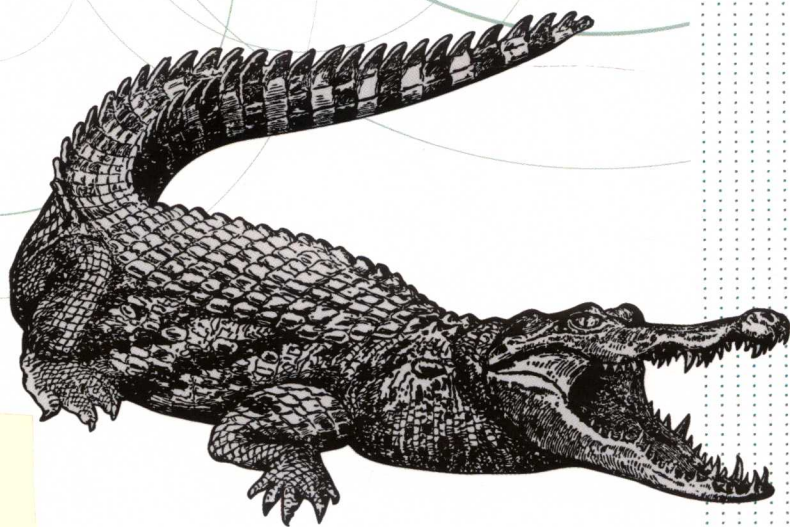


移动互联网应用「赢在起点」系列图书

# 软件工程

主编  
陆惠恩



上海交通大学出版社  
SHANGHAI JIAO TONG UNIVERSITY PRESS

移动互联应用“赢在起点”系列图书

# 软 件 工 程

主 编 陆惠恩

副主编 曹 辉



上海交通大学出版社  
SHANGHAI JIAO TONG UNIVERSITY PRESS

## 内容提要

本书着重从实用角度讲述软件工程的基本概念、原理和方法,介绍如何规范地开发和维护软件;合理地安排软件开发和维护的过程;规范地书写软件工程的文档资料,以提高软件开发的效率和质量。全书分12章,内容涵盖:软件工程概述(软件生存周期、开发模型),软件计划(问题定义、可行性研究),需求分析(方法、步骤、图形工具、数据字典),结构化设计、过程设计工具、界面设计、数据代码设计、软件设计文档,软件编码和测试,软件维护、软件重用、逆向工程、再工程,面向对象方法与UML,面向对象软件设计与实现,Web软件工程(Web软件特性、层次结构、可用技术、设计模式、WebApp设计与测试),移动互联网与软件工程(移动互联网三要素、应用技术、安全技术),软件工程管理,软件工程技术的发展(敏捷开发与极限编程、净室软件工程、面向服务的架构技术、云计算、网构软件),软件开发实例——网上商品竞拍系统。

本书可作为计算机及相关专业的教材,也可供从事计算机软件开发及维护的科技人员参考。

## 图书在版编目(CIP)数据

软件工程 / 陆惠恩主编. -- 上海: 上海交通大学出版社, 2016  
ISBN 978-7-313-14520-8

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

中国版本图书馆CIP数据核字(2016)第024999号

## 软件工程

主 编: 陆惠恩

出版发行: 上海交通大学出版社

地 址: 上海市番禺路951号

邮政编码: 200030

电 话: 021-64071208

出 版 人: 韩建民

印 制: 三河市祥达印刷包装有限公司

经 销: 全国新华书店

开 本: 787mm×1092mm 1/16

印 张: 27.25 字 数: 480千字

版 次: 2016年8月第1版第1次印刷

书 号: ISBN 978-7-313-14520-8/TP

定 价: 58.00元

版权所有 侵权必究

告读者: 如发现本书有印装质量问题请与发行部联系

联系电话: 010-62137141



# 前言

软件工程是指导计算机软件开发和维护的学科。软件工程采用工程的概念、原理、技术和方法,把良好的技术和正确的管理方法结合起来开发软件。软件工程学已成为计算机科学与技术相关专业的一门重要学科。

