“十二五”普通高等教育本科国家级规划教材

★ 高等学校软件工程系列教材

软件设计与体系结构

（第2版）

Software Design and Architecture

(Second Edition)

董威　文艳军　陈振邦　编著

高等教育出版社

“十二[redacted]家级规划教材

★ 高等学校软件工程系列教材

软件设计与体系结构

（第2版）

Software Design and Architecture

(Second Edition)

董威　文艳军　陈振邦　编著

高等教育出版社·北京

内容提要

本书是“十二五”普通高等教育本科国家级规划教材，对软件设计及体系结构的相关思想、理论与方法进行了系统的介绍。主要内容包括软件工程与软件设计、统一建模语言 UML、软件设计基础、面向对象的软件设计方法、面向数据流的软件设计方法、用户界面设计、软件体系结构风格与设计模式、基于分布构件的软件体系结构、软件体系结构评估、软件设计的进化等。

本书包含了作者多年来在软件开发实践、软件工程教学和科研活动中的认识与体会，并融入大量的案例分析，力求全书内容与组织结构的系统性、先进性、基础性和实用性。为方便读者学习，本书还配有重要知识点讲解视频等教学资源。

本书可作为高等学校计算机科学与技术、软件工程等专业的教材，以培养学生的软件设计思维能力以及方法和技术的运用能力，同时也可供开发人员和项目管理人员在软件开发实践中参考。

图书在版编目(CIP)数据

软件设计与体系结构/董威，文艳军，陈振邦编著
.--2 版.--北京：高等教育出版社，2017.12
ISBN 978-7-04-048630-8

Ⅰ. ①软… Ⅱ. ①董… ②文… ③陈… Ⅲ. ①软件设计-高等学校-教材 Ⅳ. ①TP311.5

中国版本图书馆 CIP 数据核字(2017)第 243308 号

策划编辑 倪文慧	责任编辑 倪文慧	封面设计 于文燕	版式设计 杜微言
插图绘制 于 博	责任校对 吕红颖	责任印制 赵义民	

出版发行	高等教育出版社	网 址	http://www.hep.edu.cn
社 址	北京市西城区德外大街 4 号		http://www.hep.com.cn
邮政编码	100120	网上订购	http://www.hepmall.com.cn
印 刷	大厂益利印刷有限公司		http://www.hepmall.com
开 本	787mm×1092mm 1/16		http://www.hepmall.cn
印 张	21.75	版 次	2010 年 2 月第 1 版
字 数	420 千字		2017 年12月第 2 版
购书热线	010-58581118	印 次	2017 年12月第 1 次印刷
咨询电话	400-810-0598	定 价	42.00 元

物 料 号 48630-00

作者简介：

董威，国防科技大学计算机学院教授、博士生导师，现从事高可信软件工程技术、软件智能化开发等方向的研究工作，中国计算机学会软件工程专委会、形式化方法专委会委员，入选教育部新世纪优秀人才支持计划，曾获东软—NASAC青年软件创新奖、霍英东基金会高校青年教师奖、军队院校育才银奖。主讲本科生和研究生的软件工程、需求工程、软件测试与验证、软件质量保证等多门课程。先后主持国家自然科学基金课题5项、国家863项目1项、国防预研和预研基金重点项目3项。注重研究成果与重大型号工程相结合，主持完成高可信软件技术在航空航天、交通控制、国产自主可控系统等关键领域中的应用研究课题十余项。出版国家级规划教材两部，在国内外学术刊物和会议上发表论文70余篇。

文艳军，博士，国防科技大学计算机学院副教授、硕士生导师。主要从事操作系统、软件体系结构与设计等课程的教学工作和可信软件设计、程序静态分析、软件体系结构等方面的科研工作。在国际会议和学术期刊上发表学术论文十余篇。作为负责人承担国家自然科学基金项目或子项目3项、921载人航天研究项目1项。获军队科技进步二等奖2项、三等奖1项。

陆振邦，博士，国防科技大学计算机学院副研究员、硕士生导师。目前主要从事程序分析、形式化方法、面向服务与面向构件的软件工程方面的研究。作为项目负责人或主要成员承担了多项国家自然科学基金课题及国家或国防重点研发课题。在国内外软件工程或软件理论刊物、会议上发表论文20余篇，获专利和软件著作权各1项，出版教材1部。获国家科技进步二等奖1项。

