现代软件工程应用技术

杨晶洁 主编

北京理工大学出版社

现代软件工程应用技术

主编　杨晶洁

北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

本书以一个完整项目——企业设备状况管理系统为主线，用软件工程的思想进行分析和设计。

本书共分 10 个项目，分别介绍了软件工程概述、UML 建模软件以及 Microsoft Office Visio 2010 的简单使用方法、基于“赠品管理系统”的结构化软件需求分析方法、基于“企业设备状况管理系统”的面向对象软件需求分析方法、基于“企业设备状况管理系统”的系统设计方法、基于“企业设备状况管理系统”的详细设计方法、基于“企业设备状况管理系统”的实现方法、软件项目的测试与维护、软件文档书写方法、Project 2013 等内容。

本书适合作为高等院校计算机类专业的教材使用，也可作为软件开发人员的参考用书。

图书在版编目（CIP）数据

现代软件工程应用技术 / 杨晶洁主编. —北京：北京理工大学出版社，2017.5
ISBN 978-7-5682-3989-9

Ⅰ. ①现…　Ⅱ. ①杨…　Ⅲ. ①软件工程　Ⅳ. ①TP311

中国版本图书馆 CIP 数据核字（2017）第 091693 号

出版发行 / 北京理工大学出版社有限责任公司
社　　址 / 北京市海淀区中关村南大街 5 号
邮　　编 / 100081
电　　话 /（010）68914775（总编室）
　　（010）82562903（教材售后服务热线）
　　（010）68948351（其他图书服务热线）
网　　址 / http://www.bitpress.com.cn
经　　销 / 全国各地新华书店
印　　刷 /
开　　本 / 787 毫米×1092 毫米　1/16
印　　张 / 16.75
字　　数 / 389 千字
版　　次 / 2017 年 5 月第 1 版　2017 年 5 月第 1 次印刷
定　　价 / 60.00 元

责任编辑 / 封　雪
文案编辑 / 封　雪
责任校对 / 周瑞红
责任印制 / 李志强

前　言

本书是以任务驱动教学为主线，结合实际案例，配合 UML 统一建模语言和软件工程技术课程的教学而编写的，目的是通过案例设计的综合训练，培养学生实际分析问题、解决问题的能力，帮助学生系统地掌握该门课程的主要内容，更好地完成教学任务。

本书特点

本书以“企业设备状况管理系统”项目为主线，将“企业设备状况管理系统”项目分成不同的任务。每个任务既相对独立又有一定的连续性，教学活动的过程是完成每一个任务的过程。完成“企业设备状况管理系统”项目调研、需求、分析、设计的过程，也就是完成每一个任务的过程。

本书编写偏重于面向对象的分析和设计思想的描述，对面向过程的分析和设计也做了相应的描述。本书与其他同类教材相比有以下优点：

（1）以项目调研、需求、分析、设计、开发为主线；

（2）以任务驱动案例教学为核心；

（3）先有项目讲解，后有实验实训，达到学中做的效果；

（4）本书以一个完整项目——企业设备状况管理系统为主线，用软件工程的思想进行分析、设计，学习完项目，也就完成了对本书知识点的学习。

本书共分 10 个项目，各项目内容简介如下：

项目一：主要介绍软件工程及其相关知识内容。

项目二：主要介绍 UML 建模软件以及 UML 可视化面向对象建模工具——Microsoft Office Visio 2010 的简单使用方法。

项目三：主要介绍基于“赠品管理系统”的结构化软件需求分析方法。

项目四：主要介绍基于“企业设备状况管理系统”的面向对象软件需求分析方法。

项目五：主要介绍基于“企业设备状况管理系统”的系统设计方法。

项目六：主要介绍基于“企业设备状况管理系统”的详细设计方法

项目七：主要介绍基于“企业设备状况管理系统”的实现方法。

项目八：主要介绍软件项目的测试与维护。

项目九：主要介绍基于“企业设备状况管理系统”的软件文档书写方法。

项目十：主要介绍基于“企业设备状况管理系统”的项目管理工具——Project 2013。

本书由杨晶洁主编。杨晶洁负责全书大纲和所有章节的编写，以及对全书各章的修改和审定。本书在编写过程中参阅了国内外同行编著的相关论著，在此致以诚挚的谢意。由于编者水平有限，书中有不当之处，敬请读者多提宝贵意见。