本书是作者根据 20 多年软件工程学的理论教学与实训指导以及软件项目开发的实际经验,着重从实用角度讲述软件工程的基本概念、原理和方法,介绍如何规范地开发和维护软件;合理地安排软件开发和维护的过程;培养和提高软件开发、维护的能力;规范地书写软件工程的文档资料,以提高软件开发的效率和质量。

移动互联网系统包含智能移动终端、移动互联网络及移动互联应用三个层面。对于移动互联应用专业的学生,在学习软件工程课程时,要结合移动互联系统的特点,按照工程的原理和方法来开发和维护软件,以提高软件开发效率和移动互联服务的质量。

## 本书特色

(1) 语言流畅、实用性强。每章有小结,有适量的实例和习题,有些实例贯穿于各章,可作为软件开发实训参考题。附录中有部分习题的参考答案,有利于读者对一些实例进行需求分析、建立系统模型、进行系统设计。

(2) 引入软件工程较新的技术。例如面向对象技术、统一建模语言(UML)、软件重用技术、Web 软件的层次结构和设计模式、WebApp 的设计和测试、移动互联网应用技术、面向服务的架构技术、云计算等。

(3) 介绍了软件工程各阶段的文档规范,供读者参考。

(4) 最后的软件开发实例——网上商品竞拍系统,可作为实训题目,让读者对软件开发的全过程有一个感性认识。

## 本课程教学重点

- 软件工程的结构化方法(结构化分析、结构化设计和结构化程序设计);
- 面向对象方法与 UML;

- 软件测试、软件质量保证；
- Web 软件的层次结构和设计模式、WebApp 的设计和测试；
- 移动互联网的组成——移动互联网终端设备、移动互联网网络、移动互联网应用和移动互联网相关技术四大部分。

软件工程课程在程序设计语言、数据库原理、数据结构、移动互联应用开发技术等专业课之后，毕业实习、毕业设计之前开设。建议理论学习为 45~50 学时，并适当地安排实践环节。通过软件开发的实际训练来培养和提高学生开发、维护软件的能力。软件工程实践环节可要求学生完成一个难度适当的软件设计课题。时间安排上，可在理论教学结束后集中 2~4 周安排课程设计，也可在理论教学的适当阶段同步安排实践环节，分阶段逐步完成课题。

本书可作为计算机相关专业、移动互联应用专业的教材，也可供从事计算机软件开发及维护的广大科技人员参考。

本书由陆惠恩任主编，曹辉任副主编，由张成姝、胡维荣、陆培恩参与编写。书中存在的错误和不足之处，敬请读者批评指正。



# 目录

<b>第 1 章 概述</b> .....	1	<b>第 2 章 软件计划</b> .....	29
1.1 软件工程的产生.....	2	2.1 软件问题定义及可行性研究.....	30
1.1.1 软件工程的发展史.....	2	2.1.1 软件问题定义.....	30
1.1.2 软件危机.....	5	2.1.2 可行性研究.....	31
1.2 软件工程学.....	6	2.1.3 制订项目开发计划.....	32
1.2.1 什么是软件.....	6	2.2 需求分析.....	33
1.2.2 什么是软件工程.....	10	2.2.1 需求分析方法.....	34
1.2.3 软件工程学的内容.....	10	2.2.2 需求分析步骤.....	36
1.2.4 软件过程.....	14	2.3 需求分析图形工具.....	41
1.2.5 软件工程的基本原理.....	14	2.3.1 实体—关系图.....	41
1.3 软件生存周期.....	15	2.3.2 数据流图.....	43
1.3.1 软件生存周期的概念.....	15	2.3.3 状态转换图.....	48
1.3.2 软件生存周期各阶段的任务.....	16	2.3.4 层次图.....	50
1.4 软件开发模型.....	17	2.3.5 Warnier 图.....	51
1.4.1 瀑布模型.....	17	2.3.6 IPO 图.....	51
1.4.2 快速原型模型.....	18	2.4 数据字典.....	52
1.4.3 增量模型.....	20	2.4.1 数据字典的内容.....	52
1.4.4 喷泉模型.....	21	2.4.2 数据字典使用的符号.....	53
1.4.5 螺旋模型.....	22	2.4.3 数据字典与图形工具.....	55
1.4.6 统一过程.....	24	2.5 软件计划阶段文档.....	55
本章小结.....	25	2.5.1 软件计划阶段文档 的编写步骤.....	55
习题 1.....	26		

