

高等学校公共课计算机教材系列

软件工程概论

卫红春 等编著 刘天时 主审



清华大学出版社

高等学校公共课计算机教材系列

软件工程概论

卫红春 等 编著
刘天时 主审

清华大学出版社
北 京

内 容 简 介

本书简明地讲述了软件工程的基本概念、基本理论、基本方法和基本技术,较深入地介绍了软件工程的策划、分析、设计、实现、测试和管理等工作。简明性、先进性和统一性是本书的特点。

本书可作为高等院校本科各专业软件工程课程教材,以及计算机科学技术和软件工程本科专业软件工程概论和软件工程导论教材,也可以作为从事软件开发、管理、维护和应用的工程技术和管理人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

软件工程概论 / 卫红春等编著. —北京:清华大学出版社, 2007.9
(高等学校公共课计算机教材系列)

ISBN 978-7-302-15497-6

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

中国版本图书馆CIP数据核字(2007)第089971号

责任编辑:郑寅堃 孙建春

责任校对:梁毅

责任印制:李红英

出版发行:清华大学出版社 地 址:北京清华大学学研大厦A座

<http://www.tup.com.cn> 邮 编:100084

c-service@tup.tsinghua.edu.cn

社总机:010-62770175 邮购热线:010-62786544

投稿咨询:010-62772015 客户服务:010-62776969

印装者:北京国马印刷厂

经 销:全国新华书店

开 本:185×260 印 张:13 字 数:314千字

版 次:2007年9月第1版 印 次:2007年9月第1次印刷

印 数:1~4000

定 价:19.00元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:024093-01

出版说明

随着计算机技术的普及及其向其他学科的快速渗透,非计算机专业的学生的计算机知识已普遍不能适应当今的形势,这在就业及进入新的工作方面,就更加突出。而非计算机专业的学生选修计算机专业的课程,并不符合其以应用为主、培养复合型创新人才的教学目标。目前在本科教育中有不少高校建立了以素质教育为取向的跨学科公共课体系,开设了本科生公(通)选课程,以拓宽学生的知识基础,培养不断学习和创造知识的能力和素质,以便在就业与新的工作岗位上取得更大的优势。许多高校在教学体系建设中已将计算机教学纳入基础课的范畴,在非计算机专业教学和教材改革方面也做了大量工作,积累了许多宝贵经验,起到了教学示范作用。将他们的教研成果转化为教材的形式,向全国其他学校推广,对于深化我国高等学校的教学改革具有十分重要的意义。

2005年1月,在教育部下发的《关于进一步加强高等学校本科教学工作的若干意见》中明确指出:“要科学制订人才培养目标和规格标准,把加强基础与强调适应性有机结合,着力培养基础扎实、知识面宽、能力强、素质高的人才,更加注重学生能力培养。要继续推进课程体系、教学内容、教学方法和手段的改革,构建新的课程结构,加大选修课程开设比例,积极推进弹性学习制度建设。”然而,目前明确定位于非计算机专业以应用为主这一教学目标的教材十分缺乏,使得一些公共课不得不选用计算机专业教材或非教材的店销图书及讲义来替代,在这种背景下,出版一套符合目前非计算机专业学习、拓宽计算机及相关领域知识的适用教材以填补这一空白,推进、配合高校新的教改需求,十分必要。清华大学出版社在对计算机基础教学现状进行了广泛的调查研究的基础上,决定组织编写一套《高等学校公共课计算机教材系列》。

本系列教材将延续并反映清华版教材二十年来形成的技术准确、内容严谨的风格,并具有以下特点:

1. 目的明确

本系列教材针对当前高等教育改革的新形势,以社会对人才的需求为导向,以重点学校已开设的公共课程为基础,同时也吸收一般院校的优秀公共课教材,广泛吸纳全国各高等学校的优秀教师参与编写,从中精选出版确实反映非计算机专业计算机教学方向的特色教材,以配套各高校开设公选课程。

