



“十三五”应用技术型人才培养规划教材

UML软件建模项目化教程

UML RUANJIAN JIANMO XIANGMUHUA JIAOCHENG

刘中胜 主 编

叶丽珠 龚芳海 余学文 副主编

刘中胜

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

“十三五”应用技术型人才培养规划教材

UML 软件建模 项目化教程

刘申胜 主 编

叶丽珠 龚芳海 余学文 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

UML 软件建模是软件系统分析和设计过程的重要内容,本书分析了 UML 建模技术岗位的实际需求,应用主流的建模工具软件,以经典的项目案例为载体,详细介绍了 UML 建模基础知识和实践技能。本书的 UML 软件建模工具软件采用的是 Enterprise Architect (V12) 中文版。按照“项目导向、任务驱动”的教学方法,以“eshop 购物商城系统”和“高校图书管理系统”两个项目案例贯穿整个教程。

本书通俗易懂,强化学习者能力的培养,将知识理解能力、技能实践能力有机结合,具有较高的实用价值。本书适合作为应用型本科、高职高专院校软件工程及 UML 建模的教材,也可作为广大软件设计人员学习或参考书籍。本书免费提供教学课件及相关教学资源。

图书在版编目(CIP)数据

UML 软件建模项目化教程 / 刘中胜主编. —北京:
中国铁道出版社, 2018. 7

“十三五”应用技术型人才培养规划教材

ISBN 978-7-113-24567-2

I. ①U… II. ①刘… III. ①面向对象语言-程序
设计-高等学校-教材 IV. ①TP312. 8

中国版本图书馆 CIP 数据核字(2018)第 118757 号

书 名: UML 软件建模项目化教程

作 者: 刘中胜 主编

策 划: 韩从付

读者热线: (010) 63550836

责任编辑: 陆慧萍 李学敏

封面设计: 刘 颖

责任校对: 张玉华

责任印制: 郭向伟

出版发行: 中国铁道出版社(100054,北京市西城区右安门西街8号)

网 址: <http://www.tdpress.com/51eds/>

印 刷: 北京铭成印刷有限公司

版 次: 2018年7月第1版 2018年7月第1次印刷

开 本: 787 mm×1 092 mm 1/16 印张: 19.25 字数: 465 千

书 号: ISBN 978-7-113-24567-2

定 价: 52.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 51873659

前 言

UML (Unified Modeling Language, 统一建模语言) 是软件界第一个统一的可视化建模语言, 已经成为国际软件界广泛承认的标准, 其应用领域非常广泛。通过 UML 创建软件系统静态结构和动态行为等多种结构模型是软件系统分析和设计的重要内容, 这也在很大程度上提高了软件开发的效率和质量。

本书是编者在总结了多年的软件项目实践经验和多年的教学经验的基础上编写的, 既考虑了 UML 软件建模在实践项目中知识和技能的需求, 又结合了应用型本科和高职高专的教学实际需求。因此, 通过本书的学习, 读者可以快速、全面地掌握适合真实软件项目的 UML 建模知识和技能。

本书应用主流的 UML 建模工具软件 Enterprise Architect (V12) 中文版, 以“eshop 购物商城系统”和“高校图书管理系统”两个案例贯穿整个教程。从真实软件项目的需求分析和设计过程出发, 基于“项目导向、任务驱动”的教学思想, 围绕这几个方面的专业知识和实践技能做了详细的介绍: 软件工程基础知识, 面向对象方法与软件建模, UML 基础知识, 用例建模, 静态建模, 数据库建模, 动态建模, 物理建模, 双向工程等。

本书具有如下创新之处:

(1) 认真进行软件需求分析以及设计岗位需求和就业现状分析, 全面规划、合理安排教材内容。打破传统的“以教师为中心、以某位任课教师的知识结构为中心”的教学观念, 而以“企业需求为核心、以市场需求为导向”。

(2) 仔细分析应用型本科和高职高专学生的学习特点, 以“学得懂、用得上”为目标, 筛选并重构 UML 建模技术中符合应用型本科和高职高专学生学习的内容, 把复杂的、难学懂的, 且企业岗位中极少涉及的相关内容剔除, 保留学生能学懂且在企业岗位中很实用的知识和技能等关键内容。

(3) 基于“项目导向、任务驱动”的教学方法组织教材内容, 整个教材分为十个项目, 每个项目又根据项目实际需求分解为多个任务来实现。

(4) 符合项目化教学思路。每个项目根据“项目指导→项目分析→各任务的描述与实现→项目回顾→拓展训练”的项目化教学思路来完成每个项目的内容。

(5) 理论知识和实践技能相结合。每个项目、每个任务对涉及的技术理论做恰到好处的介绍, 以“能理解、必须掌握、够用”为原则组织理论内容。实践操作技能是应用型本科和高职高专学生学习的關鍵, 因此, 各项目从提高知识和技能两个目标能力出发, 详细介绍相关内容。