2.5.2 软件计划实例·····	56	3.7.3 代码的种类·····	100
本章小结·····	62	3.7.4 代码的设计方法·····	103
习题 2·····	62	3.8 面向数据结构的设计方法·····	104
<b>第 3 章 结构化设计·····</b>	<b>65</b>	3.9 软件设计文档·····	108
3.1 软件设计步骤·····	66	3.9.1 概要设计说明书·····	108
3.1.1 概要设计的一般步骤·····	66	3.9.2 数据库设计说明书·····	110
3.1.2 详细设计的基本任务·····	69	3.9.3 详细设计说明书·····	110
3.2 软件结构设计·····	70	3.9.4 操作手册编写提示·····	111
3.2.1 软件结构设计的基本原理·····	70	本章小结·····	113
3.2.2 模块化·····	72	习题 3·····	113
3.2.3 模块独立性·····	74	<b>第 4 章 软件编码和软件测试·····</b>	<b>117</b>
3.2.4 模块设计启发规则·····	78	4.1 结构化程序设计·····	118
3.3 软件结构设计的图形工具·····	79	4.1.1 程序设计语言的选择·····	118
3.3.1 层次图和 HIPO 图·····	79	4.1.2 程序设计风格·····	120
3.3.2 结构图·····	80	4.2 软件测试目标·····	122
3.4 面向数据流的设计方法·····	82	4.3 软件测试方法·····	123
3.4.1 变换型·····	83	4.3.1 静态分析与动态测试·····	123
3.4.2 事务型·····	83	4.3.2 黑盒法与白盒法·····	124
3.5 过程设计工具·····	84	4.4 软件测试步骤·····	124
3.5.1 流程图·····	85	4.4.1 模块测试·····	125
3.5.2 盒图 (N-S 图)·····	89	4.4.2 集成测试·····	125
3.5.3 问题分析图 (PAD 图)·····	91	4.4.3 程序审查会和人工运行·····	126
3.5.4 判定表·····	93	4.4.4 确认测试·····	127
3.5.5 判定树·····	94	4.4.5 平行运行·····	128
3.5.6 过程设计语言 (PDL)·····	95	4.5 设计软件测试方案·····	128
3.6 用户界面设计·····	95	4.5.1 等价类划分法·····	129
3.6.1 用户界面设计问题·····	95	4.5.2 边界值分析法·····	129
3.6.2 用户界面设计过程 和设计标准·····	97	4.5.3 错误推测法·····	130
3.6.3 用户界面设计指南·····	97	4.5.4 逻辑覆盖法·····	130
3.7 数据代码设计·····	99	4.5.5 程序环形复杂程度的度量·····	134
3.7.1 代码的功能和性质·····	99	4.5.6 因果图法·····	137
3.7.2 代码的设计原则·····	100	4.5.7 用基本路径覆盖法设计 测试用例·····	140