2. 面向就业,突出应用

本系列教材力求突出各学科对计算机知识应用的特征,在知识结构上强调应用能力和创新能力,以使學生能比较熟练地应用计算机知识解决实际问题,满足就业单位的需求。

3. 结合教育与学科发展的需求，动态更新

本系列教材将根据计算机学科的发展和各专业的需要进行更新，同时教材的出版载体形式也会随计算机、网络和多媒体技术的发展而变化，以体现教学方法和教学手段的更新。

4. 注重服务

本系列教材都将力求配套能用于网上下载的教学课件及辅助教学资源。

由于各个学校办学特色有所不同，对教材的要求也会呈现自己的特点，我们希望大家在使用教材的过程中，及时给我们提出批评和改进意见，以便我们做好教材的修订改版工作，使其日趋完善。

清华大学出版社

联系人：郑寅堃 Zhengyk@tup.tsinghua.edu.cn

前言

本教材在策划时重点从以下三个方面考虑：第一，作为软件工程概论教材，能够简明、清晰地反映软件工程学科的基本概念、基本方法、基本技术等基本内容。第二，反映软件工程学科的最新发展，较全面地介绍软件过程、软件开发方法、软件建模、软件体系结构等内容，并用国际通用的软件建模语言 UML 对软件建模；软件开发的分析、设计和实现基于面向对象方法。第三，兼顾学科体系的合理性和工程的指导性两个方面。以软件工程学科体系构架教材结构，教材结构能够直接反映软件工程学科的体系框架；重点介绍软件开发中的策划、分析、设计、编程、测试、管理等基本工作。通过本教材的学习，学生能够掌握软件开发的基本方法和基本技能。

本书可作为高等院校本科各专业软件工程课程教材，以及计算机科学技术和软件工程本科专业软件工程概论和软件工程导论课程教材，也可以作为从事软件开发、管理、维护和应用的工程技术和管理人员的参考书。

本书第 1~4 章及第 6、7 两章由卫红春编写，第 5 章由刘天时编写，第 8 章由蒋志新编写，第 9 章由李宁编写，第 10 章由孙夕凌编写，第 11 章由朱欣娟编写，第 12 章由程国建编写。卫红春主编统稿。刘天时详细审阅了本书全稿，提出了许多中肯的意见。王魁生、黄放明等老师对本教材提出了许多宝贵意见，范莉莉、汤慧、赵辉、付强等同学为本书的改错、插图做了一定工作。清华大学出版社郑寅堃、孙建春两位老师为本书的策划和出版做了大量工作。在此对本书的编写和出版做过工作的所有老师和同学表示衷心感谢。

由于作者水平有限，书中缺点和欠妥之处在所难免，恳请读者来邮件指正。本书编辑的电子邮箱是：zhengyk@tup.tsinghua.edu.cn，作者的电子邮箱是：hchwei@x263.net

编 者

2007 年 4 月于西安

目录

CONTENTS

第1章 绪论	1
1.1 软件	1
1.1.1 软件的概念	1
1.1.2 软件的特征	2
1.1.3 软件分类	4
1.2 软件工程概述	5
1.2.1 软件危机	5
1.2.2 软件工程的定义	5
1.2.3 软件工程的基本内容	6
1.2.4 软件工程学科的形成和发展	10
1.2.5 软件工程学科特点	13
本章小结	14
习题	15
第2章 软件过程	17
2.1 软件生存周期	17
2.1.1 软件生存周期	17
2.1.2 软件生存周期模型	18
2.2 软件过程	21
2.2.1 概述	21
2.2.2 软件过程	21
2.3 统一软件开发过程 RUP	24
本章小结	26
习题	26
第3章 软件工程方法	28
3.1 概述	28
3.1.1 软件工程方法的含义	28
3.1.2 软件工程方法的分类	28

3.1.3	软件工程的基础方法	29
3.1.4	软件工程的一般方法	32
3.2	面向功能方法	32
3.2.1	结构化方法的特征	32
3.2.2	结构化分析	34
3.2.3	结构化设计	38
3.3	面向数据方法	42
3.3.1	概述	42
3.3.2	建模阶段	43
3.3.3	设计阶段	45
3.3.4	实现阶段	46
3.4	面向对象方法	47
3.4.1	概述	47
3.4.2	面向对象方法的基本概念	48
	本章小结	53
	习题	54
第4章	软件模型	55
4.1	模型	55
4.2	软件模型	56
4.2.1	软件模型的概念	56
4.2.2	软件建模过程	57
4.2.3	软件建模语言	57
4.3	统一建模语言 UML	58
4.3.1	概述	58
4.3.2	用例图	59
4.3.3	类图与对象图	61
4.3.4	交互图	62
4.3.5	状态图	64
4.3.6	活动图	65
4.3.7	构件图	65
4.3.8	配置图	66
4.3.9	包图	66
	本章小结	67
	习题	67
第5章	软件策划	69
5.1	问题定义	69
5.1.1	问题提出	69