前　言

随着计算机技术、微电子技术、网络技术和多媒体技术的迅速发展和广泛应用，今天的社会进入了以计算机为核心的信息社会。软件被认为是信息化的灵魂，已被用于政治、经济、文化、科技、教育、国防、生活等各个领域。随着软件在社会中的地位和作用越来越显著，人们对软件的质量、成本和开发周期等方面提出的要求也越来越高。软件工程强调以工程化思想和方法开发软件，而软件设计作为软件开发过程中的核心活动之一，对开发出满足需要的高质量软件起关键作用。软件设计在软件工程中所体现的重要性包括：软件设计是对软件需求的直接体现；软件设计为软件实现提供直接依据；软件设计将综合考虑软件系统的各种约束条件并给出相应方案；软件设计的质量很大程度上将决定最终软件系统的质量；及早发现软件设计中存在的错误将极大地减少软件修复和维护所需的成本；等等。在软件工程的发展过程中，出现了大量与软件设计相关的方法与技术，而软件体系结构作为软件设计过程中控制软件复杂性、提高软件系统质量、支持软件开发和复用的重要手段之一，自提出以来日益受到软件研究者和实践者的关注。

在当前我国软件产业的发展过程中，基础扎实、知识全面、经验丰富的高水平软件设计人员仍然非常缺乏，这成为我国软件产业发展的制约因素。在当前软件工程专业方向的教育中，把软件设计和软件体系结构相关理论与方法作为单独一门课程进行系统地讲述，对培养大量所需的高水平软件设计人员至关重要。而在教育部高等学校计算机科学与技术教学指导委员会制定的《软件工程专业规范》中，“软件设计与体系结构”已经作为一门核心课程单独列出，并有相应的知识单元和知识点，但亟需相应的教材以便于高等院校教学实施。针对以上背景和需求，本书对软件设计以及软件体系结构的相关思想、理论与方法进行了系统的介绍，包括软件设计与软件体系结构在软件工程中的地位和作用、软件设计的基本方法与原则、统一建模语言 UML 2.0、面向对象的软件设计方法、面向数据流的软件设计方法、人机界面设计、软件体系结构风格与设计模式、基于构件的软件体系结构、软件体系结构评估、软件设计的进化等内容。本书包含了作者多年来在软件开发实践、软件工程教学和科研活动中的认识与体会，融入大量的案例分析，力求全书内容与组织结构的系统性、先进性、基础性和实用性。

本书第二版在第一版的基础上，增加了对软件工程知识体 SWEBOK 和计算教程软

件工程卷 CCSE 中软件设计相关知识单元的描述，对设计模式、体系结构变更等内容进行了扩展，并对习题进行了更新。此外，一个重要的扩充是对文中的一些重要知识点给出了授课视频进行深入讲解。

在教学计划中，建议对第 1～10 章采用自然顺序讲授。如果课时有限，对于本科生，第 9、10 章可进行略讲或略去，第 5 章也可以根据情况省略，其他各章按照顺序讲述。本书内容曾多次在国防科技大学计算机专业、软件工程专业本科生和硕士研究生的相关课程教学中讲授。

本书由董威教授组织编写，其中第 1、2、3、4、5、10 章由董威教授撰写，第 7、8 章由文艳军副教授撰写，第 6、9 章由陈振邦副教授撰写。书中借鉴和参考了一些中外图书文献，在此谨向有关作者表示诚挚的谢意。

最后，诚恳欢迎读者和专家对本书提出批评意见和改进建议。

作　者

2017 年 12 月

目　录

第1章　软件工程与软件设计

随着微电子技术、计算机技术、网络技术和多媒体技术的迅速发展和广泛应用，今天的社会已进入了以计算机为核心的信息社会，计算机已广泛应用于航空航天、工业控制、办公自动化、商业信息处理、家用电器等领域，成为人们工作和生活中不可缺少的工具。软件被认为是信息化的灵魂，它在硬件的支持下，肩负着信息采集、存储、处理、加工、传输、显示、人机交互、控制、应用等任务。目前，形形色色的软件被用于政治、经济、文化、科技、教育、军事、生活的各个领域，人们对软件的依赖越来越紧密。由于软件在社会中的地位和作用越来越显著，人们对软件的功能、质量、成本和开发周期等方面提出的要求也越来越高。