4.6 软件测试原则和策略.....	143	6.3.2 类图和包.....	187
4.6.1 软件测试原则 .....	143	6.3.3 对象图.....	192
4.6.2 实用测试策略 .....	144	6.3.4 状态图.....	193
4.7 软件调试、验证与确认.....	144	6.3.5 顺序图.....	194
4.7.1 软件调试方法和技术 .....	144	6.3.6 活动图.....	195
4.7.2 软件验证 .....	145	6.3.7 协作图.....	196
4.7.3 软件确认 .....	146	6.3.8 构件图.....	197
4.8 软件测试文档.....	147	6.3.9 部署图.....	198
本章小结.....	148	6.4 UML 的应用.....	199
习题 4.....	148	6.4.1 UML 模型.....	200
<b>第 5 章 软件维护和软件重用.....</b>	<b>153</b>	6.4.2 UML 视图.....	201
5.1 软件维护.....	154	6.4.3 UML 使用准则 .....	203
5.1.1 软件维护的种类.....	154	6.4.4 UML 的扩展机制.....	204
5.1.2 软件维护的特点 .....	155	6.4.5 UML 的应用领域.....	205
5.1.3 软件维护过程 .....	157	本章小结.....	206
5.1.4 提高软件的可维护性 .....	161	习题 6.....	206
5.2 软件重用和再工程 .....	164	<b>第 7 章 面向对象软件设计与实现.....</b>	<b>209</b>
5.2.1 软件的重用.....	164	7.1 面向对象分析.....	210
5.2.2 软件重用过程模型.....	165	7.1.1 面向对象分析过程 .....	210
5.2.3 开发可重用的软件构件.....	167	7.1.2 面向对象分析原则 .....	211
5.2.4 软件逆向工程 .....	169	7.1.3 建立对象模型.....	212
5.2.5 软件再工程.....	170	7.1.4 建立动态模型.....	220
本章小结.....	171	7.1.5 建立功能模型.....	224
习题 5.....	172	7.2 软件架构设计和对象设计 .....	226
<b>第 6 章 面向对象方法学与 UML.....</b>	<b>175</b>	7.2.1 软件架构设计.....	226
6.1 面向对象方法概述 .....	176	7.2.2 软件架构风格.....	228
6.1.1 面向对象方法的要素和优点 .....	177	7.2.3 移动互联系统的软件架构 .....	234
6.1.2 面向对象方法的概念 .....	179	7.2.4 对象设计.....	236
6.2 UML 概述.....	182	7.2.5 面向对象设计的准则 和启发式规则.....	237
6.2.1 UML 的发展 .....	182	7.3 面向对象系统的实现.....	238
6.2.2 UML 的内容 .....	182	7.3.1 选择程序设计语言 .....	238
6.3 UML 图.....	185	7.3.2 面向对象程序设计风格.....	239
6.3.1 用例图 .....	185		

7.3.3 面向对象实现的人员分工	241	8.6.3 WebApp 需求分析	293
7.4 面向对象系统的测试	242	8.6.4 WebApp 设计过程	296
7.4.1 面向对象测试策略	242	8.6.5 WebApp 体系结构设计	300
7.4.2 面向对象的测试步骤	244	8.6.6 导航设计	303
7.5 面向对象方法实例	249	8.7 WebApp 测试	304
7.5.1 面向对象分析实例	249	8.7.1 测试过程概述	305
7.5.2 面向对象设计实例	255	8.7.2 内容测试	305
本章小结	262	8.7.3 界面测试	306
习题 7	262	8.7.4 导航测试	307
<b>第 8 章 Web 软件工程</b>	<b>265</b>	8.7.5 配置测试	308
8.1 Web 软件的特性	266	8.7.6 安全性测试	308
8.2 Web 的层次结构	268	8.7.7 性能测试	309
8.2.1 二层 C/S 结构	268	本章小结	310
8.2.2 三层 C/S 结构	270	习题 8	311
8.2.3 四层 C/S 结构	272	<b>第 9 章 移动互联网与软件工程</b>	<b>313</b>
8.2.4 B/S 结构	273	9.1 移动互联网简介	314
8.3 客户端层可用的技术	274	9.2 移动互联网应用技术	319
8.3.1 基本 HTML	275	9.2.1 移动互联网的参考模型	320
8.3.2 脚本语言	275	9.2.2 移动互联终端技术	320
8.3.3 Applet	276	9.2.3 网络接入技术	323
8.3.4 AJAX	277	9.2.4 移动应用服务技术	324
8.4 Web 服务器层使用的技术	278	9.2.5 移动互联网的三要素	326
8.4.1 Servlet	278	9.3 移动互联网应用安全技术	327
8.4.2 JSP	280	本章小结	329
8.5 Web 软件设计模式	281	习题 9	329
8.5.1 观察者模式	281	<b>第 10 章 软件工程管理</b>	<b>331</b>
8.5.2 组合模式	283	10.1 软件工程管理概述	332
8.5.3 工厂方法模式和策略模式	284	10.2 软件规模估算	333
8.5.4 模型—视图—控制器模式	285	10.2.1 软件开发成本估算方法	333
8.5.5 装饰者模式	287	10.2.2 代码行技术	334
8.6 WebApp 设计	288	10.2.3 功能点技术	335
8.6.1 WebApp 的特点及应用类型	288	10.2.4 COCOMO2 模型	337
8.6.2 WebApp 的需求工程	289		