编　者

2017.1.1

目　　录

项目一

软件工程概述

项目导读

在研究软件工程技术之前，先来讨论一下在计算机系统中硬件和软件是如何协作工作的，人们常常认为硬件控制软件，实际上硬件是不会控制软件的，而是软件去监控硬件的工作状态，然后再做出反应。计算机软件通过计算机硬件及其相关设备实现功能，硬件受控于软件，在一个系统中两者缺一不可。软件控制硬件的过程如下：软件编程人员编写的程序通过汇编编译器翻译成硬件可以读懂的语言（二进制代码），然后硬件根据这个二进制文件执行相应的操作。如果一个软件的设计、开发过程缺乏科学而有效的实施方法，那么软件系统实际运行时，就很容易漏洞频出，修改、完善困难。因此，一个计算机软件的成功开发和应用，就是要满足用户的业务需求，在一定周期内无差错地运行，并经过实践证明确实能够帮助用户提高核心竞争力、推动业务更好的发展。在实际工作中，用户都非常希望开发有利于自身业务良好发展的软件，避免可能出现的糟糕情况。

“工程”一词是指科学和数学的某种应用，通过这一应用，自然界的物质和能源的特性能够通过各种结构、机器、产品、系统和过程，以最短的时间和精而少的人力做出高效、可靠且对人类有用的东西。一言以蔽之，工程是将自然科学的理论应用到具体工农业生产部门中形成的各学科的总称。因而，“软件工程”（software engineering，SE）是一门研究用工程化方法构建和维护有效的、实用的和高质量的软件的学科。它涉及程序设计语言、数据库、软件开发工具、系统平台、标准、设计模式等方面。在设计和开发软件时，工程化方法与标准化工作往往可以帮助开发者和用户达到共同的期望。在现代社会中，软件应用于多个方面。典型的软件有游戏、嵌入式系统、办公套件、编译器、数据库等。软件工程的思想和方法是针对软件危机提出来的。软件危机出现的特征一方面表现在软件项目开发过程中，完工日期拖后、经费超支，甚至造成工程最终宣告失败等情况；另一方面从整个社会对软件产品的需求而言，软件危机的出现实质是软件产品的供应赶不上需求的增长。1968 年，北大西洋公约组织（NATO）在联邦德国召开的一次国际会议上，计算机科学家们讨论了软件危机的问题，正式提出并使用了“软件工程”这个名词，一门新兴的学科就此诞生了。

项目概要

- 软件简介
- 软件工程产生的背景
- 软件项目的生命周期

- 软件项目的开发模型
- 软件技术的发展趋势

读者学习完一些高级编程语言之后，往往习惯性地认为软件工程课程就是教给大家如何开发软件的，而开发软件就是编写程序，这样的认识是片面的，所以，下面先来了解一下软件的发展过程。

任务1.1 软 件 简 介

1.1.1 人们对软件的认识

20 世纪 40 年代，随着第一台计算机“ENIAC”的诞生，一次科学技术革命（即信息技术革命）开始了，人类社会、生活、办公都发生了巨大的变化。微型计算机以其令人瞠目结舌的速度发展，在科学、军事、经济等社会领域得到越来越广泛的应用，极大地改变了人们原有的工作和生活面貌。在计算机硬件方面，尤其是微处理器日新月异的更新速度，促进了整个运算体系的发展，计算机程序也从硬件中分离出来，从而逐渐形成了软件技术的概念。经过了 70 多年的发展演变，人们对软件有了更为深刻的认识，同时对相应软件研究和应用技术提出了越来越严格的要求。