(6) 充分考虑教学实施需求。每个项目在项目指导中包含项目分解、知识目标、技能目标、项目导航。同时, 在项目综述中提供了教学实施计划和进度安排, 以供教学参考。

(7) 充分考虑了不同层次和不同兴趣的学生。每个项目设计有拓展训练, 包括理论知识训练和实践技能训练。

(8) 提供配套的教学资源。本书提供配套的教学课件和拓展训练参考答案, 网络下载地址为:

www.tdpress.com/51eds/。

本书由刘中胜任主编，叶丽珠、龚芳海、余学文任副主编，由刘中胜负责统稿、定稿。其中，项目一、项目二、项目四、项目五和项目七由刘中胜编写，项目三、项目六、项目八和项目九由叶丽珠编写，项目十由龚芳海、余学文编写。

本书在编写过程中得到了单位领导、同事和学生们的热情帮助和支持，在此向他们表示感谢。由于时间仓促和水平有限，书中难免存在疏漏和不足之处，敬请广大读者和专家批评指正。

编者
2018年3月

项目案例说明

本教程以基于项目任务的项目化教学方式来讲解 UML 软件建模的基本知识和操作技能。贯穿本教程的项目案例有两个：教学案例以“eshop 购物商城”为主，拓展训练案例以“高校图书管理系统”为主。对于拓展训练项目，各位教师在教授过程中可要求学生参照所在学校的高校图书管理系统进行各项目的拓展训练，充分发挥学生自主学习的能力。

本教程软件建模软件使用目前软件企业所采用的主流建模软件，其中，正文教学内容的 UML 建模采用 Enterprise Architect（简称 EA，本书采用 EA V12.0 版本）建模软件进行介绍。拓展训练建议采用 Visio 或 Rational Software Architect 建模软件来完成，学生可以参照 EA 软件的操作思路自学 Visio 或 Rational Software Architect 等建模软件的实践操作技能，对于比较复杂的操作部分，教师可适当进行指导。

【教学案例——eshop 购物商城介绍】

网络购物已经成为人们的重要生活方式，购物商城的电子商务系统是信息系统重要的、非常普及的应用领域。因此，购物商城是软件企业遇到的很常见的开发项目，具有普遍性和代表性。eshop 购物商城系统要求能够实现前台购物的功能和后台管理的功能。其前台购物子系统的主要功能包括商品查询、商品展台、购物车、会员设置、支付、订单查询和商城公告等，其功能结构图如图 1 所示。

