高等学校计算机类专业
重点规划教材

软件工程

Software Engineering

张秋余　张聚礼　柯　铭
张　红　马　威　编著

西安电子科技大学出版社
http://www.xduph.com

高等学校计算机类专业重点规划教材

软 件 工 程

张秋余　张聚礼　柯　铭
张　红　马　威　编著

西安电子科技大学出版社

内 容 简 介

本书结合软件产业现状，较为全面地介绍了软件工程的基本概念、原理和方法，以期培养学生在理论及应用上的系统整合能力以及从系统的角度看待整个软件项目的能力。

本书共 14 章。第 1～6 章主要介绍了软件工程学与传统软件工程方法的基本理论，主要包括软件工程的策划、分析、设计、实现、测试和维护等工作；第 7～13 章主要介绍了面向对象软件工程，结合软件统一过程模型，从面向对象范型出发对软件工程进行重新演绎，全面、系统、清晰地介绍了面向对象软件工程的基本概念、原理、方法和工具，并将考勤系统完整实例贯穿于面向对象软件开发的整个过程；第 14 章介绍了当今比较流行的现代软件工程软件开发方法。

本书可以作为计算机科学与技术、软件工程相关专业本科生或研究生的教材，也可以作为软件工程领域专业人士的参考书籍。

图书在版编目(CIP)数据

软件工程/张秋余等编著. —西安：西安电子科技大学出版社，2014.12

高等学校计算机类专业重点规划教材

ISBN 978-7-5606-3510-1

Ⅰ. ① 软… Ⅱ. ① 张… Ⅲ. ① 软件工程—高等学校—教材 Ⅳ. ① TP311.5

中国版本图书馆 CIP 数据核字(2014)第 217570 号

策　　划　王　飞

责任编辑　马武装　董柏娴

出版发行　西安电子科技大学出版社（西安市太白南路 2 号）

电　　话　(029)88242885　88201467　邮　　编　710071

网　　址　www.xduph.com　电子邮箱　xdupfxb001@163.com

经　　销　新华书店

印刷单位　陕西天意印务有限责任公司

版　　次　2014 年 12 月第 1 版　2014 年 12 月第 1 次印刷

开　　本　787 毫米×1092 毫米　1/16　印张 26

字　　数　618 千字

印　　数　1～3000 册

定　　价　45.00 元

ISBN 978 – 7 – 5606 – 3510 – 1/TP

XDUP 3802001-1

*** * * 如有印装问题可调换 * * ***

前　言

在现代社会中，软件应用于多个方面。典型的软件有电子邮件、嵌入式系统、人机界面、办公套件、操作系统、编译器、数据库、游戏等。可以说各个行业几乎都有计算机软件的应用，如工业、农业、银行、航空、政府部门等。各种软件的应用促进了经济和社会的发展，提高了工作和生活效率。

软件工程是一门研究用工程化方法构建和维护有效、实用和高质量软件的学科，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。通过软件工程课程的学习，可以使学生系统地掌握软件开发理论、技术和方法，使用正确的软件工程方法，开发出成本低、可靠性好并在机器上能高效运行的软件，同时可了解软件工程各领域的最新发展动向，为今后从事软件开发和维护打下坚实的基础——帮助读者从程序员上升为系统设计师，实现“员”到“师”的质的变化。

本书内容丰富，组织结构严谨，原理和方法相结合，丰富的图表与实例应用相结合，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，适合作为高校本科计算机科学与技术、软件工程等相关专业本科生与研究生的教材，也可作为软件开发人员、科研人员以及有关大专院校师生的参考书籍。

本书共 14 章。第 1～6 章介绍软件工程学与传统软件工程方法的基本理论，主要包括软件工程的策划、分析、设计、实现、测试和维护等工作；第 7～13 章介绍面向对象软件工程，结合软件统一过程模型，从面向对象范型出发对软件工程进行重新演绎，全面、系统、清晰地介绍了面向对象软件工程的基本概念、原理、方法和工具，并将考勤系统完整实例贯穿于面向对象软件开发的整个过程，进行案例分析和开发心得介绍；第 14 章介绍当今比较流行的现代软件工程软件开发方法，主要包含领域工程、敏捷软件开发、极限编程、测试驱动开发、模型驱动软件开发、面向方面、面向 Agent、净室软件工程等开发方法。附件给出了软件工程师职业素质及道德规范。