然而，软件的功能越强、使用越方便，其规模和复杂程度就越高，如果软件开发水平跟不上，就会大量出现软件开发计划一拖再拖、成本失去控制、软件质量得不到保证等现象。因此，数十年来，人们十分重视软件开发方法、工具和环境的研究，并在这些领域取得了重要的成果，使得软件工程取得了长足的进步，并逐渐成熟。软件工程强调以工程化思想和方法开发软件，而软件设计作为软件开发过程中的核心活动之一，对开发满足需要的高质量软件起到关键作用。本章首先从全局的观点对软件工程的概念、软件生存周期、开发过程模型进行概述，然后阐述软件设计在软件工程和软件开发中的地位、作用及其相关特征和组成要素。在软件工程的发展过程中，出现了大量与软件设计相关的方法与技术，而软件体系结构作为软件设计过程中控制软件复杂性、提高软件系统质量、支持软件开发和复用的重要手段之一，自提出以来日益受到软件研究者和实践者的关注。另外，本章对软件体系结构的主要概念、发展过程和所关注的内容也进行了概要描述。

1.1　软 件 工 程

软件工程是计算机软件发展到一定阶段后，为了应对软件危机而出现的。本节首先对软件的定义和特点进行简要介绍，然后对软件危机、软件工程的发展、相关概念、目标与原则等方面进行回顾与总结。

1.1.1　软件概述

计算机软件是与计算机系统操作有关的程序、规程、规则及任何与之有关的文档及数据[1]，即：

计算机软件=程序+数据+文档

软件由两部分组成：其一是机器可执行的程序及有关数据；其二是机器不可执行的，与软件开发、运行、维护、使用、培训有关的文档。程序(program)是用程序设计语言描述的、适合计算机处理的语句序列，它是软件开发人员根据用户需求开发出来的。程序设计语言编译器可以将程序翻译成机器可执行的指令，这组指令亦称机器语言程序，它将根据用户的需求控制计算机硬件的运行，处理用户提供或机器运行过程中产生的各类数据并输出结果。曾经出现过的程序设计语言有几百种，但广泛使用的高级语言不过十余种，例如，用于科学计算的 FORTRAN 语言；用于事务处理的 COBOL 语言；支持结构化程序设计的 Pascal、C、Ada 语言；支持面向对象程序设计的 C++、Java 语言等。此外，还有一些专用的高级语言，例如用于数据库查询的 SQL 语言。文档(document)是一种数据媒体和其上所记录的数据。文档记录软件开发的活动和阶段成果，它具有永久性并能供人或机器阅读。它不仅可以用于专业人员和用户之间的通信和交流，还可以用于软件开发过程中管理和运行阶段的维护。在现代软件工程中，文档起着非常重要的作用。

软件是逻辑产品而不是物理产品，因此软件在开发、生产、维护和使用等方面与硬件相比均存在明显的差异。软件开发与硬件开发相比，更依赖于开发人员的业务水平、智力、人员的组织、合作和管理。在大多数场合，软件的开发、设计几乎都是从头开始的，开发的成本和进度很难估计。软件在提交使用以前，尽管经过了严格的测试和试用，但仍不能保证软件没有潜在的错误。软件开发成功之后，一般只需对原版软件进行复制即可使用。但是，软件在使用过程中的维护工作却比硬件复杂得多，包括对错误进行修改的“纠错性维护”、对软件性能和功能进行完善和改进的“完善性维护”、运行环境发生变化后的“适应性维护”等。由于软件内部的逻辑关系复杂，软件在维护过程中还可能产生新的错误，因此，软件产品在使用过程中的维护工作远比硬件产品的维护复杂。

经过数十年的发展，计算机软件已经广泛应用于各个领域。下面根据软件应用的目标和特点简单地介绍计算机软件的应用领域和类型。

(1) 系统软件

计算机系统软件是计算机管理自身资源(如 CPU、存储器、外部设备等)、提高计算机的使用效率并为计算机用户提供各种服务的基础软件。系统软件要为各类用户提供

尽可能标准、方便的服务，尽量隐藏计算机系统的某些低级特征或实现细节。系统软件包括操作系统、编译器、数据库管理系统、系统检查与诊断软件等。

（2）实时软件

监视、分析和控制现实世界发生的事件，能以足够快的速度对输入信息进行处理并在规定的时间内做出反应的软件，称为实时软件。实时软件依赖于处理机系统的物理特性，如计算速度和精度、I/O 信息处理与中断响应方式、数据传输效率等。支持实时软件的操作系统称为实时操作系统。实时软件必须有很高的可靠性和安全性。

（3）嵌入式软件

嵌入式计算机系统将计算机嵌入某一系统之中，使之成为该系统的重要组成部分，控制该系统的运行，进而实现一个特定的物理过程。用于嵌入式计算机系统的软件称为嵌入式软件。嵌入式计算机系统一般都要和各种仪器、仪表、传感器连接在一起，因此，嵌入式软件一般需要具有实时地采集、处理、输出数据的能力。