在学习软件工程之前，首先要先理解两个正确的观点：

1）开发软件不等于编写程序

在软件开发过程中，编写程序只是开发软件所应完成工作的一部分，具体的软件开发工作包括以下几个方面。

（1）问题的定义及规划：开发方调研用户需求及用户环境，开发方和需求方论证项目的技术、经济、市场等可行性并制订项目初步计划。

（2）需求分析：开发方确定系统的运行环境、建立逻辑模型、确定系统的功能和性能要求。

（3）软件设计：包括概要设计和详细设计。

概要设计：建立系统总体结构、划分功能模块、定义各个功能模块的接口，制订测试计划。

详细设计：设计各个模块的具体实现算法，确定各个模块间的详细接口，制订测试方案。

（4）程序编码：编写程序源代码、进行模块测试和调试，编写用户手册。

（5）软件测试：单元测试、集成测试、系统测试、编写测试报告。

（6）实现和运转：对修改进行配置管理，记录修改记录和故障报表，按照用户和软件设计的共同意见进行软件维护。

2）错误做法会导致软件危机

20 世纪 60 年代以前，计算机刚刚投入实际使用，软件设计往往只是为了一个特定的应用而在指定的计算机上进行设计和编制，采用密切依赖于计算机的机器代码或汇编语言，软件的规模比较小，文档资料通常也不存在，很少使用系统化的开发方法，设计软件往往等同于编制程序，基本上是个人设计、个人使用、个人操作、自给自足的私人化的软件生产方式。

60 年代中期，大容量、高速度计算机的出现，使计算机的应用范围迅速扩大，软件开发急剧发展：高级语言开始出现；操作系统的发展引起了计算机应用方式的变化；大量数据处理导致第一代数据库管理系统的诞生。软件系统的规模越来越大，复杂程度越来越高，软件可靠性问题也越来越突出。原来的个人设计、个人使用的方式不再能满足要求，迫切需要改变软件生产方式，提高软件生产率，软件危机开始爆发。软件危机是在开发和维护计算机软件的过程中所遇到的一系列严重问题，主要包含两方面的问题，如何开发软件以满足用户对软件日益增长的需求和如何维护数量不断膨胀的已有软件。

1.1.2　软件的发展

软件的发展主要经历了 4 个阶段。

1）第一阶段：20 世纪 50—60 年代

第一阶段也称为程序设计阶段。最初的二进制机器指令语言程序逐渐被汇编语言程序代替，程序是专为满足某个具体应用而编写的。这个阶段的生产方式是个体手工方式，在这个阶段，硬件成本非常昂贵，程序规模小，占用内存空间也较小。软件的设计通常是在开发者的头脑中进行的，没有什么程序设计方法，除了程序清单代码之外，没有其他文档能有效地保存下来。

2）第二阶段：20 世纪 60—70 年代

第二阶段也称为程序系统阶段。计算机软件程序出现系统化发展，这个时期计算机语言发展很快，出现了应用性高级语言，如 Basic、Pascal、Fortran 等。这个阶段的软件生产方式仍然是个体化开发方法，程序开发出现“软件作坊”形式。这个阶段的硬件价格开始降低，速度、容量、可靠性明显提高，但是此阶段的软件产品的开发仍然没有相应配套的管理体系，出现了运行质量低下、维护工作繁杂甚至不可维护等问题。“软件危机”随之出现。

3）第三阶段：20 世纪 70—90 年代

第三阶段也称为软件工程阶段。开始出现高级语言系统、数据库、网络及分布式开发等。这个阶段的软件生产方式是工程化生产，计算机硬件成本的大幅下降，同时计算机性能的快速提高促使计算机迅速普及，各类用户对计算机软件的需求不断高涨，推动了软件生产走向市场化，同时也迫使软件开发成为一门新兴的工程学科，即软件工程。软件工程技术对软件的开发技术、方法进行改进，如结构化的设计、分析方法和原型化的方法促进了软件生产的过程化和规范化。软件管理在软件生产中起着重要作用，但是，尚未完全摆脱软件危机。

4）第四阶段：20 世纪 90 年代以来

第四阶段也称为现代软件工程阶段。软件生产走向了项目工程生产方式，出现了大量的新技术，比如：面向对象技术、嵌入式系统、分布式系统和智能系统等复杂程度高、应用规模大的计算机系统日益增多。软件开发进入成熟发展阶段，软件成为人类必不可少的工具。