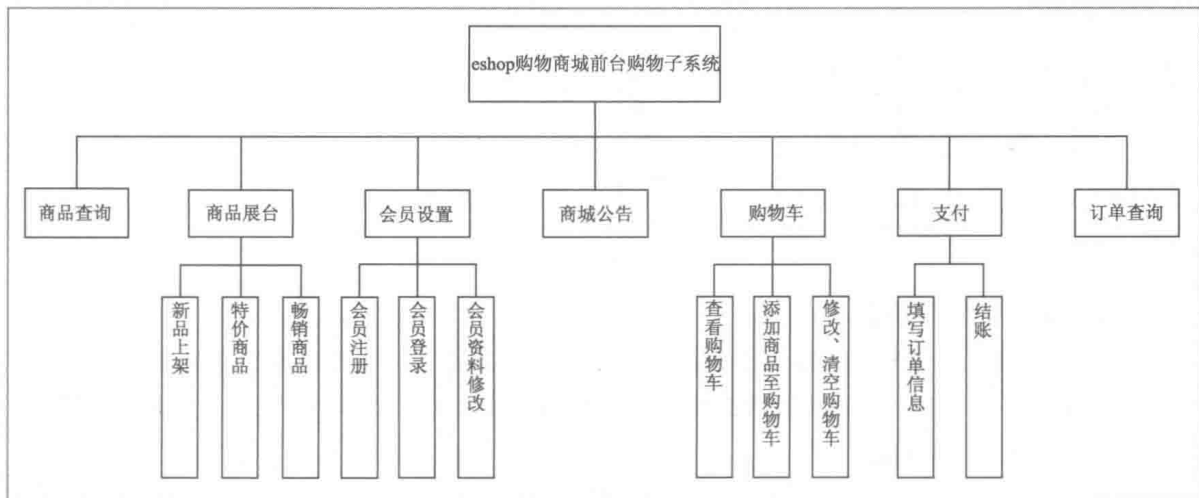


图 1 eshop 购物商城前台购物子系统功能结构图

后台管理子系统的主要功能包括商品类别管理、商品管理、会员管理、订单管理、员工账户管理、公告管理等，其功能结构图如图 2 所示。

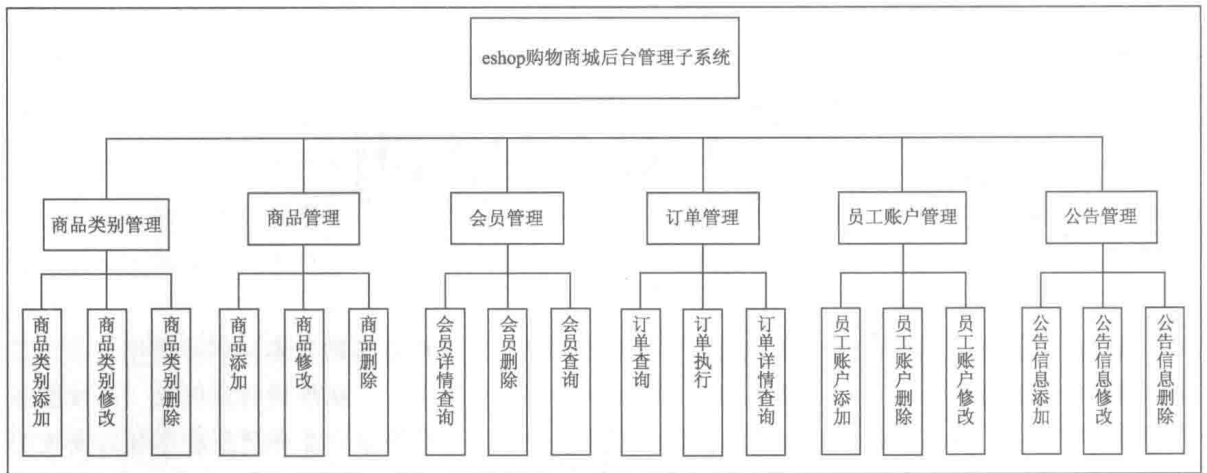


图2 eshop 购物商城后台管理子系统功能结构图

【拓展训练案例——高校图书管理系统介绍】

高校图书管理系统是学生接触较多，较为熟悉的信息管理系统，学生对图书管理系统有较深的使用体验，有利于学生自主地进行软件建模的拓展训练。学生以所在学校的图书管理系统实现的功能为基础，进行 UML 软件建模的拓展训练。也可以参照图 3 所示的图书管理系统实现的主要功能为基础进行 UML 软件建模的拓展训练。

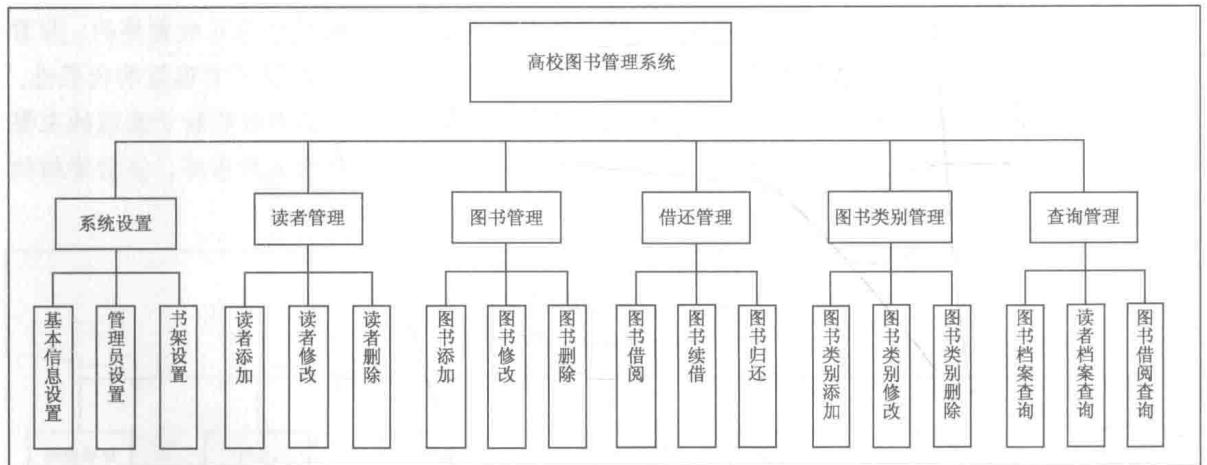


图3 高校图书管理系统功能结构图

【项目实施软件环境】

项目实施软件环境主要包括操作系统平台和建模工具软件。在进行数据库建模时采用 MS SQL Server 数据库平台，在进行双向工程时采用 Java 开发环境。具体软件环境如下：

操作系统平台：Windows 7。

建模工具软件：Enterprise Architect、PowerDesigner、Visio、Rational Software Architect。

Java 开发平台：MyEclipse、JDK1.6。

数据库平台：MS SQL Server 2008。

【项目实施计划】

UML 软件建模课程通常安排 16 周或 18 周，每周 4 课时，总计为 64 课时或 72 课时。根据

这样的课时安排，本项目教程的教学实施的主要内容包括软件工程概述、面向对象技术、UML 基础知识、UML 建模技术，以及综合项目实训等。下面给出 64 课时的实施计划建议，根据学生情况和学校教学安排等实际情况，各位教师可以做出适当调整。