10.3 进度计划	338	11.4.2 云计算的关键技术	379
10.3.1 Gantt 图	339	11.4.3 云计算平台与软件开发	381
10.3.2 工程网络技术	340	11.4.4 云计算与物联网	382
10.4 人员组织	343	11.5 网构软件	383
10.5 软件配置管理	346	11.5.1 网构软件的概念	384
10.6 软件质量保证	349	11.5.2 网构软件模型特点	385
10.6.1 软件质量的特性	349	11.5.3 网构软件开发方法	386
10.6.2 软件质量保证措施	350	本章小结	388
10.7 软件开发风险管理	351	习题 11	388
10.7.1 软件开发风险标识	351	<b>第 12 章 实例——网上商品竞拍系统</b>	<b>389</b>
10.7.2 软件开发风险估算	354	12.1 问题定义和可行性研究	390
10.7.3 软件开发风险管理	356	12.1.1 问题定义	390
10.8 软件工程标准与软件文档	358	12.1.2 可行性研究	390
10.8.1 软件工程标准	358	12.2 需求分析和概要设计	391
10.8.2 软件文档的编写	360	12.2.1 系统角色的功能设计	391
本章小结	361	12.2.2 数据库设计	393
习题 10	362	12.2.3 系统结构设计	395
<b>第 11 章 软件工程技术的发展</b>	<b>365</b>	12.2.4 网络结构设计方案	396
11.1 敏捷开发与极限编程技术	366	12.2.5 网络设计方案	397
11.1.1 敏捷开发的含义	366	12.3 模块设计	397
11.1.2 极限编程	368	12.3.1 注册登录模块设计	397
11.2 净室软件工程	370	12.3.2 用户信息管理模块设计	399
11.2.1 净室软件工程的定义	370	12.3.3 拍卖商品管理模块设计	400
11.2.2 净室软件工程技术	371	12.3.4 拍卖业务管理模块设计	400
11.3 面向服务的架构技术	373	12.4 软件测试	401
11.3.1 面向服务架构的产生	373	本章小结	403
11.3.2 面向服务架构的概念	375	习题 12	403
11.3.3 基于 SOA 架构的软件 开发方法	376	<b>附录 部分习题参考答案</b>	<b>405</b>
11.4 云计算与软件开发	377	<b>参考文献</b>	<b>423</b>
11.4.1 云计算概述	377		

# 第1章 概述

在现代社会，计算机软件普遍应用于人们生产、生活、学习和工作的各个方面，像工业、农业、银行、航空、政府部门等行业几乎都有计算机软件的应用。典型的软件有电子邮件、人机界面、办公软件、操作系统、数据库、游戏等。此外，随着科技的发展，除了打电话之外，越来越多的人使用手机来访问互联网。如今，手机上网使用率已经超越传统PC，成为第一大上网终端设备。计算机软件和移动互联网的广泛应用促进了经济和社会的发展，提高了人们的工作效率，同时提升了人们的生活质量。

如何才能开发出用户满意的软件？如何以较低的成本开发出高质量的软件？怎样使所开发的软件在运行过程中容易维护，以延长软件的使用期限？如何提高软件开发、维护过程中的自动化程度，以及软件开发效率？软件开发、维护过程如何管理？这些就是软件工程研究的问题。

软件工程是指导计算机软件开发和维护的学科，它用工程化方法构建和维护有效的、实用的和高质量的软件，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。软件工程的目的是在规定的时间和开发费用内，开发出满足用户需要的、质量合格的软件产品。