1.1.3　软件的分类及特点

1）软件的定义

软件（software）是计算机系统中与硬件相互依存的另一部分，是包含程序、数据及其相关文档的完整集合，即软件=程序+数据+相关文档。其中，程序是按事先设计的功能和性能要求执行的指令序列；数据是使程序能正常操纵信息的数据结构；文档是与程序开发、维护和使用等相关的图文材料。可以这样理解：

（1）软件=程序+数据+文档。

（2）面向过程的程序=算法+数据结构。

（3）面向对象的程序=对象+类+继承+消息。

（4）面向构件的程序=构件+构架。

2）软件的分类

软件是一种逻辑实体，不是具备一定形状的物理实体，可以把它保存在计算机的存储器内部，也可以保留在磁盘、光盘和U盘等介质上，但是无法看到软件的形态，必须通过观察、分析、思考、判断，去了解它的功能、性能及其他特性。软件的生产与硬件不同。在软件开发过程中没有明显的制造过程，它是通过人们的智力活动，把知识和技术转化成信息的一种产品。当某一软件项目研制成功后，就可以大量地复制同一内容的副本。

软件的开发和运行常常受到计算机系统的限制，对计算机系统有着不同程度的依赖性，软件可以有以下4种划分方式。

（1）按软件的规模划分。软件的规模可以采用代码行（LOC）的数量或耗用的时间、人力来度量，见表1–1。

表1–1　软件规模划分

规模	代码行数	时间	人数
微型	500以下	1～4周	1人
小型	2 000以下	半年	1人
中型	5 000～50 000	1～2年	2～5人
大型	5万～10万	2～3年	5～20人
超大型	100万以上	4年以上	100人以上

（2）按软件工作方式划分。按软件工作方式划分，软件可分为实时、分时、交互式和批处理软件等几种。

① 实时软件是对当前时间当前任务做的处理。如：卫星实时监控软件等。

② 分时软件是阶段性地处理任务的软件。它按照一定的时间间隔处理任务。如：交通岗红绿灯控制软件。

③ 交互式软件是相互性的。可以处理执行任务，也可以产生一个任务让其他设备或软件完成。如：人们使用的各种交友软件。

④ 批处理软件是一次可以执行多条指令的软件。如：垃圾处理软件。

（3）按软件应用的功能划分，见表1–2。

表1–2　软件功能划分

名称	内　　容
系统软件	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	编译软件，文本编辑器，支持需求分析、设计、实现、测试和管理的软件等
应用软件	数据处理软件、计算机辅助设计软件、系统仿真软件、人工智能软件、办公自动化软件、计算机辅助教学软件等

（4）按软件服务对象的范围划分。按软件服务对象的范围划分，可分为项目软件和产品软件。

① 项目软件。项目软件是软件开发机构受特定用户委托而开发的软件。如：商品管理系统、生产过程控制等。一般情况下，项目软件是在合同的约束下开发的。

② 产品软件。产品软件是软件开发机构直接为市场开发的软件。如：文字处理软件、多媒体播放软件、游戏软件等。产品软件的功能、性能、价格和售后服务对开发机构参与市场竞争有重要影响。

3）软件的特点

（1）软件模型有更强的表达能力、更符合人类的思维模式，属于人类抽象层次的一种，是一种逻辑实体，而不是物理实体。软件是对客观世界中问题空间与解空间的具体描述，是客观事物的一种反映，是知识的提炼和“固化”。

