是：zhengyk@tup.tsinghua.edu.cn，作者的电子邮箱地址是：liutianshi@xsyu.edu.cn。

编　者

2007年10月于西安

目录

第1章

绪　论

随着计算机的广泛应用，软件开发人员开始注重程序设计的结构、风格和可维护性，典型代表是结构化程序设计方法。20世纪60年代末，提出了软件工程的概念，其目的是倡导以工程的概念、原理和方法进行软件开发。进入20世纪80年代后开始围绕软件工程过程，展开了有关软件生产技术的研究，主要是软件重用技术和软件工程管理。近几年来，随着计算机网络的广泛应用，以软件重用技术为基础，在软件构件技术、中间件技术、分布式计算技术等方面，均取得了有影响的成果，有力地推动了软件工程学科的发展。

教学要求

(1) 掌握软件的定义、软件的特点和软件的分类；

(2) 了解软件危机的概念及导致软件危机的原因；

(3) 了解软件工程的发展历程和研究现状；

(4) 了解数据库的产生原因；

(5) 掌握数据库的特点和相关概念。

重点和难点

(1) 软件的定义、特点和分类；

(2) 数据库的特点和相关概念。

1.1　软　　件

“人们依赖于软件，但有时也毁于软件。有些软件故障令人烦恼，而有些软件故障却是灾难性的。技术带来某种风险早已不是新闻。在系统中增加软件可以使系统提供的服务更便利、更易用、更易修改，但却不会使系统更可靠。”Weiner的这段话道出了软件对系统的重要性。

1.1.1　软件的定义

软件是能够完成预定功能和性能的可执行的计算机程序和使程序正常执行所需要的数据，加上描述程序的操作和使用的文档。即“软件＝程序＋数据＋文档”。

程序是为了解决某个特定问题而用程序设计语言描述的适合计算机处理的语句序列。

数据是用来描述软件所要处理的业务和事物的静态特征，是程序处理的对象，是能被计算机存储和处理的反映客观实体信息的物理符号。

文档是软件开发活动的记录，主要供人们阅读，既可用于专业人员和用户之间的通信和交流，也可以用于软件开发过程的管理和运行阶段的维护。

1.1.2 软件的特点

软件具有以下特点：

(1) 智能性。软件是人类智力劳动的产物。

(2) 抽象性。软件是一种逻辑实体，而不是具体的物理实体。

(3) 系统性。软件是由多种要素组成的有机整体，具有确定的目标、功能和结构。

(4) 复制性。软件在开发过程中没有明显的制造过程，不像硬件可重复制造，但可无限制复制同一内容的副本。

(5) 非损性。在软件的运行和使用期间，不像硬件那样存在机械磨损、老化等问题。

(6) 依附性。软件的开发和运行常常受到计算机系统的限制，不能完全摆脱硬件而独立运行。

(7) 泛域性。软件可以服务于人类活动所涉足的各行各业。

(8) 演化性。软件在其生命周期中，其功能和性能会受到各种社会因素的影响而不断变化。

1.1.3 软件的分类

1. 按软件的功能划分

按照功能可把软件分为系统软件、支撑软件和应用软件三种类型。

系统软件：是能与计算机硬件紧密配合，使计算机系统各个部件、相关的软件和数据协调、高效工作的软件。如操作系统、数据库管理系统、设备驱动程序、通信处理程序等。

支撑软件：是协助用户开发软件的工具性软件。如文本编辑程序、文件格式化程序、磁盘向磁带传输数据的程序、程序库系统以及支持需求分析、设计、实现、测试和支持管理的软件等。

应用软件：是在特定领域内开发，为特定目的服务的一类软件。如商业数据处理软件、工程与科学计算软件、计算机辅助设计/制造软件、系统仿真软件、智能产品嵌入软件、医疗与制药软件、事务管理与办公自动化软件、计算机辅助教学软件等。