项目	项目标题	建议课时	项目	项目标题	建议课时
项目一	认识软件工程	4	项目六	数据库建模	4
项目二	面向对象方法与软件建模	6	项目七	动态建模	14
项目三	认识 UML	6	项目八	物理建模	4
项目四	用例建模	6	项目九	双向工程	4
项目五	静态建模	8	项目十	综合项目实训	8

项目一 认识软件工程	1
【任务 1-1】 认识软件与软件工程	2
【任务 1-2】 认识软件生命周期	7
【任务 1-3】 认识软件过程模型	9
【项目回顾】	15
【拓展训练】	16
项目二 面向对象方法与软件建模	17
【任务 2-1】 认识面向对象方法	18
【任务 2-2】 了解面向对象基本概念与特征	21
【任务 2-3】 软件建模	28
【项目回顾】	34
【拓展训练】	35
项目三 认识 UML	37
【任务 3-1】 了解 UML 结构	38
【任务 3-2】 了解 UML 视图和图	43
【任务 3-3】 UML 建模工具	49
【项目回顾】	55
【拓展训练】	56
项目四 用例建模	57
【任务 4-1】 了解用例图的基础知识	58
【任务 4-2】 绘制用例图	66
【项目回顾】	85
【拓展训练】	86
项目五 静态建模	87
【任务 5-1】 了解类图的基础知识	88
【任务 5-2】 绘制类图	99
【任务 5-3】 了解对象图的基础知识	115
【任务 5-4】 绘制对象图	118
【项目回顾】	129
【拓展训练】	130

项目六 数据库建模	131
【任务 6-1】 配置 PowerDesigner 安装与使用环境	132
【任务 6-2】 熟悉 PowerDesigner 建模过程	146
【任务 6-3】 创建概念数据模型	153
【任务 6-4】 创建物理数据模型	158
【项目回顾】	162
【拓展训练】	162
项目七 动态建模	164
【任务 7-1】 了解状态图的基础知识	165
【任务 7-2】 绘制状态图	173
【任务 7-3】 了解活动图的基础知识	190
【任务 7-4】 绘制活动图	195
【任务 7-5】 了解顺序图的基础知识	207
【任务 7-6】 绘制顺序图	211
【任务 7-7】 了解通信图的基础知识	220
【任务 7-8】 绘制通信图	222
【项目回顾】	227
【拓展训练】	228
项目八 物理建模	229
【任务 8-1】 了解组件图的基础知识	230
【任务 8-2】 绘制组件图	233
【任务 8-3】 了解部署图的基础知识	239
【任务 8-4】 绘制部署图	241
【项目回顾】	247
【拓展训练】	248
项目九 应用双向工程	249
【任务 9-1】 应用正向工程	250
【任务 9-2】 应用逆向工程	267
【项目回顾】	274
【拓展训练】	274
项目十 综合项目实训	276
【任务 10-1】 项目简介	277
【任务 10-2】 实训演练	282
【任务 10-3】 实训拓展	291
【项目回顾】	297
【拓展训练】	297
参考文献	298

自从世界上第一台计算机出现以后，软件经过几十年的飞速发展，为各行各业的信息化实现提供了强劲的技术保障。随着社会各行业应用需求的不断增长，导致软件系统的需求越来越复杂，规模越来越庞大，开发难度越来越大，因而，开发成本也越高，质量也难以保证，项目成功率低。软件在开发和维护过程中遇到一系列严重问题，从而产生“软件危机”。为了解决“软件危机”的问题，或者尽量减少“软件危机”带来的负面影响，人们开始使用工程的方法对软件开发过程进行管理，逐步形成了“软件工程”的知识领域。

本项目将介绍软件和软件工程的基本概念、软件生命周期和软件过程模型等方面的基础知识，是进行UML软件建模前所必须掌握的基础知识。

项目指导

项目分解	任务 1-1：认识软件与软件工程 任务 1-2：认识软件生命周期 任务 1-3：认识软件过程模型
知识目标	(1) 理解软件、软件工程、软件生命周期的概念 (2) 了解软件的特点，掌握软件的分类 (3) 了解软件危机的产生及表现 (4) 掌握传统方法学和面向对象方法学各自的特点 (5) 理解瀑布模型、V模型、快速原型模型、增量模型、螺旋模型和喷泉模型等经典软件过程模型的基本思想、特点和优缺点
技能目标	(1) 能够区别软件和硬件，并描述二者的关联和各自的特点 (2) 能够列举当今市场主流的软件产品和硬件产品，并且分别对这些软硬件产品进行分类 (3) 能够以某真实软件项目为例，描述其生命周期及各个阶段的主要任务工作和成果 (4) 能够以某真实软件项目为例，分析其开发过程中所采用的过程模型
项目导航	【项目指导】——【项目分析】——【任务 1-1】——【任务 1-2】——【任务 1-3】—— 【项目回顾】——【拓展训练】