（2）软件生产没有明显的制造过程，在高级语言出现以前，汇编语言（机器语言）是编程的工具，表达软件模型的基本概念是指令，显然，这都是抽象层次的。

（3）软件相对于硬件来讲，没有磨损、老化这些问题，但是需要按照实际用户的需求进行更新、升级。

（4）虽说软件控制硬件，但软件对计算机系统的硬件还是有不同程度的依赖。

（5）软件的开发依赖人工，主要是由它的复杂性决定的，所以，对软件的开发尚未摆脱手工操作。

（6）随着时间的推移，软件的开发成本越来越高。

（7）客观世界是不断变化的，因此，构造性和演化性是软件的本质特征。

高级语言的出现，如 Fortran 语言、Pascal 语言、C 语言等，使用了变量、标识符、表达式等概念作为语言的基本构造，并使用 3 种基本控制结构来表达软件模型的计算逻辑，因此软件开发人员可以在一个更高的抽象层次上进行程序设计。随后出现了一系列开发模型和结构化程序设计技术，实现了模块化的数据抽象和过程抽象，提高了人们表达客观世界的抽象层次。并使开发的软件具有一定的构造性和演化性。面向对象程序设计语言逐步流行，为人们提供了一种以对象为基本计算单元、以消息传递为基本交互手段的软件模型。面向对象方法的实质是以拟人化的观点来看待客观世界，即客观世界由一系列对象构成，这些对象之间的交互形成了客观世界中各式各样的系统。面向对象方法中的概念和处理逻辑更接近人们解决计算问题的思维模式，使开发的软件具有更好的构造性和演化性。

（8）人们更加关注软件复用问题，构造比对象更大且易于复用的基本单元——构件，并研究以构件复用为基础的软件构造方法，更好地凸显软件的构造性和演化特性。易于复用的软件，一定是具有很好构造性和演化性的软件。

任务 1.2　软件工程的产生

计算机软件系统通过运行程序来实现各种不同目的的应用。按照所实现功能的不同，程序包括用户为自己特定目的编写的程序、检查和诊断机器系统的程序、支持用户应用程序运行的系统程序、管理和控制机器系统资源的程序等。软件不同于硬件，它是计算机系统中的逻辑部件，是程序开发、使用和维护所需要的文档。美国电气和电子工程师学会（Institute of

Electrical and Electronics Engineers，IEEE）对软件进行的描述是：计算机程序、方法、规则和相关文档资料以及在计算机上运行时所必需的数据。

美国电气和电子工程师学会

1.2.1 软件危机的故事

1. 软件危机实例

1995 年，Standish Group 研究机构以美国境内 8 000 个软件项目作为调查样本进行调查，调查结果显示，有 84%软件计划无法于既定时间、经费中完成，超过 30%的项目于运行中被取消，项目预算平均超出 189%。

危机实例一：IBM OS/360

IBM OS/360 操作系统被认为是一个典型的软件危机案例。到现在为止，它仍然被使用在 360 系列主机中。这个经历了数十年、极度复杂的软件项目甚至产生了一套不包括在原始设计方案之中的工作系统。OS/360 是第一个超大型的软件项目，它使用了 1 000 人左右的程序员。佛瑞德·布鲁克斯在他的大作《人月神话》中承认，在管理这个项目的时候，他犯了一个价值数百万美元的错误。

危机实例二：美国银行信托软件系统开发案

美国银行 1982 年进入信托商业领域，并规划发展信托软件系统。项目原订预算 2 000 万美元，开发时程 9 个月，预计于 1984 年 12 月 31 日以前完成，但至 1987 年 3 月都未能完成该系统，且已投入 6 000 万美元。美国银行最终因为此系统不稳定而不得不放弃，并将 340 亿美元的信托账户转移出去，失去了 6 亿美元的信托生意商机。

2. 软件危机主要表现

（1）软件开发进度难以预测。拖延工期几个月甚至几年的现象并不罕见，这种现象降低了软件开发组织的信誉。

（2）软件开发成本难以控制。投资一再追加，令人难于置信。往往是实际成本比预算成本高出一个数量级。而为了赶进度和节约成本所采取的一些权宜之计又往往损害了软件产品的质量，从而不可避免地会引起用户的不满。

（3）用户对产品功能难以满足。开发人员和用户之间很难沟通，矛盾很难统一。往往是因为软件开发人员不能真正了解用户的需求，而用户又不了解计算机求解问题的模式和能力，双方无法用共同熟悉的语言进行交流和描述。双方在互不充分了解的情况下，就仓促上阵设计系统、匆忙着手编写程序，这种“闭门造车”的开发方式必然导致最终的产品不符合用户的实际需要。