2. 按软件规模划分

微型：一个人，在几天之内完成的软件。

小型：一个人半年之内完成的代码在 2 000 行以内的软件。

中型：5 个人以内，在一年多时间里完成的代码在 5 000～50 000 行的软件。

大型：5～10 个人，在两年多的时间里完成的代码在 5 万～10 万行的软件。

甚大型：100～1 000 个人参加，用 4～5 年时间完成的具有 100 万行代码的软件。

极大型：2 000～5 000 个人参加，10 年内完成的代码在 1 000 万行以内的软件。

3. 按软件工作方式划分

实时处理软件：是指在事件或数据产生时，立即予以处理，并及时反馈信息，控制需要监测和控制其过程的软件。

分时软件：是管理多个联机用户同时使用一台主机的软件。

交互式软件：是指能实现人机通信的软件。

批处理软件：是把一组输入作业或一批数据以成批处理的方式一次运行，按顺序逐个处理完的软件。

4. 按软件服务对象的范围划分

项目软件(也称为定制软件)：是受某个特定客户(或少数客户)的委托，由一个或多个软件开发机构在合同的约束下开发出来的软件。

产品软件：是由软件开发机构开发出来后直接提供给市场，或是为千百个用户服务的软件。

5. 按软件使用的频度划分

一次使用软件。

多次使用软件。

6. 按软件可靠性划分

高可靠性软件。

一般可靠性软件。

1.1.4 软件危机

1. 软件危机的含义

软件危机是指在 20 世纪 60 年代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，这些问题给软件的生产和应用造成严重的社会障碍。软件危机表现在以下几方面：

(1) 软件开发人员与用户完全沟通通常比较困难，用户对已完成系统不满意的现象经常发生。

(2) 软件应用的需求快速增长，软件开发生产率的提高赶不上硬件的发展和人们需求的增长。

(3) 软件测试技术规范和制度不够健全，软件产品的质量往往不可靠。

(4) 对软件开发成本和进度的估计常常不准确。开发成本超出预算，相比预定计划实际进度一再拖延的现象并不罕见。

(5) 软件开发常常没有依据统一的、科学的开发规范，软件的可维护程度低。

(6) 软件通常没有适当的文档资料。

(7) 软件的成本逐年上升,软件的价格昂贵。

2. 软件危机产生的原因

软件危机产生的原因主要有如下 5 个方面:

(1) 管理和控制软件开发过程相当困难,这与软件本身的特点有关。软件不同于硬件,它是计算机系统的逻辑部件而不是物理部件。在写出程序代码并在计算机运行之前,软件开发过程的进展情况较难衡量,软件开发的质量也较难评价。

(2) 软件维护通常意味着改正或修改原来的设计,客观上使得软件较难维护。

(3) 在软件开发过程中,或多或少地采用了错误的方法和技术。

(4) 对用户需求还没有完整准确地把握,就匆忙着手编写程序。

(5) 开发人员与管理人员重视开发而轻视问题的定义和软件维护。

3. 软件危机的解决途径

采用以下软件工程学的技术和方法,是解决软件危机的有效途径:

(1) 采用工程化方法和工程途径来研制、维护软件。

(2) 采用先进的技术、方法和工具来设计和实现软件。

(3) 采用必要的组织管理措施。

1.2 软件工程的发展历程

自 1946 年第一台计算机问世以来,计算机硬件技术迅速发展,其处理能力不断提高,体积、功耗和成本却不断下降,而计算机软件的发展却远远落后于计算机硬件的进步。当时的软件生产具有个体化、作坊式的特点,开发工具落后,开发平台单一,程序设计语言功能差,软件开发维护费用远远超出预算,软件生产成本急剧增长,开发周期长,进度不易控制,软件可靠性差。这种软件开发技术、开发语言和开发环境与软件需求的日益增加形成了尖锐的矛盾,从而产生了软件危机。软件危机的出现使人们认识到软件开发需要探索新的方法、技术和工具。20 世纪 60 年代末,E. W. Dijkstra 提出了结构化程序设计的概念,他强调从程序结构上来研究程序设计,指出分析设计比编码更为重要。北大西洋公约组织(North Atlantic Treaty Organization,NATO)于 1968 年召开学术研讨会,会议第一次提出了"软件工程"的概念,采用工程的概念、原理和方法来开发、管理和维护软件。自此以后,软件工程作为一门新兴的学科正式诞生,人们开始了软件工程的研究。

1.2.1 软件工程的发展阶段

1. 软件工程准备期

从 1968 年软件工程概念提出到 1975 年期间是软件工程学科的准备时期。这个阶段提出了软件工程概念,探讨了软件开发过程中存在的诸多问题,并试图通过使用局部方法和工