本书第 1、11、12 章由张红老师编写，第 2、3、4、6 章由柯铭老师编写，第 5 章由马威老师编写，第 7、8、9、10、13 章由张聚礼老师编写，第 14 章由张秋余老师编写。全书由张秋余老师统稿并审定。

在本书编写过程中，得到了兰州理工大学计算机与通信学院领导的支持与鼓励，也得到了谢鹏寿老师的校订和大力协助，西安电子科技大学出版社王飞老师为本书的策划和出版做了大量工作，在此对他们表示衷心的感谢！

由于作者水平有限，书中疏漏之处在所难免，恳请读者来邮件指正。作者电子邮箱：zhangqy@lut.cn。

编　者

2014 年 5 月于兰州

目　录

第1章　软件工程学概论

知识点

软件，软件危机，软件工程，软件工程的方法、模型、工具，软件过程模型。

难点

软件开发过程及过程模型。

基于工作过程的教学任务

• 通过本章的学习，了解和掌握软件工程的基本概念(如软件和软件工程的定义等)，软件危机的表现形式、产生的原因及消除的途径，软件工程的基本原理、方法学，软件的生存期，几种主要的软件开发模型等。

1.1　软件的基本概念

在 20 世纪中叶，软件伴随着第一台电子计算机的问世诞生了。以编写软件为职业的人也开始出现，他们多是经过训练的数学家和电子工程师。20 世纪 60 年代，美国大学里开始出现计算机专业，教学生如何编写软件。软件产业从零开始起步，在短短的五十多年时间里迅速发展成为推动人类社会发展的龙头产业，并造就了一批百万、亿万富翁。随着信息产业的发展，软件对人类社会越来越重要。

1.1.1　软件与软件特性

软件也称计算机软件。如果将计算机比喻成人体，软件就如同人体的骨架(体系结构)、外表(用户界面)、大脑(数据库)、器官(模块)、神经和肌肉(数据结构和算法)。

1．什么是软件

软件是计算机系统中与硬件相互依存的重要组成部分。高质量、多功能的软件使得计算机的应用从单一的科学计算扩展到多个领域，比如数据处理、实时控制等。

软件是计算机系统运行的指令、数据和资料的集合，包括指令程序、数据、相关文档和完善的售后服务，即

$$\text{软件} = \text{程序} + \text{数据} + \text{文档} + \text{服务}$$

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处理信息所需的数据结构及信息表示；文档是与程序开发、维护和使用有关的技术数据和图文资

料，如软件开发计划书、需求规格说明书、设计说明书、测试报告和用户手册等。

2. 软件的特性

软件的特性主要分为有形特性和无形特性。其中，软件的有形特性是软件的各种具体表现形式，包括软件文档、程序代码、二进制代码、用户界面、输出报表等；而软件的无形特性是软件的内部逻辑，是软件本身所包含的思想。

软件的无形特性使得我们只能从软件之外去观察、认识软件，与硬件和传统的工业产品相比较，软件具有以下独特的特性：

(1) 软件是一种逻辑产品，与物质产品有很大的区别。软件产品是看不见摸不着的，因而具有无形性；软件是脑力劳动的结晶，是以程序和文档的形式出现的，通过计算机的执行才能体现其功能和作用。

(2) 软件产品生产的关键主要是研制，软件产品的成本主要体现在软件的开发和研制上。软件一旦研制开发成功后，通过复制就可以产生大量软件产品。

(3) 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。

(4) 软件的开发和运行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，这导致了软件移植的问题。

(5) 软件的开发主要是进行脑力劳动，至今尚未完全摆脱手工作坊式的开发方式，生产效率低，且大部分产品是定制的。

(6) 软件是复杂的，而且以后会更加复杂。软件是人类有史以来生产的复杂度最高的工业产品。软件涉及人类社会的各行各业、方方面面，软件开发常常涉及其他领域的专门知识，这对软件工程师提出了很高的要求。

