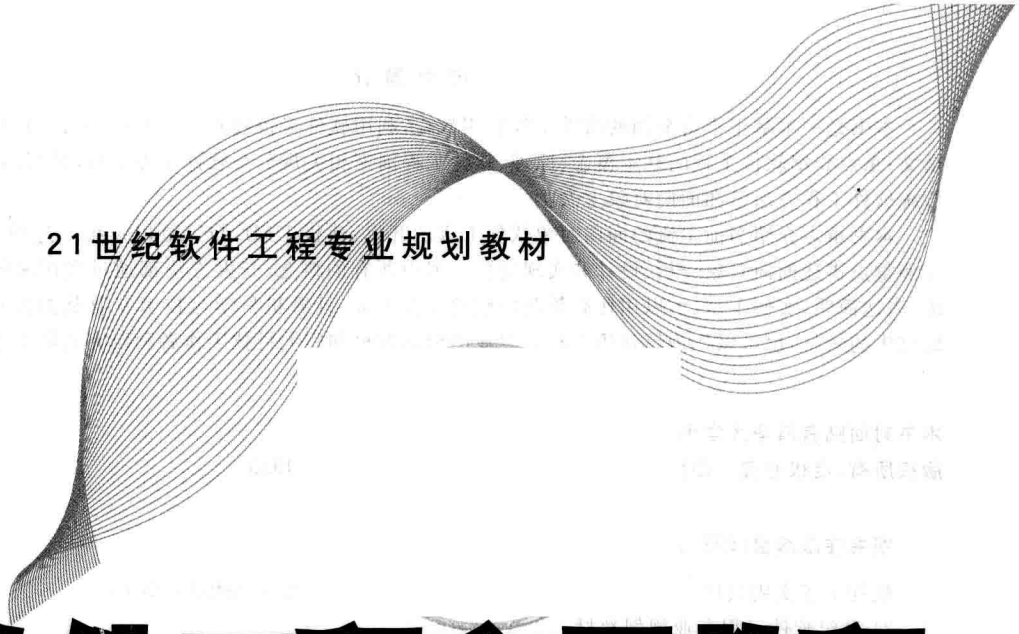
高等学校计算机教材建设立项项目

21世纪软件工程专业规划教材

# 软件工程实用教程

田保军 刘利民 主编
张林丰 张丽霞 许志伟 编著

清华大学出版社



21世纪软件工程专业规划教材

# 软件工程实用教程

田保军 刘利民 主编
张林丰 张丽霞 许志伟 编著

清华大学出版社
北京

## 内 容 简 介

本书是一本基于丰富案例的软件工程实用教程，利用软件工程核心三要素（方法、工具和过程）贯穿全文；重点介绍软件工程的基本概念、原理、软件工程国家相关规范与软件工程文档，撰写国家标准以及传统软件工程方法学和面向对象方法学。

本书重点介绍当前主流的面向对象软件工程的开发、建模 UML 与工具以及统一过程 RUP。通过实例重点讲述面向对象分析、设计和实现流程。书中所有的概念、开发方法都通过实例来演示，内容精练，表达简明，实例丰富，可以用作高等院校软件工程专业、计算机科学与技术专业及相关专业本科生、研究生的教材，也可以作为培训机构相关专业的培训教材和广大科技工作者和研究人员参考。



**图书在版编目(CIP)数据**

软件工程实用教程 / 田保军，刘利民主编．--北京：清华大学出版社，2015
21 世纪软件工程专业规划教材
ISBN 978-7-302-40761-4

Ⅰ．①软…　Ⅱ．①田…　②刘…　Ⅲ．①软件工程－高等学校－教材　Ⅳ．①TP311.5

中国版本图书馆 CIP 数据核字（2015）第 161622 号

**责任编辑**：张　玥　赵晓宁
**封面设计**：傅瑞学
**责任校对**：焦丽丽
**责任印制**：刘海龙

**出版发行**：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　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
课 件 下 载：http://www.tup.com.cn，010-62795954

**印 装 者**：北京密云胶印厂
**经　　销**：全国新华书店
**开　　本**：185mm×260mm　　**印　　张**：21.75　　**字　　数**：497 千字
**版　　次**：2015 年 8 月第 1 版　　**印　　次**：2015 年 8 月第 1 次印刷
**印　　数**：1～2000
**定　　价**：44.50 元