项目分析

人们对软件和软件工程的认识，无论是概念，还是实现的方法、工具、技术和管理，都在

不断地发展，对软件的理解越来越透彻，对软件的应用越来越广泛，并且不断地创新和发展。本项目将详细介绍软件的概念和分类、软件危机、软件工程的概念、软件生命周期和过程模型等各方面的内容。

本项目分为 3 个任务来实现不同的目标，具体的任务分解如下：

任务 1-1：认识软件与软件工程。了解软件的概念、软件分类、软件危机、软件工程的概念、软件工程方法学等相关内容。

任务 1-2：认识软件生命周期。了解软件生命周期的划分和各个阶段的主要工作等相关内容。

任务 1-3：认识软件过程模型。理解瀑布模型、V 模型、快速原型模型、增量模型、螺旋模型和喷泉模型等主要软件过程模型。

【任务 1-1】 认识软件与软件工程

随着计算机技术、信息技术的发展，软件的概念、软件项目开发的方法、工具、技术和管理都在不断地发展而逐步完善、成熟和创新。随着软件危机的出现，使用工程学的方法来实现和管理软件开发项目已成必然，因而形成了“软件工程”知识领域。

任务描述

软件已经渗透到了社会的各个领域。对软件及软件工程的概念、相关知识的理解与应用是软件技术或计算机技术专业的人员必须了解和掌握的。

本任务需要学习如下内容。

- 软件的概念。
- 软件分类。
- 软件危机的产生。
- 软件工程的概念。
- 软件工程方法学。

任务实现

1. 软件的概念

软件经过几十年的发展，到现在，已经在当今的信息产业中占绝对的主导地位，扮演着重要的、多重的角色。什么是软件呢？对软件定义的理解是随着计算机技术、信息技术的发展而逐步变化、完善和成熟的。人们对软件的概念理解可以分为四个阶段。

第一阶段，20 世纪 50 年代到 60 年代。人们通过编写越来越多的程序让计算机硬件运行起来，达到相应的应用目的。在当时，人们认为软件就是程序，即“软件=程序”。而程序就是一系列指令序列的集合，这些指令序列集合能被计算机硬件识别、存储、理解和执行。

第二阶段，20 世纪 60 年代到 70 年代。随着对计算机系统的需求逐渐变多且复杂，程序的规模逐渐增大，编写的周期也逐渐变长，一个程序的编写团队人员数量也比较多，编写过程中产生的中间成果也较多。因此，人们开始认识到在程序编写过程中，开发文档的作用不容忽视，开始提出将程序分析、设计、实现、测试和维护等文档归于软件中来。这时，形成了软件就是程序和相关的开发文档的概念，即“软件=程序+文档”。

第三个阶段，20 世纪 70 年代以后至 20 世纪末。软件需求变得更复杂，软件规模也变得

更加庞大，数据是程序执行的基础和对象，数据量也不断增大，程序在运行过程中需要存储、访问和处理数据的需求已经相当重要了，对数据这些方面的需求也是衡量软件效率和质量的重要依据。因此，人们又给软件的定义加上了数据，即“软件=程序+文档+数据”。1983年IEEE为软件下的定义是：计算机程序、方法、规则和相关的文档资料以及在计算机上运行所必需的数据。

第四阶段，当今21世纪。随着互联网技术、云计算和大数据技术的广泛应用，更多不同的软件部署在互联网上，各种应用可以通过互联网提供。这样软件作为一种服务，通过“按需分配”“付费使用”的模式提供应用服务。因此，软件作为一种服务，正被人们广泛接受，即“软件=程序+文档+数据+服务”。

软件与硬件是计算机系统中相互依存的两个重要组成部分，也都是产品。但软件有自身的一些特点，软件的特点比较多，主要表现在如下几个方面。

① 软件不具备明显的物理特性。软件是一种逻辑实体，在物理上并不是一种有形产品。其特性是抽象的，有别于现实世界中具体物理实体，不具备外观、重量、颜色等物理特性。

② 软件对计算机系统有一定的依赖性。软件的开发和运行通常会受到计算机系统的限制，如CPU的处理能力、计算机系统的总体结构、计算机系统的内存和外存、计算机的显示设备等各种硬件设备的限制。

③ 软件是被开发或设计出来的。软件不是一种有形的物理产品，一般有形的物理产品是被制造出来的，如通过磨具、机器等设备制造出来。软件是软件开发人员的智力劳动成果，通过手工编写程序代码开发出来的，软件产品是软件开发人员智力结果的一种载体。

④ 软件在使用过程中，没有磨损和老化问题。一般情况下，有形的物理硬件产品在使用过程中不断有磨损和老化现象。物理硬件产品在使用初期，是磨合期，物理磨损通常是比较严重的，但经过一段时间的磨合后，将进入相当稳定的时期，随着时间的流逝，硬件的磨损将再次趋于严重，这个时候，硬件的使用寿命实际上即将结束。硬件因磨损导致的失效率与时间关系如图1-1所示。