本章介绍软件工程的产生，软件工程的基本概念、内容及基本原理，软件生存周期，软件开发模型等。

★ **本章重点：**软件工程；软件生存周期。

## 1.1 软件工程的产生

计算机系统的发展、软件应用的日益广泛和软件危机的困扰，促进了软件工程的产生和发展。

### 1.1.1 软件工程的发展史

自从20世纪40年代电子计算机发明以来，计算机软件随着计算机硬件的发展而逐步发展，软件和硬件一起构成计算机系统。一开始只有程序的概念，后来才出现软件的概念。

随着计算机系统的发展，软件的生产大体经历了程序设计、软件、软件工程、第4代技术等阶段。在此过程中，软件危机产生并越来越严重，因而逐步形成了研究如何消除软件危机，以及如何合理地开发和维护软件的学科——软件工程学。

#### 1. 程序设计阶段

20世纪40年代中期到20世纪60年代中期，电子计算机价格昂贵、运算速度低、存



储量小。计算机程序主要是描述计算任务的处理对象和处理规则。早期的程序规模小，程序往往是个人设计、自己使用。在进行程序设计时，通常要注意如何节省存储单元、提高运算速度。除了程序清单之外，没有其他任何文档资料。

## 2. “软件=程序+文档”阶段

20世纪60年代中期到20世纪70年代中期，集成电路计算机的运算速度和内存容量大大提高。随着程序数量的增加，人们把程序区分为系统程序和应用程序，并把它们称为软件（Software）。随着计算机技术的发展，计算机软件的应用范围越来越广泛，当软件需求量大大增加后，许多用户去“软件作坊”购买软件。

人们把软件视为产品，确定了软件生产的各个阶段必须完成的，有关计算机程序的功能、设计和使用的文字或图形资料，并把这些资料称为文档（Documents）。软件是对计算机程序及其有关数据和文档的总称。

软件产品交付给用户使用之后，为了纠正错误或适应用户需求的改变，对软件进行的修改，称为软件维护（Software Maintenance）。此时，由于在软件开发过程中很少考虑到将来的维护问题，软件维护的费用以惊人的速度增长，不能及时满足用户需求，质量得不到保证。所谓的“软件危机”就是由此开始的。人们开始重视软件的“可维护性”问题，软件开发采用结构化程序设计技术，规定软件开发时必须书写各种规格书、说明书、用户手册等文档。

1968年，北大西洋公约组织（NATO）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，正式提出了“软件工程”（Software Engineering）这一术语。从此一门新兴的工程学科诞生了。

## 3. 软件工程阶段

20世纪70年代中期到20世纪90年代，大规模集成电路计算机的功能和质量不断提高，个人计算机已经成为大众化商品，计算机应用领域不断扩大。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的趋势，软件产品供不应求，软件危机日益严重。为了维护软件要耗费大量的成本。美国当时的统计数据表明，对计算机软件的投资占计算机软件、硬件总投资的70%，到1985年软件成本大约占总成本的90%。

为了对付不断增长的软件危机，软件工程学把软件作为一种产品进行批量生产，运用工程学的基本原理和方法来组织和管理软件生产，以保证软件产品的质量和提高软件产品生产率。软件生产使用数据库、软件开发工具、开发环境等，软件开发技术有了很大的进步，开始采用工程化开发方法、标准和规范，以及面向对象技术等。



#### 4. 第4代技术阶段

计算机系统发展的第4阶段不再是单台的计算机和计算机程序，而是面向计算机和软件的综合影响。由复杂操作系统控制的强大的桌面系统，以及局域网和因特网、高带宽的数字通信，与先进的应用软件相互配合，产生了综合的效果。计算机体系结构已从集中的主机环境转变为分布式的客户机/服务器环境。

随着计算机技术的突飞猛进，软件开发的第4代技术有了新的发展。计算机辅助软件工程（Computer Aided Software Engineering, CASE）将工具和代码生成器结合起来，为许多软件系统提供了可靠的解决方案。面向对象技术已在许多领域迅速取代了传统的软件开发方法，专家系统和人工智能软件有了实际应用，人工神经网络软件展示了信息处理的美好前景；并行计算、网络计算机、多媒体技术、云计算技术和现代通信技术使人们开始采用和原来完全不同的方法进行工作。