产品编号：061779-01

# 前言

PREFACE

软件工程作为支撑软件产业的一级学科，其发展正方兴未艾。“软件工程”课程是SWEBOK软件工程知识体系中一门基础、核心课程。“软件工程”课程涉及的内容广泛，其涉及的各项技术和项目管理方法对于即将从事IT产业的学生来说是非常重要的。但是由于种种原因，许多学生认为这门课程比较空洞乏味。作者结合多年的教学和工程实践经验，参阅大量国内外有关软件工程的教材和资料，遵循“理论为基础、实用为目的”的原则，理论联系实践，编写本书。

本书着重从实用角度出发，讲解目前软件工程比较成熟的、广泛使用的两大方法学——结构化方法学和面向对象方法学。以软件生命周期为主线，基于软件工程核心三要素(方法、工具和过程)贯穿全文。内容主要包括软件工程概述、可行性与计划研究、需求分析、软件设计、软件实现、软件测试、软件运行与维护、软件过程管理，同时介绍当今流行的软件工程建模语言和工具，如面向对象建模语言UML、软件绘图工具Microsoft Visio、数据建模工具PowerDesigner、面向对象建模工具Rational Rose与StarUML、测试工具LoadRunner、Quality Center与QuickTest Professional等。并且，为了帮助学生通过“做中学”的模式掌握扎实而实用的软件工程技术，本书以学生学籍管理系统为项目案例，贯穿全文。

本书内容丰富，组织结构严谨，原理、方法与案例相结合，讲解由浅入深，既体现知识点的连贯性、完整性，又体现知识在实际中的应用。

本书在内容的编排、语言的叙述等方面都有一些特点：

(1) 内容系统全面，结构清晰。全书分为两大部分：面向过程的软件工程和面向对象的软件工程，按照软件生命周期的各个阶段分别进行讲述。

(2) 描述简明易懂。本书从基本概念和原理出发，注重内容的可理解性，深入浅出，循序渐进；文字描述通俗易懂，简明扼要，重点突出。

(3) 注重案例分析。以学生学籍管理系统为案例贯穿全文是本书的最大特色。本书不仅增加了案例数量，而且保持案例的连续性，使读者更容易掌握相关知识。

(4) 每章列出学习目标和小结，配有精选的适量习题，便于学生对所学内容的复习和理解。

本书由田保军和刘利民担任主编，张林丰、张丽霞和许志伟参与本书的编写。参加编写的人员及分工如下：田保军编写第5、第10和第11章；刘利民编写第1和第3章；张林丰编写第6～第8章；张丽霞编写第4和第9章；许志伟编写第2章、附录A～C。全书由田保军、刘利民统稿。

本书参考和引用了许多教材、著作和网站内容，除了确实无法查证出处以外，在参考文献中都一一列出，在此对作者们表示衷心感谢。张志林、胡皎月、王宇、胡培培等研究生在教材的编写过程中，也做了不少工作，一并表示感谢。由于水平有限，书中不够完善乃至缺点和错误之处，恳请专家、学者提出宝贵意见，以便我们再版时进行修订补充，使之日臻完善。

编　者

于内蒙古工业大学

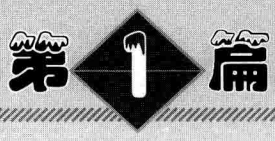
2015 年 2 月

# 目 录

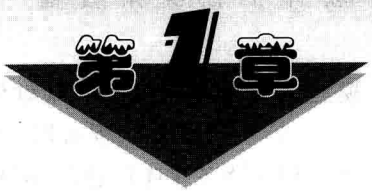
CONTENTS

## 第 1 篇　面向过程的软件工程

## 第 2 篇 面向对象的软件工程




# 面向过程的软件工程




# 软件工程概述

**本章学习目标**

- 熟练掌握软件工程概念、原理和方法；
- 熟练掌握软件生命周期和软件过程模型；
- 掌握软件危机产生的原因和解决办法；
- 掌握软件及其特点；
- 了解软件工程的发展历程；
- 了解软件工程相关规范。