(7) 软件的成本相当昂贵。软件开发需要投入大量、高强度的脑力劳动，成本非常高，风险也大。现在软件的成本开销已大大超过了硬件的成本开销。

(8) 脆弱性。随着 Internet 的普及，计算机之间经常相互通信和共享资源，这给用户带来方便和利益的同时，也给计算机系统的安全性带来了威胁。

(9) 软件工作牵涉很多社会因素。许多软件的开发和运行涉及机构、体制和管理方式等问题，还会涉及人们的观念和心理等因素。这些人的因素，常常成为软件开发的困难所在，直接影响到项目的成败。

1.1.2 软件的分类

计算机软件是一个涉及多个领域、应用广泛的概念，可以从各个不同的角度对计算机软件进行分类。目前一般的分类方法有以下几种。

1. 基于软件功能的划分

基于软件功能，可将软件划分为系统软件、应用软件和支撑软件等。

系统软件是与计算机硬件紧密结合，以使计算机的各个部件与相关软件及数据协调、高效工作的软件。它有基础性和通用性两大特点。例如，操作系统、数据库管理系统、设备驱动程序等。这些软件一般由专业的软件公司有目的地开发并较好地维护。

应用软件是为特定领域应用、为特定目的服务而开发的一类软件。例如，商业处理软件、科学计算软件、计算机辅助设计软件、人工智能软件等。

支撑软件是协助用户开发软件的工具性软件，包括帮助程序人员开发软件产品的工具和帮助管理人员控制开发进程的工具。例如，需求分析工具、设计工具、编码工具、测试工具、维护和管理工具等。

2．基于软件规模的划分

软件规模是软件项目可量化的结果，通常采用代码行数量或耗用人工时的多少来衡量。

3．基于软件工作方式的划分

基于软件工作方式，可将软件划分为实时处理软件、分时软件、交互式软件和批处理软件。

实时处理软件，例如卫星实时监控软件、外汇实时行情软件等。实时软件既可以应用于信息处理，也可以应用于过程控制。

分时软件，允许多个联机用户同时使用计算机的软件。

交互式软件，能实现人机通信的软件，能接收用户给出的信息，但在时间上没有严格规定。

批处理软件，把一组输入作业或一批数据以成批处理的方式一次运行，顺序逐个处理的软件。

4．按软件服务对象的范围进行划分

按软件服务对象的范围，可将软件划分为项目软件和产品软件。

项目软件也称定制软件，是受某特定客户的委托，由软件开发机构在合同约束下开发的软件，例如气象预测分析软件、交通监控指挥系统、卫星控制系统等。

产品软件也称通用软件，指由软件开发机构开发并直接提供给市场，为众多用户服务的软件，例如文字处理软件、图片处理软件、财务处理软件、人事管理软件等。

5．按使用的频度进行划分

有些软件开发出来仅供一次使用(例如用于人口普查、工业普查的软件)，另外有些软件具有较高的使用频度(例如天气预报软件等)。

6．按软件失效的影响进行划分

有些软件在工作中出现故障而失效后，可能对整个软件系统的影响不大；而有些软件一旦失效，就可能带来灾难性后果(例如财务金融软件、交通通信软件、航空航天软件等)，这类软件称为关键软件。

7．其他几类软件

嵌入式软件。嵌入式计算机系统将计算机嵌入在某一系统中，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。大型的嵌入式软件可用于航空航天系统。小型的嵌入式软件可用于工业的智能化产品中，如移动电话、电子词典、数码相机、机顶盒、MP4、洗衣机、空调机的自动控制等。

基于 Web 的软件。该类软件是基于 B/S(Browser/Server，即浏览器/服务器)结构的软件，如网络游戏软件、在线考试系统、网络银行等。

1.2 软件危机

现代计算机应用系统中，软件的地位日益重要和突出。如何满足日益增长的软件需求，如何维护应用中的大量已有软件，已经成为计算机应用系统进一步发展的瓶颈。20 世纪 60 年代末至 20 世纪 70 年代初，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的那一天起就出现了，只不过到了 1968 年，NATO(北大西洋公约组织)的计算机科学家在原联邦德国召开的国际学术会议上才第一次提出了软件危机(Software Crisis)这个名词。

举例：下面是一个比较典型的软件危机的例子。

IBM 公司在 1963 年—1966 年间开发了 IBM360 机的操作系统。这一项目花了 5000 人一年的工作量，最多时有 1000 人投入开发工作，写出了近 100 万行源程序，总投资 5 亿美元。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 个程序错误后修正的结果。