具以及改善软件管理手段来解决这些问题。该阶段的主要工作有：

(1) 对软件开发以及程序设计中存在的问题进行了深入分析，对问题进行归类，并思考问题出现的原因。

(2) 对程序设计方法、数据的抽象表示以及程序实现技巧进行了深入研究，并提出了诸如自顶向下设计、结构化设计、可靠性软件设计等有效的程序设计方法。

(3) 开始重视软件测试技术和容错设计。

(4) 提出改进软件质量的方法。

(5) 提出软件生产化的必要性和设想。

2. 软件工程形成期

1975 年到 1980 年是软件工程学科体系的形成时期。这个阶段的典型特征是：

(1) 软件工程作为一个学科体系初步形成。明确了软件工程学的涵义，提出了软件生存周期的概念，规定了软件生存期各阶段的划分，并对软件维护给予充分重视。

(2) 出现了软件工程方法学、程序设计方法学、软件度量学及成本核算技术、软件开发工具与环境、软件工程规范与标准等软件工程学科的多个分支领域。

(3) 数据库技术、微处理机技术、通信网络技术、人机界面技术、多语言处理技术等相关技术的出现和发展。

3. 软件工程发展期

1981 年至今，软件工程学科进入持续发展时期。这个时期软件开发相关技术得到迅速发展，软件应用向深度和广度发展，软件的规模和范围不断扩大，并由独立系统向开放、互连和集成的一体化系统发展，软件产业化速度加快，软件体系结构和软件建模技术开始受到重视，软件开发环境逐步成熟。尤其进入 20 世纪 90 年代中期以来，软件工程环境得到迅速发展，可视化技术、以 C/S 和 B/S 为代表的软件体系结构模式、面向对象和面向服务的方法、以统一建模语言(Unified Modeling Language，UML)为代表的软件建模语言和建模技术以及以统一软件过程(Rational Unified Process，RUP)为代表的软件工程过程的规范和统一等把软件工程引入一个新的发展时期。

1.2.2 软件工程的发展过程

1968 年北大西洋公约组织会议第一次在国际上正式提出软件工程概念之后，软件工程化一直在不断发展，具有代表性事件或产品出现的年代如图 1.1 所示。

软件工程发展过程中，具有代表性事件或产品的含义如下：

- WFL。工作流语言(Work Flow Language)；
- 3GL。第三代编程语言(Third Generation Language)；
- PC。微机(Personal Computer)；
- TQM。全面质量管理(Total Quality Management)；
- 4GL。第四代编程语言(Fourth Generation Language)；
- COTS。商品化的产品和技术(Commercial Off-The-Shelf)；

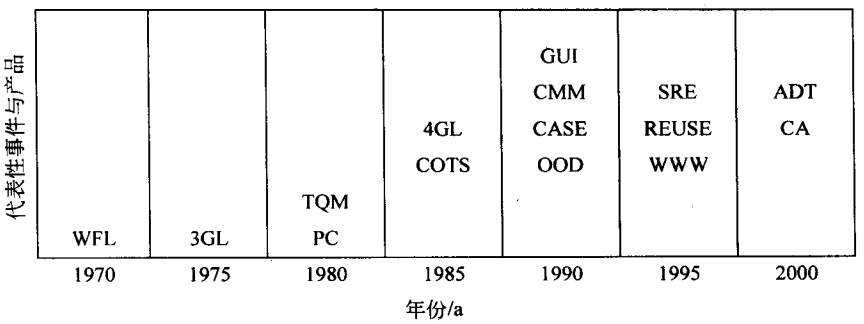

图 1.1 软件工程发展过程

- CMM。软件能力成熟度模型(Capacity Maturity Model)；
- CASE。计算机辅助软件工程(Computer Aided Software Engineering)；
- OOD。面向对象设计(Object Oriented Design)；
- SRE。软件可靠性工程(Software Reliability Engineering)；
- GUI。图形用户界面(Graphic User Interface)；
- REUSE。复用；
- WWW。万维网(World Wide Web)；
- ADT。抽象数据类型(Abstract Data Type)；
- CA。计算机辅助(Computer Aided)。

软件工程初步形成于20世纪70年代。1970年，W. Royce提出了“瀑布模型”。由于瀑布模型描述了软件生命周期过程的基本活动，因此也称为传统的生命周期模型，如图1.2所示。在瀑布模型中，系统分析员先收集系统需求，并由用户方和软件开发方共同确认；然后编写需求规格说明文档且需要双方共同确认；随后进入软件项目计划阶段，用户方确认后进入设计阶段；设计人员认可之后，由程序员进行编码实现；编码完成由用户方进行验收测试。经过一段时间后，可能用户方最终会将该软件废弃，如使用新系统代替等。瀑布模型的优点包括强制性的分阶段方法。该方法要求每一阶段任务完成后，都必须对其阶段性的产品(包括文档)进行评审，方可进入下一阶段的工作，强调软件开发过程的阶段性。瀑布模型是一种理想的线性模型，开发过程缺乏灵活性，软件出现的错误只能在后期发现，要修改前期的错误将要付出非常大的代价，因此软件维护较难。

从20世纪50年代中期到70年代，出现了以FORTRAN、COBOL、BASIC、C等为代表的第三代程序设计语言。为了满足软件工程化的迫切要求，1976年美国电气和电子工程师协会(Institute of Electrical and Electronics Engineers，IEEE)标准化部成立了一个软件工程组，负责起草软件工程标准。70年代后期还出现了规范化的软件开发学。