本章首先向读者介绍软件工程的发展历程；然后介绍软件及其特点、软件危机产生的原因和解决办法；继而介绍软件工程原理和方法；最后介绍软件工程相关规范。

## 1.1 软件工程的发展历程

随着计算机的广泛应用，软件在计算机系统中的地位越来越重要。市场需要的软件逐年增多，而且趋向大型化和复杂化，软件变得越来越复杂，软件开发人员越来越满足不了实际需要，软件产品的质量也变得难以满足各方面的要求，加上软件生产率低、软件成本上涨等，出现了一系列严重问题，导致软件危机产生。

软件产业在工业发达国家中占有非常重要的地位。软件危机的产生使人们认识到软件开发必须以新的方法作指导，必须改变原有的软件开发方法。1968 年，北大西洋公约组织在联邦德国召开国际会议，讨论软件危机问题，第一次提出了“软件工程”这个概念。

随着软件技术的发展，软件工程的研究范围和内容也在不断变化和发展。其发展大致经历了传统软件工程、过程软件工程和构件软件工程 3 个阶段。

(1) 传统软件工程阶段。20 世纪 70 年代，为了解决软件项目错误率高以及软件维护任务重等问题，人们提出软件开发工程化的思想，希望使软件开发走上一条规范化的道路，并努力克服软件危机。形成了软件工程的概念、框架、方法和手段。

(2) 过程软件工程阶段。20 世纪 80 年代末逐步发展起来的面向对象方法，形成了完整的面向对象技术体系，适应更大规模、更广泛的应用。这时，进一步提高软件生产率、保证软件质量成为软件工程追求的更高目标。人们认识到，应从软件生命周期的总费用及总价值来决定软件开发方案。在重视发展软件开发技术的同时，提出了软件能力成熟度

模型、个体软件过程和群组软件过程等概念。软件开发过程从目标管理转向过程管理。

(3) 构件软件工程阶段。20 世纪 90 年代以后，软件开发技术的主要处理对象为网络计算和支持多媒体信息的 WWW。为了适应超企业规模、资源共享、群组协同工作，需要开发大量的分布式处理系统。这时软件工程的目的在于不仅提高个人生产率，而且通过支持跨地区、跨部门、跨时空的群组共享信息，协同工作来提高群组的整体生产效率。因整体性软件系统难以更改、难以适应变化，所以提倡基于部件(构件)的开发方法。同时人们认识到计算机软件开发领域的特殊性，不仅要重视软件开发方法和技术的研究，更要重视总结和发展包括软件体系结构、软件设计模式、互操作性、标准化、协议等领域的重用经验。软件重用和软件构件技术正逐步成为主流软件技术。迭代、敏捷、持续集成的理念，正成为业界的共识。

随着软件产业的进步，软件工程必将朝着流水线装配软件工程的方向发展，以顺应软件蓬勃发展的趋势。流水线生产、网络化、服务化与全球化将成为主流，移动互联成为趋势。

## 1.2 软件危机

### 1.2.1 软件的概念、特点及分类

#### 1. 软件的概念

计算机软件无所不在，已成为人们生活中的一部分。长期以来，人们关于软件的许多问题存在一些争议：

在软件开发周期上，为什么需要那么长时间才能结束软件开发？

在软件的成本上，软件的成本为什么如此之高？

在软件的错误上，为什么不能在把软件交给客户之前就发现所有的错误？

在软件的度量上，为什么在软件开发过程中难以度量其进展？

……

以上这些，其实都与软件及其自身特点有着必然的联系。

一般公认的软件定义是：软件是计算机系统中与硬件相互依存的另一部分，它是程序、数据及其相关文档的完整集合。程序是按事先设计的功能和性能要求执行的指令序列，而数据是使程序能正常操纵信息的数据结构，文档则是与程序开发、维护和使用有关的图文材料。这三部分构成了软件，缺一不可。

1983 年，美国电气与电子工程师学会(Institute of Electrical and Electronics Engineers，IEEE)为软件作了如下定义：软件是计算机程序、方法、规则、相关的文档资料以及在计算机上运行程序时所必需的数据。