软件不是一种有形的物理产品，不存在磨损和老化问题。但也会因程序错误而失效。在软件运行初期，未知的错误使程序的失效率比较高，当修正这些错误后而未引入新错误后，软件运行稳定。但在运行一段时间后，又可能会引入新的错误后而产生失效，这时又需要修正错误使软件运行稳定，经过多次修改，形成了锯齿形的失效率曲线。软件因错误导致的失效率与时间的关系如图1-2所示。

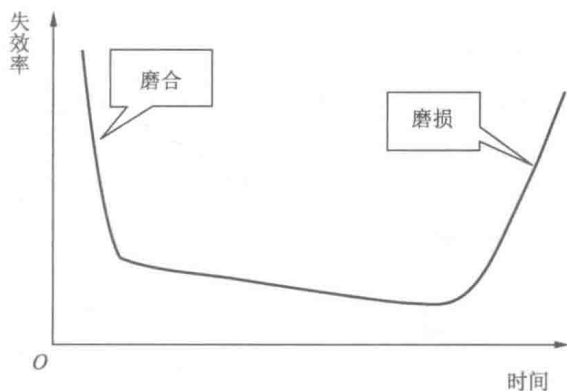


图 1-1 硬件失效率曲线

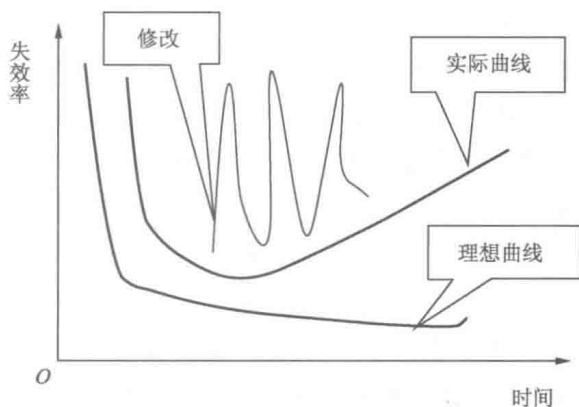


图 1-2 软件失效率曲线

⑤ 软件开发过程中，具有复杂性和不确定性。现代软件需求越来越复杂多样，并且需求变化比较频繁，规模也更大，开发过程更长且复杂，涉及技术和学科更多，无论从管理方面还是技术方面，涉及的因素相当多，遇到的风险也多，这就导致软件开发是个复杂的过程，具有复杂性、多样性和不确定性等特征。

2. 软件的分类

软件在各行各业应用非常普及，软件数量和规模都相当惊人。对软件进行分类没有一个统一的、严格的标准，但根据市场主流的常用软件产品的应用情况，可从不同的角度对软件进行适当的分类。常用软件的情况如表 1-1 所示。

表 1-1 常用软件分类情况表

序号	分类依据	类别	描述
1	按功能分类	系统软件	计算机系统不可缺少的组成部分，能使计算机各个部件、其他软件协调工作，如操作系统（Windows、Linux）、驱动程序、通信处理程序等
		支撑软件	用于开发软件的工具性软件，如各种开发软件包（jdk）和专用工具软件等
		应用软件	在某一领域中，为解决某特定目的的服务而开发的一类软件，如电子商务系统、电子政务系统、企业管理信息系统等
2	按版权分类	商业软件	版权受法律保护，经授权才能使用，并且需付费购买的软件
		共享软件	与商业软件类似，但可以“先试用再付费”，其获取途径主要是通过 Internet
		自由软件	无须支付任何费用即可得到并使用的软件，获取途径类似于共享软件
3	按服务对象分类	通用软件	由软件开发机构开发出来直接提供给市场，为大多数用户服务的软件，具有通用性，如 Office、Photoshop 等
		定制软件	受某特定用户的委托，在合同的约束下开发出来的软件，如为某个企业开发购物商城系统、为某个政府机构开发电子政务系统

3. 软件危机

软件危机出现于 20 世纪 60 年代末，在 1968 年在德国 Garmisch 软件工程大会上第一次提出“软件危机”术语。Edsger Dijkstra 在 1972 年 ACM 图灵奖演讲中解释：“软件危机的主要原因是与之前相比，机器能力已经增强了几个数量级！直截了当地说：只要没有机器能力的变化，编程是一点问题都没有；当用一些很差的计算机时，编程是一个小问题，现在有庞大的计算机，编程也成为同样大的问题。”

软件危机是指在计算机软件的开发与维护过程中所遇到的一系列严重问题。这些问题可能导致软件不能正常运行，或运行效率严重下降，运行效果无法满足实际需求。实际上，绝大多数软件都不同程度地受这些问题的影响。概括性地说，软件危机体现在两个方面：一是如何开发软件，以满足对软件日益增长的需求；二是如何维护软件，以使已经开发出来的软件能提供高质量的服务。

