

高等院校软件工程专业规划教材

软件工程导论

免费提供



电子教案

陈明 编著



机械工业出版社
CHINA MACHINE PRESS



高等院校软件工程专业规划教材

软件工程导论

陈明 编著



机械工业出版社

本书是软件工程方面的教材,主要内容包括软件工程概述、可行性研究、需求分析、概要设计、详细设计、面向对象的分析与设计方法、编码、软件质量与质量保证、项目计划与管理、软件开发工具与环境概述、UML 与 Rose 等。本书的特点是内容系统、注重应用、面向实践、语言精练、概念准确。本书可作为高等院校软件工程课程的教材,也可作为软件工程人员的参考书。

图书在版编目(CIP)数据

软件工程导论/陈明编著. —北京:机械工业出版社,2009. 10

(高等院校软件工程专业规划教材)

ISBN 978 - 7 - 111 - 28382 - 9

I. 软… II. 陈… 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(2009)第 172703 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:唐德凯

责任印制:杨 曦

保定市中国画美凯印刷有限公司印刷

2010 年 1 月第 1 版·第 1 次印刷

184mm×260mm·19.25 印张·477 千字

0001 - 3000 册

标准书号:ISBN 978 - 7 - 111 - 28382 - 9

定价:33.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010)88361066

门户网:<http://www.cmpbook.com>

销售一部:(010)68326294

教材网:<http://www.cmpedu.com>

销售二部:(010)88379649

读者服务部:(010)68993821

封面无防伪标均为盗版

出版说明

计算机软件产业是发展迅速的朝阳产业，软件企业需要大量软件开发的专门技术人才。为了适应社会对软件人才的大量需求，许多高等院校相继开设了软件工程专业以培养专门的软件开发人才。软件工程的主要目标是开发系统模型，研究在有限预算下按时开发高质量软件的可靠技术。软件工程寻找计算机科学中科学与工程相结合，有效地开发和管理软件系统。在人才培养中，教学目标、教学计划、课程建设和教材建设异常重要。为了配合软件工程领域的教材建设，机械工业出版社邀请了高等学校从事软件工程的教学和研究的专家和教师，共同策划了适用于高校软件工程专业系列的教材，主要包括软件工程导论、需求工程、软件体系结构、软件测试技术、软件项目管理、UML 及应用等。

本系列教材的主要特色如下：

1. 从培养软件人才的需要出发，规化本系列教材。
2. 在内容的选取上，结合了软件工程的最新发展和实际应用，具有先进性和实用性。
3. 在教材的编写中，注重理论与实践的结合，注重引入案例，有助于实际能力的培养。
4. 在教材的体例上，各章内容具有一定的独立性，便于选择性学习。

本系列教材可作为软件工程专业、计算机科学与技术专业软件工程方向的教学用书，也可作为从事软件开发和软件工程领域工作的科技人员的参考书。

机械工业出版社

前 言

软件产业是创新的与充满活力的朝阳产业，需要大量的、不同层次的软件人才。计算机软件是软件产业的主要产品，并且计算机软件是逻辑产品、而不是物理产品。软件与硬件具有完全不同的特征。

计算机软件现已成为一种新的驱动力，是进行决策的引擎，是现代工程研究和解决问题的基础，在各种类型的应用系统中无所不在地广泛应用。

软件危机是指软件开发和维护过程中遇到的一系列严重问题，如如何开发软件、如何满足对软件的日益增长的需求、如何维护数量不断膨胀的已有软件等一系列问题。

为了克服与摆脱软件危机，诞生了软件工程学。从 1969 年提出软件工程概念以来，历经 30 多年的飞速发展，软件工程逐渐成熟，现已成为一门重要的学科。软件工程的目的是以保证软件生产的质量与效率为宗旨，研究一套科学的工程方法以及相应的方便的软件工具系统，用来指导和帮助软件的开发与研究，在软件的开发与研究起到重要的技术保障与促进作用。

软件工程是研究开发大型软件应用系统的学科，软件工程不仅覆盖了构建软件系统的相关技术层面的问题，还包括指导开发团队、安排进度及预算等管理层面的问题。软件工程不仅包括编写程序代码所涉及的技术，还包括所有对软件开发能够造成影响的问题。不存在任何单一开发技术或管理技术能够解决软件工程所面临的所有问题。因而，软件工程是包括一系列概念、理论、模式、语言、方法及工具的综合性学科。软件工程技术可分为产品实现技术及开发管理技术，产品实现技术主要涉及软件系统开发的相关问题，为实现软件产品提供支持；开发管理技术通常不针对某个软件开发项目，而是为管理和改进软件，组织所有的业务活动提供技术支持。

软件开发工具支持软件生存期中某一阶段的任务实现而使用的计算机程序。软件开发环境是一组相关的软件工具的集合，将它们集成在一起支持某种软件开发方法或某种软件开发模型。软件开发工具与环境是软件工程的重要组成部分，对于提高软件生产率，改进软件质量有越来越大的作用。