（4）科学和工程计算软件

科学和工程计算软件以数值算法为基础，对数值量进行处理和计算，主要用于科学和工程计算，例如数值天气预报、计算机系统仿真、计算机辅助设计（CAD）等。从 20 世纪 50 年代起，有经验的程序员就用程序设计语言把许多常用算法编制成标准程序，如今已经积累了大量的科学和工程计算软件，为计算机在科学和工程上的应用做出了重要贡献。

（5）事务处理软件

事务处理软件是用于处理事务信息，特别是商务信息的计算机软件。事务信息处理是软件最大的应用领域，它已由初期零散的、小规模的软件系统，如工资管理系统等发展成为管理信息系统（MIS），如世界范围内的飞机订票系统等。事务处理软件需要访问、查询、存放有关事务信息的一个或几个数据库，经常按某种方式和要求重构存放在数据库中的数据，能有效按照一定要求和格式生成各种报表。它们往往具有良好的人机界面，在大多数场合采用交互工作方式。

（6）人工智能软件

人工智能软件是支持计算机系统产生类似人类某些智能的软件。它们不是用传统的计算或分析方法求解复杂问题，而是采用诸如基于规则的演绎推理技术和算法，在很多场合还需要知识库的支持。迄今为止，在专家系统、模式识别、自然语言理解、人工神经网络、自动程序设计、机器人学等领域开发了许多人工智能应用软件，用于医疗诊断、图像和语音自动识别、语言翻译等。

（7）个人计算机软件

个人计算机上使用的软件也可包括系统软件和应用软件两类。在个人计算机上已

出现大量的文字和表格处理软件、图形和多媒体信息处理软件、个人和商业上的财务处理软件、网络软件等，并采用多窗口、多媒体等技术，使个人计算机具有用文字、图形、图像、声音进行人机交互的能力，为个人计算机的普及创造了必要条件。目前，个人计算机普遍与计算机网络相结合，使得通信、网络资源共享等更加方便，加速了人类社会信息化的进程。

1.1.2　软件危机

20世纪60年代末期开始，“软件危机”一词在计算机界广为流传。事实上，软件危机几乎从计算机诞生之日就出现了。当时训练有素的程序员大多数还处于“技艺式”的工作状态，他们不用框图和注释就能够熟练编写出几百行程序代码并很快在机器上调试通过。他们熟悉程序的全部操作和结构，为了减少计算机的时/空开销，他们创造并使用了许多令人惊叹的技巧。然而，客观上，工业、商业、科学技术和国防等部门对软件功能的需求很高，软件规模很大，往往有几万、几十万甚至几百万行代码。在当时的条件下，要完成这样一个系统，在一定的时间内依靠一个人或几个人的智力和体力是无法实现的。由于软件是逻辑和智力产品，盲目增加软件开发人员并不能成比例地提高软件开发效率。相反，随着人员数量的增加，人员的组织、协调、通信、培训和管理等方面出现的问题将更为严重。人们在大型软件项目开发面前显得力不从心，一些公司或团体承担的大型软件开发项目经常出现预算超支、软件交货时间延迟、软件质量差、维护困难、在软件维护过程中很容易产生新的错误、软件的可移植性差、软件很少能够复用等问题，工业界为维护软件支付的费用甚至占全部硬件和软件费用的40%~75%。许多重要的大型软件开发项目在耗费了大量的人力和财力之后，由于离预定目标相差甚远不得不宣告失败。这些都使得软件危机达到了令人难以容忍的地步。

从软件危机的种种表现和软件作为逻辑产品的特殊性，可以发现产生软件危机的原因有以下几点：

① 用户对软件需求的描述不精确，可能存在遗漏、二义性、错误等。在软件开发过程中，用户甚至还提出修改软件功能、界面、支撑环境等方面的要求，导致需求不断变化。

② 软件开发人员对用户需求的理解与用户的期望有差异，这种差异必然导致开发出来的软件产品与用户要求不一致。

③ 大型软件项目需要组织一定的人力共同完成，但多数管理人员缺乏开发大型软件系统的经验，而多数软件开发人员又缺乏管理方面的经验。各类人员的信息交流不及时、不准确，有时还会产生误解。