20世纪80年代是软件产业化和软件工程规范化的开端。由于软件市场需求迅猛增长和计算机硬件、外部硬件设备及有关接口的迅速标准化，软件逐步脱离对硬件的依赖，成为世界上发展最快的独立产业。与此同时，软件公司大量涌现，其中最知名的是微软公司的成立与崛起。软件开发也逐步走入专业化、规范化的轨道。IEEE于1980年出版了第一个软件工程标准IEEE STD730(软件质量保证标准)，该标准成为IEEE整个软件工程早期标准的基础。之后，美国宇航局(National Aeronautics and Space Administration，NASA)、欧洲

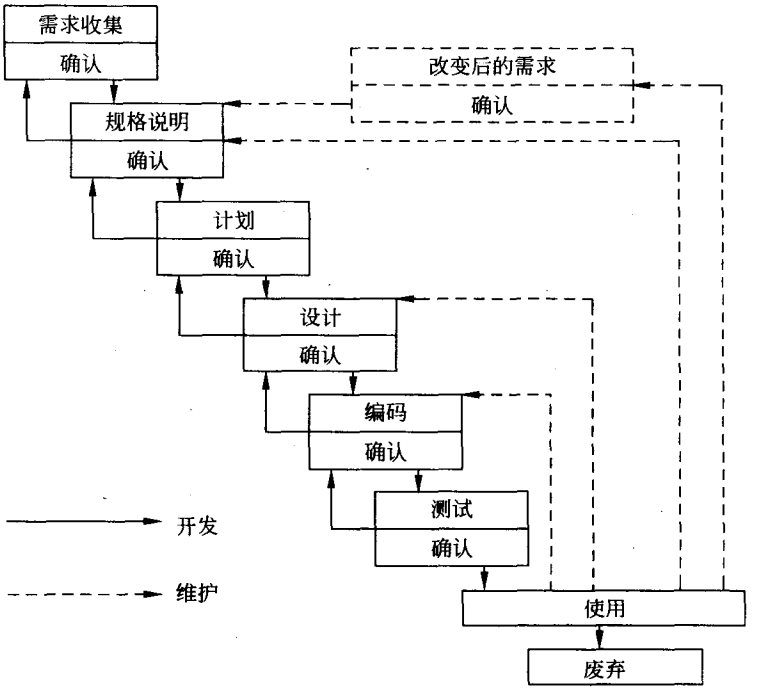

图 1.2 瀑布模型

空间局(European Space Agency,ESA)、美国国防部(US Department of Defense,DOD)等机构也陆续发布了一些技术指令和标准,并通过培训来推进软件工程的规范化和标准化。在这些软件质量保证标准中,系统管理方面广泛应用的"全面质量管理"的理论也被引入到软件工程之中。20 世纪 80 年代中期出现了面向对象的、非过程化的第四代程序设计语言。

20 世纪 80 年代后期,由于瀑布模型存在的缺陷,人们又提出了改进模型,即原型模型,如图 1.3 所示。原型模型的首要任务是根据用户提出的软件基本需求快速建立一个系统原型,然后重复让用户通过使用原型对其提出意见,软件开发人员根据意见快速修改原型,直至用户对开发的系统原型满意为止。这一模型的最大优点是采用逐步求精、反复修改完善的方法使原型逐步完善,开发出真正满足用户需要的软件。但在实际操作中也存在几个问题:(1)要求软件开发人员迅速生成这些原型,如果没有理想有效的工具软件的支持,其开发速度和进度是非常慢的,特别对于大型复杂系统的开发有可能因原型的失败而导致失败。(2)由于对原型模型繁杂的修改活动,而忽视开发过程中的其他因素。如果对原型的修改不能收敛于期望值,很可能导致失败。(3)原型化方法实际上是简化了的软件生存周期过程,其实现的策略偏于非形式化,对原型项目的控制管理比较困难。之后出现的喷泉模型、螺旋模型等大都是对瀑布模型和原型模型的一种扩展和改进。

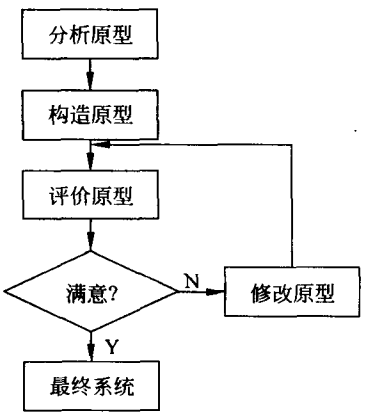

图 1.3 原型模型