21世纪高等学校规划教材 | 软件工程

面向对象软件工程

石冬凌 等 编著

清华大学出版社

面向对象软件工程

石冬凌 任长宁 贾跃 高兵 编著

清华大学出版社
北京

内 容 简 介

本教材阐述了软件工程的基本思想、软件开发过程、面向对象的分析与设计技术及项目管理的内容。在各章节中以软件生命周期阶段为主线，介绍了软件开发过程中的每个阶段需要达成的任务目标、涉及的基本原理及采用的技术。在每一章中都会使用同一业务背景下的案例带领读者运用讲述的知识进行实践，指导读者灵活解决实际问题。每一章节后面都为读者准备了相应的练习题，帮助读者巩固和加深对知识点的理解。教材的最后一章设置了综合实训环节，将前面讲述的知识进行完整的应用，起到将所学知识融会贯通的作用。

本教材适合高校信息类专业“软件工程”课程的教学，也可作为广大软件开发爱好者的参考资料。

图书在版编目(CIP)数据

面向对象软件工程/石冬凌等编著. —北京：清华大学出版社，2016
21世纪高等学校规划教材·软件工程
ISBN 978-7-302-44888-4

Ⅰ. ①面… Ⅱ. ①石… Ⅲ. ①面向对象语言－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(2016)第209324号

责任编辑：贾　斌　薛　阳
封面设计：傅瑞学
责任校对：梁　毅
责任印制：杨　艳

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：北京国马印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：18.75　　**字　　数**：458千字
版　　次：2016年10月第1版　　**印　　次**：2016年10月第1次印刷
印　　数：1～2000
定　　价：39.80元

产品编号：067279-01

出版说明

随着我国改革开放的进一步深化，高等教育也得到了快速发展，各地高校紧密结合地方经济建设发展需要，科学运用市场调节机制，加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度，通过教育改革合理调整和配置了教育资源，优化了传统学科专业，积极为地方经济建设输送人才，为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是，高等教育质量还需要进一步提高以适应经济社会发展的需要，不少高校的专业设置和结构不尽合理，教师队伍整体素质亟待提高，人才培养模式、教学内容和方法需要进一步转变，学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007 年 1 月，教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》，计划实施“高等学校本科教学质量与教学改革工程（简称‘质量工程’）”，通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容，进一步深化高等学校教学改革，提高人才培养的能力和水平，更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中，各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势，对其特色专业及特色课程（群）加以规划、整理和总结，更新教学内容、改革课程体系，建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上，经教育部相关教学指导委员会专家的指导和建议，清华大学出版社在多个领域精选各高校的特色课程，分别规划出版系列教材，以配合“质量工程”的实施，满足各高校教学质量和教学改革的需要。

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 21 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”。推出的特色精品教材包括：

(1) 21 世纪高等学校规划教材·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) 21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21 世纪高等学校规划教材·信息管理与信息系统。

(6) 21 世纪高等学校规划教材·财经管理与应用。

(7) 21 世纪高等学校规划教材·电子商务。

(8) 21 世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力，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做出了重要贡献。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

联系人：魏江江

E-mail：weijj@tup.tsinghua.edu.cn

目 录

第1章 软件工程概述

1.1 项目导引

为了欢迎刚刚入职的应届毕业生小张，项目组秉承一贯的公司文化，下班后在××酒店聚餐。加入这个新的大家庭，小张一方面感到欣喜，一方面感到紧张和陌生，他急于知道这里不同于学校的一切。

组长老李看出了小张的心思，“小张啊，有什么不懂的，尽管问吧。”小张终于等到了机会，赶紧说道：“我有很多不懂的，比如说，虽然在学校里我们学习了很多开发软件的技能和技巧，可还是很难想象，一个大规模的项目是怎么开发出来的？软件开发中有那么多失败的案例，有的不能满足用户要求，有的超出预算，有的无法控制开发周期，有的后期维护很困难，这些案例失败的原因都是什么呢？软件开发到底是一个怎样的流程呢？最初应该只是用户的要求吧，怎么就能够最终变成功能强大的软件呢？这一定不仅仅是编码就能解决的问题，我觉得这一定很神奇。”项目组同事听到后哈哈大笑，技术顾问老丁说：“孺子可教啊，一连串这么多问题，没错，你说的是一个很现实很复杂的问题，我给你概括一下，就是说软件如何从无到有，如何以团队的形式，在规定的时间及有限的预算内，开发出保证质量，并满足用户需求的软件产品，对吧？”小张点点头。老李接着说：“的确问得好，这就是软件工程要解决的问题，在我们做项目时，软件工程的理论必须贯穿始终。”