软件开发周期长、成本高、质量差、维护困难，这些都是软件危机的体现，软件危机与硬件和软件开发过程的复杂性是分不开的。具体地说，软件危机主要表现在如下几个方面。

① 对软件开发的进度和成本估算不准，难以预测，难以控制。在进度方面，拖延很严重，实际进度比预期计划进度拖延几个月甚至几年的现象经常出现。在成本方面，超支很严重，实际花费的成本往往比预算的成本超出很多，甚至有可能超出一个数量级别。而为了赶进度或节

约成本，采取的一些措施往往损害了软件产品的质量。

② 用户对软件产品的需求难以满足。客户对软件需求无法明确且经常变化，开发人员无法与用户进行充分的交流，导致开发人员很难真正掌握客户的确切需求，在这样的情况下进行“闭门造车”式的软件开发，必将导致开发出来的产品与客户实际需求不相符合。

③ 软件产品的质量无法保证。由于软件开发过程中的技术缺陷或管理不科学等原因，往往无法达到软件的可靠性和质量保证要求，导致软件产品发生质量问题。

④ 软件产品难以维护。软件维护是在软件开发完成后的艰巨又复杂的工作，并且花费代价也很高。由于在开发过程中的技术原因、管理原因，或者对软件问题的认识错误，导致软件问题在运行维护过程中无法修正，或者修正代价特别高。

⑤ 软件产品缺乏适当的文档。软件不仅仅是指程序，还应包括一套完整的相关文档。无论是开发过程中，还是维护过程中，这套文档资料是至关重要、必不可少的。如果缺乏必要的文档资料，必然给软件开发和维护带来许多严重的困难和问题。

⑥ 软件开发生产率跟不上计算机应用的发展趋势。软件开发的效率跟不上硬件的发展速度，也跟不上用户对计算机应用需求的增长，必将无法及时交付合适的软件，导致软件最终满足不了社会发展需求的高速增长。

软件在开发和维护过程为什么会出现那么多严重问题而导致产生“软件危机”呢？原因是多方面的，但总的来说，可以归结为客观和主观两方面的原因。在客观方面是由于软件本身的特点而导致的，在主观方面是在软件开发和维护过程中所采用的方法不正确是一个重要原因。

软件是一种逻辑实体，不同于物理实体，不具备外观形状、质量等物理特性，是一种无明显“可见性”的逻辑产品，在开发过程中遇到的影响因素和问题是无法完全掌控的，因而，在开发过程中很难预见将来在运行时会出现什么问题。一般情况下，软件开发过程是复杂的，由许多人一起合作完成，他们的技术、经验和方法都不同，更容易产生疏漏和错误。

软件最终是解决用户的需求问题，但在开发过程中，用户和开发人员就软件的需求往往不能达成百分之百的共识，用户无法描述其真正的全部需求，尤其是在开发初期，开发人员无法准确地、完整地理解用户的实际需求。同时，在开发过程不断变更需求，必将导致软件系统的不稳定性和不可靠性，这些将带来很多问题隐患。

在软件开发和维护过程中，缺乏科学的方法、技术和工具，过程管理混乱，以及“轻需求、重编程”“轻维护、重开发”“轻文档、重代码”等错误的观念，严重降低了软件质量、可靠性和稳定性，大大提高了“软件危机”的发生率。

随着软件危机的不断出现，以及软件规模越来越大，人们认识到过程管理混乱、文档繁杂而无规范、开发方式作坊化无法保证软件的质量，无法避免软件危机的出现。因此，人们开始探索用“工程化”的思想来指导软件开发，即在软件开发、管理、维护和更新过程中，使用软件工程的概念、原理、技术和方法，从技术方面和管理方面采取相应的措施，解决软件开发过程中的混乱和困难，力求从根本上解决软件危机。逐渐形成了一个新的领域——“软件工程”。

4. 软件工程的定义

“软件工程”的首次提出，是于1968年北大西洋公约组织（NATO）在联邦德国召开的一次会议上。主要思想是“把软件当成一种工业产品”，要求采用工程化的原理与方法对软件进行计划、开发和维护，以实现生产高质量、低成本的软件产品的目标。

1983 年, IEEE 给软件工程的定义是:“软件工程是开发、运行、维护和修复软件的系统方法。”软件工程是研究把系统化、规范化、可度量的工程化途径应用于软件开发、运行和维护的过程。

软件工程涉及软件开发技术和软件工程管理两个方面。2004 年, IEEE 发布的《软件工程知识体系指南》(SWEBOK) 中将软件工程知识体系划分为 10 个知识领域。

- 软件需求 (software requirements)。
- 软件设计 (software design)。
- 软件构建 (software construction)。
- 软件测试 (software testing)。
- 软件维护 (software maintenance)。
- 软件配置管理 (software configuration management)。
- 软件工程管理 (software engineering management)。
- 软件工程过程 (software engineering process)。
- 软件工程工具和方法 (software engineering tools and methods)。
- 软件质量 (software quality)。