如今，随着移动通信技术的快速发展和智能终端的普及，人们进入了移动互联网时代。移动互联网通过智能移动终端，采用移动无线通信方式获取移动通信网络服务和互联网服务，它包含终端层、软件层和应用层3个层面。其中，终端层包括智能手机、平板电脑、电子书阅读器、移动互联网设备（Mobile Internet Device, MID）等；软件层包括操作系统、中间件、数据库和安全软件等；应用层包括休闲娱乐类、工具媒体类、商务财经类等不同应用与服务。

光计算机、化学计算机、生物计算机和量子计算机等新一代计算机的研制发展，以及移动互联网的广泛应用等，必将给软件工程技术带来一场革命。



### 提示

移动通信是指利用无线通信技术，完成移动终端之间以及移动终端与固定终端之间的信息传送，即通信的双方，至少有一方处于运动中。

第3代移动通信系统（3rd-Generation, 3G）是指将无线通信与国际互联网等多媒体通信结合的移动通信系统。它能处理图像、语音、视频流等多种媒体形式，提供包括网页浏览、电话会议、电子商务等多种信息服务。

第4代移动通信系统（4th-Generation, 4G）是运用路由技术为主的网络架构，其网络结构应该是，核心网络不是专门用作移动通信，而是作为一种统一的网络，支持有线及无线的接入，它就像具有移动管理功能的固定网格（络），其接入点可以是有线或无线。不论何种接入方式，均具有统一的信令结构，信息的格式可以是ATM信元或IP分组（或包）。无线用户可在不同点接入，并可在通信时改变接入点。这样，核心网络必须能登记用户的位置，核査用户身份的鉴别以及在通信中的切换等。



## 1.1.2 软件危机

软件危机是指在计算机软件开发和维护时所遇到的一系列问题。软件危机主要包含两方面的问题：一是如何开发软件以满足社会对软件日益增长的需求；二是如何维护数量不断增长的已有软件。本节研究软件危机产生的原因、主要表现形式以及解决的途径。

### 1. 软件危机产生的原因

软件危机产生的原因与软件的特点有关，也与软件开发的方式、方法、技术和软件开发人员本身的素质有关。具体来说，有以下几点：

(1) 软件是计算机系统逻辑部件，软件产品往往规模庞大，给软件的开发和维护带来客观的困难。

(2) 软件一般要使用 5~10 年，在这段时间里，很可能出现开发时没有预料到的问题。例如，当系统运行的硬件、软件环境发生了变化时，系统需求发生了变化时，需要及时地维护软件，使软件可以继续使用。

(3) 软件开发技术落后，生产方式和开发工具落后。

(4) 软件开发人员忽视软件需求分析的重要性，对软件可维护性不重视，也是造成软件危机的原因之一。

### 2. 软件危机的主要表现形式

(1) 软件的发展速度跟不上硬件的发展和用户的需求。

计算机硬件成本逐年下降，软件应用日趋广泛，软件产品“供不应求”，与硬件成本相比，软件成本越来越高。

(2) 软件的成本和开发进度不能预先估计，用户不满意。

由于软件应用范围越来越广，很多应用领域往往是软件开发者不熟悉的，加之开发人员与用户之间信息交流不够，导致软件产品不符合要求，不能如期完成。因而，软件开发成本和进度都与原先的估计相差太大，引起用户不满。

(3) 软件产品质量差，可靠性不能保证。

软件质量保证技术没有应用到软件开发的全过程，导致软件产品质量问题频频发生。

(4) 软件产品可维护性差。

软件设计时不注意程序的可读性，不重视软件的可维护性，程序中存在的错误很难改正。软件需求发生变化时，维护相当困难。

(5) 软件没有合适的文档资料。

软件开发时文档资料不全或文档与软件不一致，会引起用户不满，同时也会给软件维护带来很大的困难。



### 3. 解决软件危机的途径

现代计算机普遍采用的是冯·诺依曼计算机体系结构，硬件的基本功能只是做简单的运算与逻辑判断，主要还是适用于数值计算。随着计算机应用的日益广泛，许多企事业单位的计算机，80%以上用于管理方面。管理方面大多为非数值计算问题，需要设计计算机软件来进行处理，因而可能会使软件变得复杂、庞大，从而导致软件危机的产生。