④ 软件项目开发人员不能有效、独立自主地处理大型软件的全部关系和各个分支,因此容易产生疏漏和错误。

⑤ 缺乏有力的方法学和工具方面的支持,过分依靠程序设计人员在软件开发过程中的技巧和创造性,加剧了软件产品的个性化。

⑥ 软件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"。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的开发效率和能力,新的、更大且更复杂的问题又出现在人们面前。

在认真分析了产生软件危机的原因之后,人们开始用工程方法探索软件生产的可能性,即用现代工程的概念、原理、技术和方法进行计算机软件的开发、管理、维护和更新。于是,计算机科学技术的一个新领域——"软件工程"诞生了。多年来,软件工程的研究与应用已经取得了很大成就,包括软件开发方法、工具、管理等多个方面,大大缓解了软件危机造成的被动局面。

1.1.3 软件工程的概念

1968 年,在德国 Garmish 召开的 NATO(北大西洋公约组织)计算机科学会议上首先提出了"软件工程"的概念,试图建立并使用正确的工程方法开发出低成本、高可靠性并能高效运行的软件,从而解决或缓解软件危机。

软件工程的定义有不同的表述方式,典型的定义包括:

① 软件工程是将系统的、规范的、可度量的方法应用于软件的开发、运行和维护过程,以及对上述方法的研究。

② 软件工程是用工程、科学和数学的原则与方法,研制、维护计算机软件的有关技术及管理方法。

一般认为,软件工程由方法、工具和过程三个要素组成,如图 1-1 所示。在三个要素中,方法支撑过程和工具,而过程和工具促进方法学的研究。

软件工程方法是完成软件工程项目的技术手段,它支持项目计划和估算、系统和软件需求分析、软件设计、编码、测试和维护等活动。软件工程中使用的软件工具是人类在软件开发活动中智力和体力的扩展和延伸,它自动或半自动地支持软件的开发和管理,支持各种软件文档的生成。软件工具最初是零散的,不系统、不配套,后来根据不同类型软件项目的要求建立了各种软件工具箱或集成环境,支持软件开发的全过程。软件工程中的过程贯穿于软件开发的各个环节。管理者在软件工程的过程中,要对软件开发的质量、进度、成本进行评估、管理和控制,包括人员组织、计划跟踪与控制、成本估算、质量保证、配置管理等。

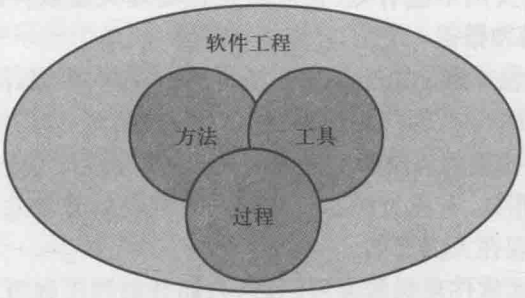

图1-1　软件工程的组成要素

1.1.4　软件工程的目标与原则

软件工程的目标是:在给定成本、进度的前提下,开发出具有可修改性、有效性、可靠性、可理解性、可维护性、可复用性、可适应性、可移植性和可追踪性,并满足用户需求的软件产品。追求这些目标有助于提高软件产品的质量和开发效率,减少维护的困难[2]。

(1) 可修改性(modifiability)

可修改性是指允许对系统进行修改而不增加原系统的复杂性。它支持软件的调试与维护,但度量起来比较困难。

(2) 有效性(efficiency)

有效性是指软件系统能最有效地利用计算机的时间资源和空间资源,一般将系统的时/空开销作为衡量软件质量的一项重要技术指标。很多场合,在追求时间有效性和空间有效性方面会发生矛盾,这时不得不牺牲时间效率换取空间有效性或牺牲空间效率换取时间有效性,因此时/空折中是经常出现的。

(3) 可靠性(reliability)

可靠性是指软件在给定环境和时间下不发生故障的概率。对于实时嵌入式计算机系统,可靠性是一个非常重要的目标,因为软件要实时地控制一个物理过程,如宇宙飞船的导航、核电站的运行等,如果可靠性得不到保证,一旦出现问题可能会导致灾难性的结果,后果不堪设想。

(4) 可理解性(understandability)

可理解性是指系统具有清晰的结构,能直接反映问题的需求。可理解性有助于控制软件系统的复杂性,并支持软件的维护、移植或复用。

(5) 可维护性(maintainability)

可维护性是指软件产品交付用户使用后能够方便地对它进行修改,以改正潜在的错误以及改进性能和其他属性,使软件产品适应环境的变化等。软件的可理解性和可