5.1.2 初步调查	70
5.1.3 定义问题	70
5.2 可行性分析	72
5.2.1 可行性分析的意义	72
5.2.2 可行性分析的内容	72
5.3 软件规划	74
5.3.1 概述	74
5.3.2 软件规划的内容	75
5.4 软件开发计划	76
5.4.1 概述	76
5.4.2 软件开发计划的内容	76
本章小结	78
习题	79
第 6 章 软件分析	81
6.1 领域分析	81
6.1.1 领域分析的意义和作用	81
6.1.2 事务型软件领域分析的一般工作	82
6.1.3 业务调查	83
6.1.4 组织目标分析	83
6.1.5 组织机构分析	84
6.1.6 组织职能分析	85
6.1.7 业务流程分析	85
6.1.8 组织实体分析	86
6.2 需求分析	88
6.2.1 概述	88
6.2.2 需求获取	89
6.2.3 需求分析	89
6.2.4 需求描述	94
6.2.5 需求验证	95
本章小结	96
习题	97
第 7 章 软件设计	98
7.1 软件设计的任务和工作	98
7.1.1 平台设计	98
7.1.2 软件体系结构设计	99
7.1.3 详细设计	99
7.1.4 界面设计	99

7.2	软件平台设计	99
7.2.1	网络	99
7.2.2	物理设备	101
7.2.3	操作系统	102
7.2.4	支撑软件	102
7.2.5	CASE 平台	102
7.3	软件体系结构设计	103
7.3.1	软件系统拓扑结构设计	103
7.3.2	软件体系结构模式设计	104
7.3.3	软件结构设计	107
7.4	详细设计	112
7.4.1	业务对象设计	112
7.4.2	功能逻辑设计	116
7.5	界面设计	119
7.5.1	概述	119
7.5.2	输入设计	120
7.5.3	屏幕界面设计	121
	本章小结	127
	习题	127
第 8 章	编程与实现	129
8.1	程序设计	129
8.1.1	程序设计语言	129
8.1.2	程序设计方法	131
8.1.3	程序设计风格	131
8.1.4	面向对象程序设计	135
8.2	系统实现	136
8.2.1	代码实现	137
8.2.2	系统集成	140
8.2.3	软件部署与发布	140
	本章小结	141
	习题	141
第 9 章	软件测试	143
9.1	概述	143
9.1.1	测试的概念	143
9.1.2	测试的过程	144
9.1.3	测试的原则	145
9.2	静态测试	145

9.2.1	静态测试的概念	145
9.2.2	静态测试方法	146
9.3	动态测试	146
9.3.1	白盒法与黑盒法的区别与联系	147
9.3.2	白盒法	147
9.3.3	黑盒法	150
9.4	单元测试	151
9.4.1	概述	151
9.4.2	单元测试的策略	152
9.4.3	面向对象的单元测试	153
9.5	集成测试	154
9.5.1	集成测试概述	154
9.5.2	集成测试的策略	154
9.5.3	面向对象的集成测试	155
9.6	系统测试	156
9.7	验收测试	157
	本章小结	157
	习题	157
第 10 章	软件维护	159
10.1	概述	159
10.2	软件维护的难点和软件的可维护性	160
10.2.1	软件维护的难点	160
10.2.2	软件的可维护性	161
10.3	软件维护的工作	162
10.3.1	收取故障信息	162
10.3.2	排查方法	162
10.3.3	修改程序	163
10.3.4	维护管理	163
	本章小结	163
	习题	163
第 11 章	软件管理	165
11.1	软件质量管理	165
11.1.1	软件质量管理概述	165
11.1.2	软件质量评价	167
11.2	软件文档管理	169
11.2.1	软件文档的分类	170
11.2.2	软件文档的编写	171