要解决软件危机问题，需要采用以下措施：

- (1) 使用好的软件开发技术和方法。
- (2) 使用好的软件开发工具，提高软件生产率。
- (3) 开发软件时有良好的组织、严密的管理，各方面人员相互配合共同完成任务。

为了解决软件危机，既要有技术措施（好的方法和工具），也要有组织管理措施。软件工程正是从技术和管理两方面来研究如何更好地开发和维护计算机软件的。

## 1.2 软件工程学

### 1.2.1 什么是软件

简单地讲，软件是计算机程序及其有关的数据和文档的完整集合。

#### 1. 软件

软件是指能够实现预定功能和性能的可执行的计算机程序，使程序正常运行所需要的数据，描述软件开发及其管理、程序的操作和使用的有关文档的完整集合。

其中，计算机程序是能够完成预定功能的可执行的指令序列；数据是程序能适当处理的信息，具有适当的数据结构，并且可以储存；软件文档（Software Documentation）是开发、使用和维护程序所需要的图文资料。

### 提示

国家标准 GB/T 11457—2006《信息技术 软件工程术语》对软件文档的定义为：软件文档是以人们可读的形式出现的技术数据和信息。包括计算机列表和打印输出，它们描述或规定软件设计或细节，说明软件具备的能力，或为使用软件以便从软件系统得到所期望的结果而提供的操作指令。



著名软件工程专家 B. W. Boehm 指出：“软件是程序以及开发、使用和维护程序所需要的所有文档。”特别是当软件成为商品时，文档更是必不可少的。没有文档仅有程序，是不能称为软件产品的。

## 2. 软件产品

软件产品是指向用户提供的计算机软件、信息系统或设备中嵌入的软件，或在提供计算机信息系统集成、应用服务等技术服务时提供的计算机软件。它是通过人的脑力劳动，通过知识、技术、环境信息的高度整合，所产生的逻辑产品。

软件产品以程序和文档的形式保存在作为计算机存储器的磁盘和光盘介质上，软件的功能和作用必须通过操作计算机才能体现，不能离开硬件和支撑环境而单独存在。

软件产品的生产主要是研制。软件产品的成本主要体现在软件的开发和研制上，软件开发研制完成后，通过复制就可以产生大量的软件产品，不需要再花费人力和物力。

软件产品不会用坏，不存在磨损、消耗等问题。但是随着时间的推移，软件产品会由于不适应新的软件生态环境而相对退化。

软件产品的生产主要是靠脑力劳动，还未完全摆脱手工开发方式，大部分产品是“定做的”。在软件产品的开发过程中，由于软件开发与用户的知识领域不同，需要大量的交流与沟通，并不断地修改需求，这会使软件开发费用不断增加。

软件产品在测试阶段难以发现的隐藏错误和缺陷，会在投入使用后不断暴露，因此需要不断地维护，“补丁”越来越多，维护成本也就越来越高。

## 3. 软件产品的类型

在信息社会，人们随时随地都会与各种各样的软件打交道。在家休息时，用水、用电会与供水、供电机构的各种计费软件产生关联；洗衣机、微波炉、电视机等很多家用电器都需要软件控制。出门在外时，从公共交通到手机通讯、从商场购物到休闲娱乐，处处都离不开软件的应用。

由于软件产品各式各样、种类繁多，要对其进行分类，并不是件容易的事情。按照不同的标准，会有不同的分类结果。比如，按软件产品的使用环境来区分，可以分为嵌入式软件、桌面级（PC）软件、大型（巨型）机软件、云计算软件等；按工作方式区分，可以分为实时软件、交互式软件、批处理式软件等；按使用对象区分，可以分为个人软件、商业通用软件、企业定制软件、工程及科学应用软件、人工智能软件等。

从软件工程学的角度来划分软件产品，可以分为基础软件、支撑软件和应用软件等。

- **基础软件**是指保证计算机运行并支撑应用软件运行的基础环境，包括操作系统、数据库、中间件。