也有不少人认为，软件除了上述三部分，还应包括知识。软件都是为特定领域服务的，领域知识也是需要软件开发人员掌握的。

### 2. 软件的特点

软件作为一种产品，是一种不同于物质性产品的逻辑性产品，与其他制造类产品有着本质的不同。它除具有一般产品的诸多属性外，还具有自己特有的属性。具体表现在以下几个方面。

1）软件形态的逻辑性

软件是一种逻辑实体，而不是具体的物理实体。因而它具有抽象性、不可见性。可以将软件存储在不同的介质上，但却无法直接看到软件的形态，开发过程的进度也难以衡量，质量难以评价，必须通过观察、分析、判断，利用抽象思维能力去了解软件的功能、性能和其他特性。软件的管理和控制相当困难。

2）生产过程的非制造性

软件的生产过程与硬件不同，在它的开发过程中没有明显的制造过程。

软件是由工程师开发或工程化形成的，而不是传统意义上的制造产生的产品副本。软件成本、质量集中于开发和研制上，这意味着软件项目不能像硬件制造项目那样进行管理。

但软件一经确定，需要批量生产时，相对容易。可以复制产生大量产品副本，即它可以被无限复制成若干副本。

3）使用方式的无磨损性

在软件的运行和使用期间，没有硬件那样的机械磨损、老化问题。软件不会被“用坏”、“用旧”，也不会因为长时间运行而被损耗或产生新的故障，不过它的功能可能会发生退化，需要开发人员不断地更新和维护。随着时间的推移和环境的变化，软件最终可能会被废弃。

此外，当一个硬件构件磨损时，可以用另外一个备用零件替换它，但对于软件则不然。每一个软件故障都表明了设计、编程中存在错误。因此，软件较难维护，维护意味着改正或修改原来的设计。

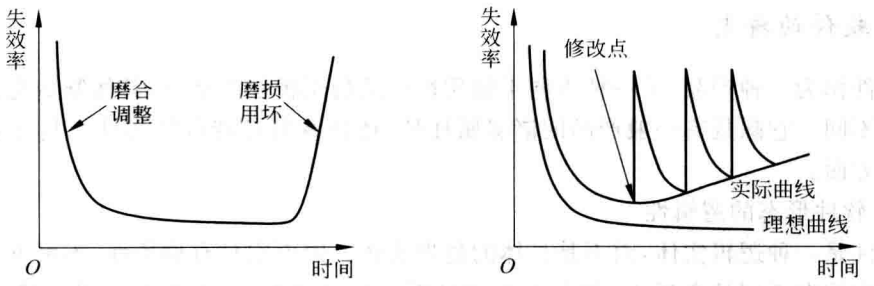
图 1.1 所示为硬件与软件失效率经验曲线，其中硬件的失效率曲线又称 U 形曲线（或浴缸曲线）。而软件一开始需要一段时间磨合期，之后就可以稳定运行，随后逐渐退化。

4）开发和运行对环境的依赖性

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。或者对硬件有一定依赖，或者对特定的操作系统有一定依赖。为了解除或减少这种依赖性，在软件的开发过程中提出了软件的可移植性和跨平台问题。

5）开发方式的手工化

软件的开发至今尚未完全摆脱传统的手工艺开发方式。大多数软件产品都是定制的，而不是通过已有的构件组装而来的。随着面向对象技术、构件技术的发展，情况略有好转。但由于软件开发是一种高强度脑力劳动，开发人员必须利用自己的智力去理解需求，并综合运用软件技术提高开发效率和质量，因此软件开发还无法完全实现自动化。