1.2 项目分析

那么什么是软件工程呢？软件工程就是告诉人们怎样去开发软件和管理软件。具体地讲，它表现在与软件开发和管理有关的人员和过程上。

从软件项目团队来讲，软件工程的作用在于：在规定的时间内，以预算内的成本，完成预期质量目标（软件的功能、性能和接口达到需求报告标准）的软件。从软件企业本身来讲，软件工程的作用在于：持续地规范软件开发过程和软件管理过程，不断地优化软件组织的个人素质和集体素质，从而逐渐增强软件企业的市场竞争实力。从软件发展进程来讲，软件工程的作用在于：克服软件危机，控制软件进度，节约开发成本，提高软件质量。

简言之，软件工程是研究和应用如何以系统性的、规范化的、可定量的过程化方法去开

发和维护软件，以及如何把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来。

1.3 软件工程的历史

在计算机系统发展早期，软件开发基本上沿用“软件作坊”式的个体化方法，这种方法在软件开发和维护过程中遇到了一系列严重问题：程序质量低下，错误频出，进度延误，费用剧增等，这些问题导致了“软件危机”。1968 年，北大西洋公约组织的计算机科学家在联邦德国召开国际会议讨论软件危机问题，正式提出并使用了“软件工程”这个名词，从此诞生了一门新兴的工程学科。

1. 软件的发展和软件危机

从 20 世纪 40 年代中期世界上第一台计算机出现以后，程序的概念就产生了，在随后的几十年中，计算机软件经历了三个发展阶段，即程序设计阶段（约为 20 世纪 50～60 年代）、程序系统阶段（约为 20 世纪 60～70 年代）和软件工程阶段（约为 20 世纪 70 年代以后），如表 1-1 所示。

表 1-1 计算机软件发展的三个阶段及其特点

阶段 / 描述内容	程序设计	程序系统	软件工程
软件所指内容	程序	程序及说明书	程序、文档及数据
主要程序设计语言	汇编及机器语言	高级语言	软件语言 *
软件工作范围	程序编写	包括设计和测试	包括整个软件生存周期
需求者	程序设计者本人	少数用户	市场用户
开发软件的组织	个人	开发小组	开发小组及大、中型软件开发机械
软件规模	小型	中、小型	大、中、小型
决定质量的因素	个人程序设计技术	小组技术水平	管理水平
开发技术和手段	子程序、程序库	结构化程序设计	数据库、开发工具、工程化开发方法、标准和规范、网络和分布式开发、面向对象技术、软件过程与过程改进
维护责任者	程序设计者	开发小组	专职维护人员
硬件特征	价格高、存储容量小、工作可靠性差	降价、速度、存储容量及工作可靠性有明显提高	向超高速、大容量、微型化及网络化方向发展
软件特征	完全不受重视	软件技术的发展不能满足需求，出现软件危机	开发技术有进步，但未获突破性进展，价格高，未完全摆脱软件危机

* 软件语言包括需求定义语言、软件功能语言、软件设计语言、程序设计语言等。

(1) 软件发展最根本的变化体现在以下几个方面。

① 人们改变了对软件的看法。早在 20 世纪 50～60 年代，程序设计曾经被看做是一种自由发挥创造才能的技术领域。当时人们认为，只要能在计算机上得出正确的结果，程序的

写法可以不受任何约束。随着计算机的使用日趋广泛，人们不断提出更高的要求（例如，要求程序易懂、易用、易于修改和扩充），于是程序便从按个人意图创造的“艺术品”转变为能被广大用户接受的工程化产品。

② 需求是软件发展的动力。早期为了满足自己的需要，程序开发者不拘风格地自由创作。这种自给自足的生产方式是其初级阶段的表现。进入软件工程阶段后，软件开发的成果具有社会属性，它要在市场中流通以满足广大用户的需要。

③ 软件工作的考虑范围从只顾程序的编写扩展到涉及整个软件生命周期。

④ 随着计算机硬件技术的进步，要求软件能与之相适应。这个时期出现了“软件作坊”，它基本上仍然沿用早期的个体化软件开发方法。缺乏统一的管理和协调导致许多开发项目由于软件质量问题造成巨大损失，同时随着产品的增加，软件开发力量不得不全部投入维护，没有能力继续开发新的应用系统，这就造成了计算机应用进一步发展的停滞，即20世纪60年代的“软件危机”现象。软件危机是指计算机软件在开发和维护过程中所遇到的一系列严重问题，主要有以下一些表现形式。