概括地说, 软件工程就是指导计算机软件开发和维护的工程学科。采用工程化的概念、原理、技术和方法来开发和维护软件, 把科学的、正确的、已经验证的管理技术与当前最好的开发技术和方法结合起来, 降低开发成本, 缩短开发周期, 提高软件质量, 从而避免软件危机。

5. 软件工程方法学

软件工程方法学是指在软件生命周期全过程中使用的一套完整的工具、技术和方法。以关注软件质量为目标, 由方法、工具和过程 3 个要素构成。其中, 方法为软件开发中完成各种任务提供“如何做”的技术和方法, 涉及项目成本估算、软件需求分析、软件设计、软件实现和软件维护等各个方面。工具为软件工程方法提供自动或半自动的软件支撑环境。过程将工程的方法和工具综合起来, 以达到合理地、及时地进行软件开发的目的, 为开发出高质量的软件提供了一系列的需要完成的任务框架和工作步骤。

目前, 使用最广泛的软件工程方法学主要有传统方法学和面向对象方法学。

(1) 传统方法学

传统方法学也称为生命周期方法学或结构化法。在完成软件分析、设计和实现的各项任务时采用结构化技术, 并使用适当的软件工具或软件工程环境支持结构化技术的运用。这种方法学将软件生命周期的全过程依次划分多个阶段, 每个阶段都有相对独立的任务, 然后顺序地完成每个阶段的任务。各个阶段有严格的先后次序, 前一个阶段的工作完成和工作成果是后一个阶段的前提和基础, 每个阶段的开始和结束都有严格的标准。每个阶段都有标志性的工作完成或标志性的成果产生, 每个阶段的开发成果必须经过严格的检查和评审, 评审通过, 该阶段的工作才算结束, 然后进入下一个阶段的开发工作, 如果评审没有通过, 则须返工, 返工后再进行检查和评审, 直到评审通过。这样每个阶段都有高质量的工作成果, 尤其是具有规范的文档资料, 提高了软件开发的成功率, 保证了软件的质量, 提高了软件的可维护性。

传统方法学的明显特点是自顶向下、逐步求精, 有严格的阶段性评审。这种方法具有操作简单、降低软件开发的复杂度、保证了软件质量等优点。但也有缺点, 在开发前期, 就需要了解详细的、明确的用户需求, 这往往是不现实的, 也难于适应需求变化频繁的系统, 因此, 不

太适合前期需求不明确、需求变化频繁的软件项目。

(2) 面向对象方法学

当软件规模庞大、需求复杂且变化频繁的时候，采用传统方法学进行软件开发不是一个很好的选择，这时选择面向对象方法学可以达到理想的效果。面向对象方法学引入了类和对象的概念，尽量模拟人类的思维习惯，把人类认识世界、解决问题的方法和过程应用到软件工程领域。对象是融合了数据及在数据上操作行为的一个软件构件，面向对象程序是由不同的对象构成，对象是程序的构成元素，对象和对象之间通过消息进行通信。具有相同数据和相同操作的一组相似对象抽象成一个类。同时，通过父类和子类的继承关系，使若干个相关的类形成一个层次结构的系统。

面向对象方法学在软件的分析、设计和实现等过程中都采用面向对象技术，其明显的特点是以对象作为基本的构成要素，利用抽象、分类、继承、聚合、封装等人类常用思维习惯分析和设计系统。使用面向对象方法开发软件的过程是一个多次反复迭代的演化过程。通过概念上和表示方法上的一致性，保证了开发过程中各项任务的平滑的过渡和实现。降低了软件开发的复杂度，提高了软件的可理解性和可维护性，并且更容易地实现了软件的重用，降低了软件开发成本。

课堂实训

- 列举当今市场上主流的系统软件产品，并对这些软件产品做简单介绍。
- 列举几个软件危机的现象，并分析产生这些危机的主要原因。

【任务 1-2】 认识软件生命周期

软件如同现实世界的客观物体一样，具有生命周期，软件的生命周期就是从软件提出到软件废弃的全过程。软件生命周期是软件工程中的一个重要概念。一个软件的生命周期可划分为多个相互区别又相互联系的不同阶段，每个阶段都有各自相对独立的任务，每个阶段的相关开发人员各有自己的职责范围，分工协作，保证软件开发各项任务顺利完成。

任务描述

软件生命周期一般由软件定义、软件开发和软件运行维护 3 个时期组成。每个时期又可以进一步划分成若干阶段。本任务将描述软件生命周期各个阶段的划分以及主要任务。

- 软件定义时期。
- 软件开发时期。
- 软件运行维护时期。

任务实现

1. 软件定义时期

软件定义时期是完成软件规划和需求分析的阶段。其主要任务是：确定软件系统“要解决什么”，确定软件工程必须达到的总目标；从技术方面、资源方面和管理方面进行可行性研究；分析并确定软件用户的实际需求。软件定义时期可以进一步划分为问题定义、可行性研究和需