11.3 软件项目管理.....	172
11.3.1 软件项目管理概述.....	172
11.3.2 软件开发成本估计.....	173
11.3.3 软件开发进度安排.....	174
本章小结.....	176
习题.....	177
第 12 章 软件工程新技术	178
12.1 形式化方法.....	178
12.1.1 形式化方法的引入.....	178
12.1.2 形式化规格说明语言.....	179
12.1.3 形式化方法的十条戒律.....	180
12.2 净室软件工程.....	181
12.2.1 净室方法的引入.....	181
12.2.2 净室过程模型.....	181
12.2.3 净室软件工程的特点.....	183
12.3 基于构件的软件工程.....	183
12.3.1 基本概念.....	183
12.3.2 基于构件的软件工程.....	184
12.3.3 构件库的建立与使用.....	185
12.4 软件复用与再工程.....	185
12.4.1 基本概念.....	186
12.4.2 业务过程再工程.....	186
12.4.3 软件再工程.....	187
12.5 敏捷软件过程.....	189
12.5.1 基本概念.....	189
12.5.2 敏捷软件开发方法的指导原则.....	189
12.5.3 典型的敏捷过程模型.....	190
本章小结.....	192
习题.....	192
参考文献	193

第 1 章

绪 论

信息技术是 20 世纪以来改变人类生活方式的最为关键的技术之一，计算机技术是信息技术的核心，而软件可以说是计算机系统的灵魂。软件工程是指导软件开发、服务和管理的科学，对软件的生产 and 应用具有重要作用。

教学要求

- ① 了解软件的概念、软件的特征和软件分类。
- ② 了解软件危机产生的原因和背景，以及对软件发展的影响。
- ③ 掌握软件工程的概念。
- ④ 掌握软件工程的基本内容。
- ⑤ 了解软件工程学科的特点。

重点和难点

- ① 软件工程的概概念。
- ② 软件工程的基本内容。

1.1 软件

1.1.1 软件的概念

1. 软件概念的提出

计算机诞生的初期，人们把它看成一种智能型产品，认为计算机主要由控制器、存储器、运算器、输入设备、输出设备等五部分组成，计算机的功能由计算机的各个部件通过有机协调地工作来完成。当时，程序在计算机系统中的作用并没有引起足够的重视，人们认为程序只是计算机系统的附属部分。

随着计算机技术的发展和应用范围的扩大，到 20 世纪 60 年代初出现了高级语言。一方面，同一个程序经过不同的编译可以在不同的机器上运行，程序具有了相对的独立性；另一方面，程序在计算机系统中的作用也日显重要，同一台计算机能够应用不同的业务领域，发挥不同的作用，其主要原因是因为运行了不同的程序。程序在计算机系统中的作用开始引起人们的重视。

程序是为了完成确定的应用需求而编制的，并且是计算机系统的组成部分。计算机系统要完成特定的任务，除了需要具有计算机硬件设备之外，还需要能够运行的程序和处理的数据，另外，开发程序的过程、方法和文档对计算机系统也十分必要。于是人们根据这些要素的形态特征，把它们划分为两种类型：一类是构成计算机系统的有形设备，称为硬件；另一类是计算机系统的重要组成部分，但又不像硬件那样具有明显的可见性，像程序、数据、方法、过程、文档等，被称为软件。

2. 软件的定义

软件是计算机系统程序、数据及其相关文档的总称。程序是能完成确定任务，用计算机语言描述的，并能够在计算机上执行的语句序列。程序是软件的重要组成部分，同时也是软件的主要表现形式。数据用来描述软件的处理对象。程序的作用就是把输入数据加工处理成为人们所需要的信息。文档是指与程序开发、维护和使用有关的图文资料，是对软件开发和维护全过程的书面描述和记录。

随着计算机技术的发展，软件已经形成为一门学科，被称为软件学，简称软件。软件学科主要包括软件语言、软件方法、软件工程和软件系统等方面的内容。

1.1.2 软件的特征

软件是一类独特的客观事物，它具有以下一些特征。

1. 智能性

软件是人类智力劳动的产物。软件开发是高度的智能活动，软件中的程序、流程、算法、数据结构是通过人的思维进行设计、编排和组织的。当然，机械、建筑、化工、纺织等领域所生产和加工的产品之中都蕴藏着人的智能，但比较起来，软件的智能性特征更为集中和明显。