这个项目的负责人事后总结他在组织开发过程中的沉痛教训时说："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。"

如今，虽然软件开发的技术和工具不断改进，但是软件危机依然没有彻底消除。那么，什么是软件危机呢？

简单地说，所谓软件危机，就是指在软件开发和软件维护过程中所存在的一系列严重问题。这类问题绝不仅仅是不能正常运行的软件才具有的，实际上几乎所有软件都不同程度地存在这类问题。软件危机包含两方面的问题：

(1) 如何开发软件，怎样满足对软件的日益增长、日趋复杂的需求；

(2) 如何维护数量不断膨胀的软件产品。

具体来说，软件危机的主要典型表现与产生的原因有以下几方面。

(1) 对软件开发成本和进度的估计常常很不准确。产生的原因主要是拖期、项目管理经验欠缺。

(2) 软件不能符合用户的要求，用户对已完成的软件系统不满意的现象经常发生。产生的原因主要是模糊的需求、闭门造车、忙于编程、仓促上阵。

(3) 软件产品的质量往往靠不住。产生的原因主要是可靠性和质量保证欠缺，缺少测试。

(4) 软件常常是不可维护的。产生的原因主要是设计死板，没有整体考虑。

(5) 软件通常没有适当的文档资料。产生的原因主要是缺少设计资料、难以维护，写文档嫌麻烦。

(6) 软件成本在计算机系统总成本中所占的比例逐年上升。产生的原因主要是软件过于庞大，成本过高。

(7) 软件开发生产率提高的速度，远远跟不上计算机应用迅速普及深入的速度。产生的原因主要是软件开发的方法跟不上计算机和软件技术的发展速度，技术落后。

(8) 开发者只专注于技术，风险意识薄弱。

消除软件危机的途径主要有以下几个：

(1) 理解软件的概念，软件是程序、数据及相关文档的完整集合。

(2) 应该推广使用在实践中总结出来的开发软件的成功技术和方法。

(3) 应该开发和使用更好的软件工具。

(4) 尽量减少软件维护的代价，提高软件的可维护性，这也是软件工程学的一个重要目标。

所以要解决软件危机中的问题，既要有技术措施(方法和工具)，又要有必要的组织管理措施，必须用工程化的方法管理软件开发过程，用先进的软件开发技术进行软件开发，从管理和技术两方面保证软件开发的质量。

1.3　软 件 工 程

1968 年秋季，NATO 的科技委员会召集了近 50 名一流的编程人员、计算机科学家和工业界巨头，讨论和制定摆脱软件危机的对策，会议上第一次提出了软件工程(Software Engineering)这个概念。当时提出这个概念的 Fritz Bauer 的主要思路是想将系统工程的原理应用到软件的开发和维护中。

所谓软件工程，提倡的是一种软件开发中的系统思想的具体实现，是一门科学，也被称为是软件产业中指导计算机软件开发和维护的软科学，还可以定义为：软件工程是一类设计软件的工程。

《计算机科学技术百科全书》中对软件工程的定义是：应用计算机科学、数学及管理科学等原理，借鉴传统工程的原则、方法，创建软件以达到提高质量、降低成本的目的。其中：计算机科学、数学用于构建模型与算法；工程科学用于制定规范、设计规范、评估成本及确定权衡；管理科学用于计划、资源、质量、成本等管理。

软件工程一直以来都缺乏一个统一的定义，很多学者、组织机构都给出了自己的定义。归纳起来其定义可以总结为：

软件工程是开发、运行、维护和修复软件的系统方法，是一门工程学科，即采用工程的概念、原理、技术和方法来开发和维护软件；也即软件工程是把系统的、有序的、可量化的方法应用到软件的开发、运营和维护上的过程；也即软件工程 = 工程原理 + 技术方法 + 管理技术。