- 软件代价高。随着软件产业的发展，软件成本日益增长，而计算机硬件随着技术的进步、生产规模的扩大，价格却不断下降，造成了软件代价在计算机系统中所占的比例越来越大。20世纪50年代，软件成本在整个计算机系统中所占的比例不大，为10%～20%；到20世纪60年代中期已经增长到50%左右；20世纪70年代以后，软件代价高的问题不仅没有解决，反而进一步加深了。图1-1大体表示了一个计算机系统中，硬件和软件所占费用的比例。

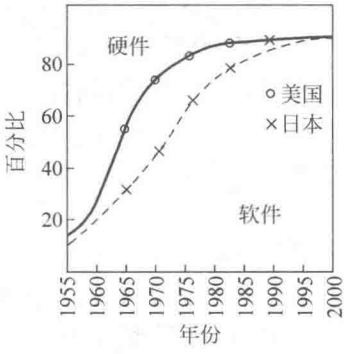

图1-1　软件与硬件费用之比

- 开发进度难以控制。软件是一种逻辑的系统元素。为了完成一个复杂的软件，常要建立庞大的逻辑体系。同样的算法可以由差别甚大的不同程序形式来实现，在研究大型系统时遇到的困难也是越来越多，软件的开发过程很难加以控制。
- 工作量估计困难。通常要根据任务的复杂性、工作量及进度要求来安排人力，但这种工作量估算方式仅对各部分工作独立互不干扰的工作适用，而软件系统整体各部分之间存在的任务合作与交流活动也会增加工作量。由于软件系统结构复杂，各部分之间的附加联系极大，在拖延的软件项目上增加人力通常只会使其更难按期完成。这对于一般的工业产品来说是难以想象的。
- 质量差。软件的产品质量与其他商品的质量问题有着很大的不同。使用“软件作坊”开发软件的方法沿袭了早期形成的个体化方式，软件（程序）开发过程没有交互性，软件的规划、设计、测试和维护都只能由某一个人全部负责，只有程序清单而没有任何正式的软件规划文档。这就使得软件修改和维护十分困难，有时甚至变得不可能。此外，软件规模和数量的急剧增加，用户需求的不断变化，使得软件的质量控制成为一个很难解决的问题，这是由于软件所处的特殊地位造成的。

- 修改、维护困难。当软件系统变得庞大、问题变得复杂时，常常还会发生“纠正一个错误却带来更多新错误”的问题。此外，人们习惯于认为软件易于修改、容易扩充，因此在系统投入运行后为适应新环境，经常提出要求进行维护。这样产生的维护工作量将难于估算。根据 1999 年美国的 Standish Group 对当年美国的软件项目的统计数字表明(如图 1-2 所示)，只有 26%的软件项目是真正成功的，其余的项目全部是失败的或是有问题的，28%的项目是完全失败的。这些存在问题的或是失败的项目带来的直接损失是 870 亿美金，占美国当年全部 IT 投资(2550 亿美金)的近 40%，而由于这些项目所带来的间接损失是无法估量的，在全部这些项目中，平均超期 189%，平均超预算 222%，平均 27 个月滞后于最终用户的需求，更有 80%的资源被花费在对应用的维护上。

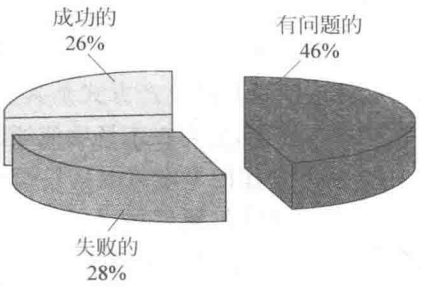

图 1-2 Standish Group 小组报告

总之，在软件整个生命周期中，错误发现得越晚，纠正错误所要付出的代价也就越大。其原因是：测试日趋复杂，修改的文件和文本需分发的范围更广，多次反复以前的测试，问题和修改信息传递的范围更大，涉及的人员更多。

有资料表明，在数据处理方面，30%～80%的预算往往用于软件维护工作，这就直接造成了软件成本的提高。

(2) 产生软件危机的根本原因是软件面临的问题空间的复杂性。

软件的应用领域很广，面临的问题很复杂，所以涉及的处理技术也十分广泛，包括信息技术、网络技术、人机界面技术、人机会话环境技术等。另外，面临的问题空间往往还牵涉到管理体制、组织机构、内外部环境、用户水平、经济学、心理学等许多非技术问题。问题空间的复杂性决定了软件系统的复杂性。