软件是对人类智能劳动的代替和延伸。对于本来需要人的智力参与的工作，可以由软件来代替完成。软件在数据统计、科学计算、事务处理、辅助决策等方面的应用，都是对人类智能劳动的代替和延伸。

2. 无形性

软件不像有形产品那样具有明显的物理形貌，人们无法直接观察到它的形态。有人把计算机系统软件和硬件比做人的思维和大脑，大脑的生物组织比做硬件，大脑中的思维和意识则比做软件。没有思维的大脑不具有大脑的功能，所以思维和意识是人类大脑的重要组成部分。同样道理，没有软件的计算机系统，无法完成它所应具备的功能，所以软件是计算机系统的重要组成部分。

软件的无形性基于计算机系统的光电特征。电子和绝大部分光处在人的感觉空间之外，人的视觉和听觉无法直接感觉到这些物质的存在，软件存储在光、电、磁等介质之中，人们无法直接感觉到。软件的无形性增加了人们对它认识 and 理解的困难。

3. 抽象性

软件属逻辑实体，而非物理实体，具有抽象性。首先，软件的无形性和智能性使得软件难以认识和理解，这是软件在认识上的抽象性；其次，在软件研制过程中，需要进行调研和分析，需要对软件进行逻辑设计和组织，要大量运用到抽象思维和抽象方法；另外，软件被封装在计算机之中，人们只能通过用户界面来与软件交互，软件丰富内涵被蕴涵在计算机内部，使软件具有高度的抽象性。软件的抽象性增加了人们对于软件理解和开发的难度。

4. 系统性

软件是由多种要素组成的有机整体，具有显著的系统特性。软件具有确定的目标、功能、结构和要素。软件所服务的业务领域和运行的软硬件平台是软件的环境，环境承托和约束着软件作用的发挥。软件的系统性还体现在必须采用系统方法来进行软件开发。

5. 泛域性

泛域性指软件可以服务于各种行业和领域。在所有人类智能可以涉足的领域，软件都能够发挥作用。现在已经没有哪一个行业和领域可以不使用计算机系统。使用计算机系统必然要运用软件，科学计算、事务处理、企业管理、智能决策都要使用软件，所以软件服务面向自然、社会和思维的各个领域。

软件对各领域的服务还体现在软件与所服务领域的知识的结合性。软件不像其他产品那样以独立的产品形态服务于应用领域（像电话、书柜、工具等产品在厂家的车间里生产出来后直接应用于所需要的领域），软件必须把所服务的领域中的知识、过程、业务、方法、技术、信息等内容结合进去。所以，软件与业务领域不只是一种服务关系，而是与领域的知识结合、渗透和相融的关系。这种关系决定了软件的复杂性和软件生产的困难性。开发软件不仅要考虑软件自身的问题，更多地要考虑软件所服务的领域中的知识、过程、业务、方法、技术、信息等问题，而这些问题常常是开发软件的难点所在。

6. 依附性

软件不像其他产品和设备能够独立存在和工作，软件的开发和运行必须依附于软件环境。软件环境由计算机系统硬件、通信网络、支撑软件及所服务的业务领域等要素构成。软件的依附性决定了人们在软件开发过程中必须考虑软件环境对软件的制约和影响。

7. 非损性

软件在使用过程中不像其他产品存在损耗和老化现象，只要硬件环境不发生故障和变化，软件可以永久使用。

8. 复制性

由于软件存储在光、电、磁等介质上，所以软件可以复制。一个软件可以被大量复制，

而对原软件没有任何影响。软件的可复制性，决定了软件开发成本主要体现在软件首次开发过程中，软件一旦开发出来，复制和传播的费用一般较低。它不像其他有形产品的研制和生产，任何一批产品的生产都需要经过完整的从原材料到成品的完整的加工过程。

9. 演化性

软件投入运行之后，其功能、性能、硬件环境都处于不断变化之中，我们把软件在其生存周期中不断变化的特性称为软件的演化性。软件具有演化性是因为软件所处的问题域环境在不断变化，人们对软件的需求不断变化，计算机技术在不断变化，因此，软件需要随着环境、需求和技术的变化而变化。软件的演化性也决定了软件在整个生存周期中要不断地改进和完善，这就是软件维护的工作。