软件工程具有以下本质特性与重点：

(1) 软件工程关注于大型程序的构造——分析与设计；

(2) 软件工程的中心课题是控制系统的复杂性——分解；

(3) 软件经常变化——要有准确的需求；

(4) 开发软件的效率非常重要——经验技巧；

(5) 和谐地合作是开发软件的关键——团队精神；

(6) 软件必须有效地支持它的用户——构造正确的软件系统；

(7) 在软件工程领域中，一般由具有一种文化背景的人替具有另一种文化背景的人进行开发——须具有知识面非常广的领域业务背景。其中，缺乏应用领域的相关知识，是软

件开发项目出现问题的常见原因。

1.3.1 软件工程的基本原理

自从在 1968 年召开的国际会议上正式提出并使用了软件工程这个术语以来，研究软件工程的专家学者们陆续提出了 100 多条关于软件工程的准则。美国著名的软件工程专家 B.W.Boehm 综合这些学者们的意见并总结了 TRW 公司多年开发软件的经验，于 1983 年在论文中提出了软件工程的七条基本原理。人们虽然不能用数学方法严格证明它们是一个完备的集合，但是可以证明在此之前已经提出的 100 多条软件工程准则都可以由这七条原理的任意组合蕴含或派生。

下面简要介绍软件工程的七条基本原理。

1．用分阶段的生命周期计划严格管理

统计表明，50%以上的失败项目是由于计划不周而造成的。Boehm 认为，在整个软件生命周期中应指定并严格执行六类计划：项目概要计划、里程碑计划、项目控制计划、产品控制计划、验证计划、运行维护计划。

2．坚持进行阶段评审

统计结果显示：大部分错误是在编码之前造成的(大约占 63%)，错误发现得越晚，改正它要付出的代价就越大，有时要差 2 到 3 个数量级。因此，软件的质量保证工作不能等到编码结束之后再进行，应坚持进行严格的阶段评审，以便尽早发现错误。

3．实行严格的产品控制

开发人员最痛恨的事情之一就是改动需求。但是实践告诉我们，需求的改动往往是不可避免的。我们要采用科学的产品控制技术来顺应这种要求，也就是要采用变动控制，又叫基准配置管理。当需求变动时，其他各个阶段的文档或代码随之相应变动，以保证软件的一致性。

4．采纳现代程序设计技术

从 20 世纪 60、70 年代的结构化软件开发技术，到面向对象技术，从第一、第二代语言，到第四代语言，人们已经充分认识到：方法比气力更有效。采用先进的技术既可以提高软件开发和维护的效率，又可以减少软件维护的成本，进而提高软件产品的质量。

5．结果应能清楚地审查

软件是一种看不见、摸不着的逻辑产品。软件开发小组的工作进展情况可见性差，难于评价和管理。为更好地进行管理，应根据软件开发的总目标及完成期限，尽量明确地规定开发小组的责任和产品标准，从而使所得到的标准能清楚地审查。

6．开发小组的人员应该少而精

开发人员的素质和数量是影响软件质量和开发效率的重要因素，应该少而精。这基于两点原因：高素质开发人员的效率比低素质开发人员的效率要高几倍到几十倍，开发工作中犯的错误也要少的多；当开发小组为 N 人时，可能的沟通信道为 N(N−1)/2，可见随着人数 N 的增大，沟通开销将急剧增大。

7．承认不断改进软件工程实践的必要性

遵从上述六条基本原理，就能够较好地实现软件的工程化生产。但是，它们只是对现有经验的总结和归纳，并不能保证赶上技术不断前进发展的步伐。因此，Boehm 提出应把承认不断改进软件工程实践的必要性作为软件工程的第七条原理。根据这条原理，不仅要积极采纳新的软件开发技术，还要注意不断总结经验，收集进度和消耗等数据，进行出错类型和问题报告统计。这些数据既可以用来评估新的软件技术效果，也可以用来指明必须着重注意的问题和应该优先进行研究的工具和技术。

1.3.2　软件工程学科包含的领域

软件工程是一门交叉性的工程学科，它将计算机科学、数学、工程学和管理学等基本原理应用于软件开发的工程实践中，并借鉴传统工程的原则和方法，以系统、可控、有效的方式生产高质量的软件产品。即软件工程学结合了工程学和计算机科学的部分内容，主要包含了开发技术和工程管理两方面的内容，具体的组成形式如图 1-1 所示。