产生软件危机的另一个重要原因是计算机硬件体系结构的发展速度滞后于软件应用面拓展速度。时至今日，硬件的体系结构基本未变。从 5 大组成部件来看，出现了图形扫描仪、光笔、绘图机等许多新式输入输出设备，多 CPU 的计算机在实时系统中得到应用，内外存的容量和存取速度有很大提高，但是，这些部件的变化都只是硬件功能的完善、性能的提高，属于改良性质的变化。时至今日，计算机的硬件体系结构仍属于冯·诺依曼计算机，它的基本特征是：顺序地执行程序指令，按地址访问线性的存储空间，数据和指令在机内采用统一的表示形式，只能完成四则运算和一部分逻辑运算。冯氏计算机的初衷是为数值计算服务的，然而随着计算机应用领域的扩大，所面临的问题 90%以上是非数值计算。为了满足用户的需求，或在逻辑上构建许多的软件层次，每一软件层次都可以看作是一种语言的翻译器或解释器，用这种方法来填补用户和裸机之间的鸿沟。简单地说，就是把解题过程分解成一系列能由冯氏计算机处理的四则运算和逻辑运算，这就使软件非常庞大，开发工作十分困难，软件的可靠性和可维护性很差。因此，可以说，正是因为把以科学计算为基础的冯氏计算机应用在非数值计算的数据处理中(会计信息系统属于此类处理)，所以也把危机转嫁在软件上。

软件危机的产生，除了上述两个主要原因之外，还与人们在软件开发和维护中采用错误

的方法有关。

软件系统的复杂性虽然给开发和维护带来了客观困难，但是人们在开发和使用计算机系统的长期实践中，也积累和总结了许多经验，如果坚持不懈地使用经过实践证明是正确的方法，许多困难是完全可以克服的。但是，目前相当多的开发人员对软件开发和维护还有不少糊涂的观念，在实践中或多或少地采用错误的技术和方法，如忽视软件的维护性等。这些关于软件开发和维护的错误认识和做法是产生软件危机的第三个重要原因。

此外，造成软件危机还有如下一些原因。

① 用户需求不明确，体现在4个方面：软件开发出来之前，用户自己不清楚其具体需求；用户对软件需求的描述不精确(有遗漏或者二义性)甚至有错误；软件开发过程中用户不停地提出修改要求(例如修改软件功能、界面、支撑环境等)；软件开发人员对用户的理解与用户本来愿望有差异。

② 缺乏正确的理论指导，特别是缺乏有力的方法学和工具方面的支持。

③ 软件规模越来越大。

④ 软件复杂度越来越高。

⑤ 软件灵活性要求高。

影响软件生产率与质量的因素十分复杂，包括个人能力、团队联系、产品复杂度、合适的符号表达方式，可利用的时间地点以及其他因素(诸如技术水平、变更控制、采用的方法、所需要的可靠性、对问题的理解、需求稳定程度、设施及资源、相应的培训、管理水平、恰当的目标、期望的高低等)。

2. 软件工程的诞生

1968年，北大西洋公约组织的计算机科学家在联邦德国召开的国际学术会议上，讨论和制定摆脱“软件危机”的对策。同时，第一次提出了软件工程(Software Engineering)这个概念，从此一门新兴的工程学科——软件工程学——为研究和克服软件危机应运而生。

“软件工程”的概念是为了有效地控制软件危机的发生而被提出来的，它的中心目标就是把软件作为一种物理的工业产品来开发，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”。软件工程是一门旨在开发满足用户需求、及时交付、不超过预算和无故障的软件的学科。软件工程的主要对象是大型软件。它的最终目的是摆脱手工生产软件的状况，逐步实现软件开发和维护的自动化。

我们要求工程目标能在一定的时间、一定的预算之内完成。软件工程是针对软件危机提出来的。从微观上看，软件危机的特征正是表现在完工日期一再拖后、经费一再超支，甚至工程最终宣告失败等方面。而从宏观上看，软件危机的实质是软件产品的供应赶不上需求的增长。

自从软件工程概念提出以来，经过几十多年的研究与实践，虽然“软件危机”没有得到彻底解决，但在软件开发方法和技术方面已经有了很大的进步。尤其应该指出的是，自20世纪80年代中期，美国工业界和政府部门开始认识到，在软件开发中，最关键的问题是软件开发组织不能很好地定义和管理其软件过程，从而使一些好的开发方法和技术都起不到所期望的作用。也就是说，在没有很好定义和管理软件过程的软件开发中，开发组织不可能在好的软件方法和工具中获益。