1.1.3 软件的分类

1. 按作用分类

按照作用可以把软件划分成为系统软件、支撑软件和应用软件三种类型。

系统软件是最靠近计算机硬件的一类软件，它能与计算机硬件紧密结合，使计算机系统的各个部件和相关软件能够协调有效地工作。系统软件在计算机系统中起着最关键的作用，其他软件都建立在系统软件的基础之上，且系统软件与具体的应用领域没有关系。操作系统是最重要也是最典型的系统软件，计算机系统通过操作系统来管理系统资源、协调计算机系统的工作，并方便用户使用计算机系统。

支撑软件是协助人们开发和维护软件的工具和环境软件。编辑程序、数据库系统、集成开发环境等都属于支撑型软件。

应用软件则是指为特定应用领域服务的一类软件。在日常工作和生活中，大量使用的是应用软件。股票分析软件、航空售票软件、企业综合信息系统、地理信息系统等都属于应用软件。

2. 按规模分类

按照软件开发所需要的人力、时间以及软件的大小，可以把软件划分成为微型、小型、中型、大型和超大型这五种类型，如表 1-1 所示。现在微型软件和小型软件较少，绝大部分是中大型软件。

表 1-1 软件规模划分

类型	开发人员	开发期限	源程序行数
微型	1 名	1~4 周	小于 500 行
小型	1~2 名	1~6 月	500~5000 行
中型	2~5 名	1~2 年	5000~50000 行
大型	5~20 名	1~3 年	50000~100000 行
超大型	20 名以上	3 年以上	10 万行以上

1.2 软件工程概述

1.2.1 软件危机

在 20 世纪 60 年代中期的一段时间内，计算机软件的开发和应用过程中出现了几种带有普遍性的问题：软件开发不能按照预定计划控制和完成，存在拖延工期的现象；开发出来的软件满足不了用户的需要，用户对软件的作用和效果感到失望；软件开发和维护费用远远超出预算，软件耗资巨大；软件的可靠性和可用性差。这些问题并不是仅仅存在于部分软件中，而是当时软件开发中普遍存在的问题，并成为影响软件应用和发展的严重障碍。人们甚至开始怀疑软件到底能不能在社会中得到广泛应用。软件在应用和发展过程中出现的这种现象被称为软件危机。

软件危机的出现打碎了人们对软件应用前景过分乐观的期待，迫使人们开始冷静地思考危机出现的原因，以及解决软件危机的途径和方案。产生软件危机的原因可以概括为以下几个方面：

- 人们对软件本质认识不足。软件是一种特殊产品，它的智能性、无形性、抽象性等特性决定了软件不同与一般物理产品的生产，软件的生产过程更多体现了在人的智能参与下的创造性活动。因此，软件生产应该采用一套适合软件特点的方法和技术。
- 重视软件开发的技巧性，忽视软件开发的工程性。软件开发离不开程序设计，但软件开发不等同于程序设计。当时，人们还没有脱开软件开发就是程序设计的观念，因此，软件开发过程中，更多追求的是程序设计的方法 and 技巧，而忽略了软件开发是一项复杂的工程过程，没有主动采用工程的观点和方法来指导和组织软件开发。
- 软件的开发技术、开发方法和开发工具落后。软件开发是一项复杂的工程过程，要求有能够有效指导软件开发的方法和技术，以及协助软件开发的工具和手段。20 世纪 60 年代，还没有出现成熟的软件开发技术和方法，软件开发工具也很落后，因此软件开发的水平和效率低下，开发成功率很低。

为了解决软件危机，人们开始认识软件的自身特点，研究软件开发的内在规律，把工程化思想引入软件开发，并在软件开发的方法、技术和工具等方面进行了深入研究，产生了软件工程这门新兴学科。

1.2.2 软件工程的观念

软件工程 (software engineering) 的概念是 1968 年在北大西洋公约组织举行的一次学术会议上首次提出来的。经过几十年的发展，软件工程已经成为一门独立的学科，称为软件工程学 (software engineering science)。《计算机科学技术百科全书》中对软件工程的定义是：软件工程是应用计算机科学技术、数学、管理学的原理，运用工程科学的理论、方法和技术，研究和指导软件开发和演化的一门交叉学科。应该从以下几个方面理解软件工程的观念。