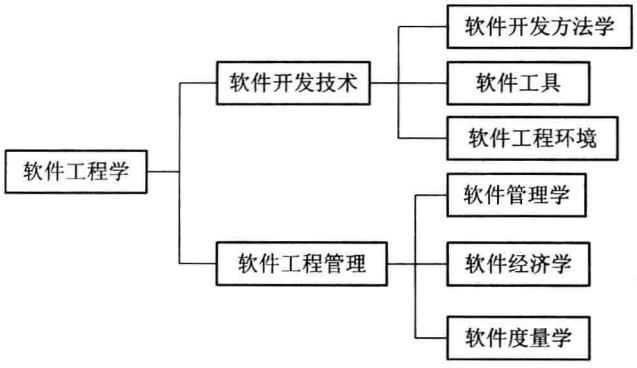

图 1-1　软件工程学领域

1．软件开发技术

软件开发技术又包括软件开发方法学、软件工具和软件工程环境。

1) 软件开发方法学

软件工程方法为软件开发提供了如何做的技术，是指导研制软件的某种标准规范。它包括多方面的任务，如项目计划与估算、软件系统需求分析、数据结构设计、系统总体结构设计、算法设计、编码、测试以及维护等。其中，包括面向对象需求分析、面向对象设计、面向对象编码在内的软件开发方法已成为现在许多软件工程师的首选方法。面向对象技术还促进了软件复用技术的发展，有组件、控件等软件构件方法。

2) 软件工具

软件工具为软件工程方法提供了自动的或半自动的软件支撑环境。目前，已经推出了许多软件工具，这些软件工具集成起来，建立起了称为计算机辅助软件工程(CASE)的软件开发支撑系统。CASE 将各种软件开发方法工具、开发机器和一个存放开发过程信息的工程数据库组合起来形成一个软件工程环境。

3) 软件工程环境

方法与工具的结合，加上配套的系统软、硬件支持就形成了软件开发的环境。在软件

开发工作中，人们不懈地创造着良好的软件开发环境，各种 UNIX 版本操作系统、Microsoft Windows 系列操作系统以及近几年开源文化推动的 Linux 操作系统，还有形式繁多的网络计算环境等，将软件工程环境的研究推到了一个新的领域，如能够支持软件开发过程的辅助工具 CASE。

2．软件工程管理

由于软件本身的特性，软件工程管理既运用管理学的知识，又结合软件特性，形成了软件管理学、软件经济学和软件度量学三个分支。软件工程管理的目的就是为了按照进度和预算来完成软件开发计划，成功地生产软件产品，并实现预期的经济效益和社会效益。软件工程管理的任务是有效地组织人员，按照适当的技术、方法，利用好的工具来完成预定的软件项目。软件工程管理的内容包括软件费用管理、人员组织、工程计划管理、软件配置管理等。

1.4 软件工程的方法、工具与环境

软件工程方法学包含三个要素：方法、工具和过程。其中，方法是完成软件开发的各项任务的技术方法，回答怎样做的问题；工具是为运用方法而提供的自动的或半自动的软件工程支撑环境；过程是为了获得高质量的软件所需要完成的一系列任务框架，它规定了完成各项任务的工作步骤。

1.4.1 软件工程的方法、工具与环境

1．方法

20 世纪 60 年代中期爆发了众所周知的软件危机。为了克服这一危机，在 1968、1969 年连续召开的两次著名的 NATO 会议上提出了软件工程这一术语，并在之后不断发展、完善。与此同时，软件研究人员也在不断探索新的软件开发方法。至今已形成了七类软件开发方法。

1) *传统的结构化软件开发方法——SASD 方法*

1978 年，E．Yourdon 等人提出了结构化方法(SASD 方法)，也可称为面向功能的软件开发方法或面向数据流的软件开发方法。这是 20 世纪 60～70 年代使用最广泛的软件开发方法。

当使用结构化开发方法进行软件开发时，首先要用结构化分析(Structured Analysis，SA)方法对软件进行需求分析；然后用结构化设计(Structured Design，SD)方法进行软件系统的总体设计；最后用结构化程序设计(Structured Programming，SP)编程实现系统。结构化开发方法将软件系统分为两类典型的软件结构：变换型和事务型，使软件开发的成功率大大提高。

2) *面向数据结构的软件开发方法——面向数据的方法*

(1) Jackson 方法。

该方法是 1975 年由 M．A．Jackson 提出的一类至今仍广泛使用的软件开发方法。该