（4）软件产品质量无法保证。系统中的错误难以消除，软件是逻辑产品，质量问题很难以统一的标准度量，因而造成质量控制困难。软件产品并不是没有错误，而是盲目检测很难发现错误，而隐藏下来的错误往往是造成重大事故的隐患。

（5）软件产品难以维护。软件产品本质上是开发人员的代码化的逻辑思维活动，他人难以替代。除非是开发者本人，否则很难及时检测、排除系统故障。为使系统适应新的硬件环境，或根据用户的需要在原系统中增加一些新的功能，又有可能增加系统中的错误。

（6）软件缺少适当的文档资料。文档资料是软件必不可少的重要组成部分。实际上，软件的文档资料是开发组织和用户之间的权利与义务的合同书，是系统管理者、总体设计者向

开发人员下达的任务书，是系统维护人员的技术指导手册，是用户的操作说明书。缺乏必要的文档资料或者文档资料不合格，将给软件开发和维护带来许多严重的困难与问题。

3. 软件危机产生的原因

1）用户需求不明确

在软件开发过程中，用户需求不明确问题主要体现在四个方面：

（1）在软件开发出来之前，用户自己也不清楚软件开发的具体需求；

（2）用户对软件开发需求的描述不精确，可能有遗漏、有二义性，甚至有误；

（3）在软件开发过程中，用户还提出修改软件开发功能、界面、支撑环境等方面的要求；

（4）软件开发人员对用户需求的理解与用户本来愿望有差异。

2）缺乏正确的理论指导

软件开发缺乏有力的方法学和工具方面的支持。由于软件开发不同于大多数其他工业产品，其开发过程是复杂的逻辑思维过程，其产品极大程度地依赖于开发人员高度的智力投入。过分地依靠程序设计人员在软件开发过程中的技巧和创造性，加剧了软件开发产品的个性化，也是发生软件开发危机的一个重要原因。

3）软件开发规模越来越大

随着软件开发应用范围的增广，软件开发规模越来越大。大型软件开发项目需要组织一定的人力共同完成，而多数管理人员缺乏开发大型软件开发系统的经验。各类人员的信息交流不及时、不准确，有时还会产生误解。软件开发项目开发人员不能有效地、独立自主地处理大型软件开发的全部关系和各个分支，因此容易产生疏漏和错误。

4）软件开发复杂度越来越高

软件开发不仅是在规模上快速地发展扩大，而且其复杂性也急剧增加。软件开发产品的特殊性和人类智力的局限性，导致人们无力处理“复杂问题”。所谓“复杂问题”的概念是相对的，一旦人们采用先进的组织形式、开发方法和工具提高了软件开发效率和能力，新的、更大的、更复杂的问题又摆在人们的面前。

4. 软件危机的解决途径

软件工程诞生于 20 世纪 60 年代末期，它作为一个新兴的工程学科，主要研究软件生产的客观规律性，建立与系统化软件生产有关的概念、原则、方法、技术和工具，指导和支持软件系统的生产活动，以期达到降低软件生产成本、改进软件产品质量、提高软件生产率水平的目标。软件工程学从硬件工程和其他人类工程中吸收了许多成功的经验，明确提出了软件生命周期的模型，发展了许多软件开发与维护阶段适用的技术和方法，并应用于软件工程实践，取得良好的效果。在软件开发过程中人们开始研制和使用软件工具，用以辅助进行软件项目管理与技术生产，人们还将软件生命周期各阶段使用的软件工具有机地集合成为一个整体，形成能够连续支持软件开发与维护全过程的集成化软件支援环境，以期从管理和技术两方面解决软件危机问题。

此外，人工智能与软件工程的结合成为 20 世纪 80 年代末期活跃的研究领域。基于程序变换、自动生成和可重用软件等的软件新技术研究也已取得一定的进展，把程序设计自动化的进程向前推进一步。在软件工程理论的指导下，发达国家已经建立起较为完备的软件工业化生产体系，形成了强大的软件生产能力。软件标准化与可重用性得到了工业界的高度重视，在避免重用劳动、缓解软件危机方面起到了重要作用。