(a) 硬件失效率曲线　　(b) 软件失效率曲线

图 1.1　硬件与软件失效率经验曲线

6) 软件的复杂性

软件本身是非常复杂的，软件是人类能够创造的最复杂的产物。主要体现在实际问题的复杂性、程序逻辑结构的复杂性、其他领域专门知识的复杂性上。

例如，下述软件所花费的人力资源相当大。

(1) Windows 95 操作系统有 1000 万行，花费上千人。

(2) Windows 2000 操作系统有 5000 万行，花费 5000 多人。

(3) Lotus 1-2-3 V3.0 统计制表软件总计有近 40 万行，花费 263 人·年，成本 2000 多万美元。

(4) WWMCCS(全球军事指挥与控制系统)花费 3500 多人，拖了几年，交付后发现 100 多个错误，最后失败。

因此如果把一个简单程序看作是独唱的话，那么一个较复杂程序就可以看作小合唱，而一个复杂的软件产品则可以看作合唱，甚至大合唱。可见，它们的复杂程度有着显著的区别。

7) 成本的昂贵性

软件成本相当昂贵。在软件开发过程中，会投入大量的、复杂的、高强度的脑力劳动，投入的研发成本相对比较高。

随着计算机硬件技术的不断发展，在一个计算机系统中软件成本的比例在逐步增加，而硬件成本的比例逐步下降，图 1.2 所示为软件成本的经验曲线。据统计，目前软件成本已占到了计算机系统的 90% 以上。

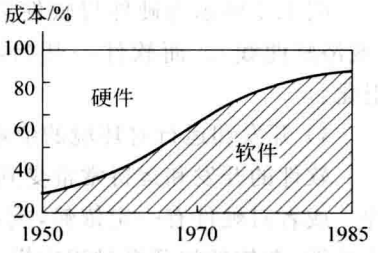



图 1.2　软件成本的经验曲线

8) 社会因素

相当多的软件工作涉及社会因素。软件大大提高了工作的效率，必然释放劳动力，促进机构的重组，由此带来一系列的社会问题。

许多软件的开发和运行涉及组织机构设置、体制运作及日后管理方式等问题，甚至涉及企业文化、使用人员的观念和心理等，这些因素会直接影响软件项目的实际应用效果，甚至软件的成败。

### 3. 软件的分类

软件的分类有许多种方法，可以按软件的功能划分，也可以按软件的工作方式划分，或者按软件的规模划分。此处只介绍按软件的功能进行分类。

按软件的功能划分，软件可以分为系统软件、支撑软件和应用软件。

1）系统软件

系统软件是负责计算机系统中各种独立的硬件部件、相关软件和数据，使得它们可以协调、高效地开展工作的软件。系统软件使得用户和其他软件将计算机当作一个整体而不需要顾及底层每个硬件是如何工作的，如操作系统、数据库管理系统、设备驱动程序、通信处理程序等软件均属于系统软件。

2）支撑软件

支撑软件是为方便用户使用而开发的各种工具软件，是一种通用性较强的软件。例如，支持需求分析、设计、实现、测试和支持管理的工具软件，微软公司的支持各种图表制作的 Visio 软件，SYBASE 公司的数据库设计建模工具 PowerDesigner 软件等均属于支撑软件。

3）应用软件

各行各业都需要软件，但解决的问题不同。为解决特定应用领域问题而开发的软件，称为应用软件，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、医疗制药软件、办公自动化软件、计算机辅助教学软件等均属于应用软件。

## 1.2.2 软件危机

随着软件规模的扩大，软件产品越来越复杂，在软件的开发和维护过程中不可避免地出现了一些问题。例如，在软件运行过程中发现了错误，软件人员必须设法去改正；用户在软件使用过程中有新的需求，软件人员必须相应地修改程序；当计算机系统的硬件或操作系统发生了更新，通常也需要相应地修改软件。上述的软件维护工作，以令人吃惊的比例消耗着资源。更有甚者，有些软件是不可维护的，于是出现了软件危机。软件危机就是在计算机软件开发、维护过程中所遇到的一系列严重问题，导致软件的开发、维护出现风险。

最典型的是美国 IBM 公司在 1963—1966 年开发的 IBM 360 机的操作系统。这一项目花了 5000 人·年的工作量，最多时有 1000 人投入开发工作，写出了近 100 万行源程序。据统计，这个操作系统每次发行的新版本都是从前一版本中找出 1000 个程序错误而修正的结果。

这个项目的负责人 F. D. Brooks 事后总结了他在组织开发过程中的沉痛教训，在撰写的《人月神话》中写道："……正像一只逃亡的野兽落到泥潭中做垂死的挣扎，越是挣扎，陷得越深，最后无法逃脱灭顶的灾难……程序设计工作正像这样一个泥潭……一批批程序员被迫在泥潭中拼命挣扎……谁也没有料到问题竟会陷入这样的困境……"IBM 360 操作系统的历史教训成为软件开发项目的典型事例为人们所记取。