软件工程不是刻板的学科，需要丰富的想象力。软件工程是一门科学，它是有规律和步骤可循的。软件工程又是一个实践性极强的实用学科，在学习过程中，不仅要能掌握其理论原则与方法，更重要的是能熟练地应用。计算机科学与技术专业和相近专业的毕业生，有相当部分的同学要从事计算机软件开发和应用工作，通过软件工程的理论学习与实践，可以培养学生以软件工程的方法开发软件的习惯和素质，并在软件开发的工作中得以贯彻。因此，专业实践是软件工程本科教育的重要组成部分。

学习软件工程应注重计算思维能力、算法设计与分析能力、程序设计和实现能力、系统能力（系统的认知、设计、开发、应用）的培养，尤其是系统能力。系统能力包含两个层面上的含义：一方面是对一定规模的系统的全局掌控能力；另一方面是能够在构建系统时，系统地考虑问题的求解。

本书是软件工程导论教程,全书共 11 章,介绍了软件工程概述、可行性研究、需求工程、概要设计、详细设计、面向对象的分析与设计方法、编码、软件质量与质量保证、项目计划与管理、软件开发工具与环境板、UML 与 Rose 等内容。

本书在内容选择上,注重先进和实用,各章呈模块化,有助于快速掌握软件工程原则和方法。

由于作者水平有限,书中不足之处在所难免,敬请广大读者批评指正。

编 者

目 录

出版说明

前言

第 1 章 概述	1
1.1 软件	1
1.1.1 软件发展	1
1.1.2 软件定义	2
1.1.3 软件特点	2
1.1.4 软件分类	3
1.2 软件工程概念	6
1.2.1 软件危机与软件工程的定义	6
1.2.2 软件工程的基本内容与目标	8
1.2.3 软件工程的原则	8
1.3 软件生存周期与软件开发模型	9
1.3.1 软件生存周期	9
1.3.2 软件开发模型	10
1.4 小结	18
1.5 习题	18
第 2 章 可行性研究	19
2.1 可行性研究的任务	19
2.2 可行性研究的步骤	20
2.3 系统流程图	22
2.4 成本效益分析	23
2.4.1 成本估计	23
2.4.2 费用估计	24
2.4.3 度量效益的方法	25
2.5 小结	26
2.6 习题	26
第 3 章 需求工程	27
3.1 概述	27
3.1.1 软件需求定义	28
3.1.2 软件需求分类	28
3.1.3 需求规格说明	28
3.1.4 需求工程概念	30
3.1.5 需求工程过程	30
3.2 需求获取方法	31

3.3	需求分析的任务与原则	32
3.3.1	需求分析的任务	32
3.3.2	需求分析的原则	33
3.4	需求建模方法	34
3.4.1	结构化需求建模方法	34
3.4.2	数据流图	37
3.4.3	数据字典	43
3.5	需求图形工具	48
3.5.1	层次方框图	48
3.5.2	Warnier 图	48
3.5.3	IPO 图	49
3.6	需求验证	49
3.6.1	目的与任务	49
3.6.2	内容与方法	49
3.6.3	需求评审	50
3.7	需求管理	52
3.7.1	需求管理的目标	52
3.7.2	需求管理的原则	52
3.7.3	需求开发的管理	53
3.7.4	需求管理活动	54
3.8	小结	55
3.9	习题	56
第 4 章	概要设计	57
4.1	软件体系结构	57
4.1.1	概述	57
4.1.2	系统构成	59
4.1.3	控制模型	61
4.2	概要设计任务与步骤	62
4.2.1	概要设计任务	62
4.2.2	概要设计过程	62
4.3	软件设计的基本概念	64
4.3.1	模块化与模块独立性	64
4.3.2	抽象	68
4.3.3	结构设计原则	70
4.4	面向数据流的设计方法	73
4.4.1	基本概念	73
4.4.2	系统结构图的组成	74
4.4.3	变换分析	76
4.4.4	事务分析	79

4.4.5	设计优化	81
4.5	面向数据结构的分析设计方法	81
4.5.1	Jackson 系统开发方法	81
4.5.2	Warnier 方法	90
4.6	概要设计文档评审	91
4.7	小结	92
4.8	习题	92
第5章	详细设计	93
5.1	详细设计的任务与原则	93
5.1.1	详细设计的任务	93
5.1.2	详细设计的原则	93
5.2	详细设计工具	94
5.2.1	程序流程图	94
5.2.2	N-S 图	95
5.2.3	PAD 图	96
5.2.4	PDL	98
5.2.5	HIPO 图	99
5.2.6	详细设计工具的选择	100
5.3	详细设计规格说明与复审	101
5.3.1	详细设计说明书	101
5.3.2	设计复审	102
5.4	小结	102
5.5	习题	102
第6章	面向对象设计	104
6.1	面向对象方法	104
6.1.1	面向对象方法概述	104
6.1.2	面向对象的软件工程	105
6.1.3	面向对象的基本概念和特征	109
6.2	面向对象的分析	116
6.2.1	面向对象分析基本过程及原则	116
6.2.2	确定对象与类	120
6.2.3	确定属性	123
6.2.4	定义服务	124
6.2.5	对象间通信	127
6.3	面向对象设计	133
6.3.1	面向对象设计的基本概念	133
6.3.2	面向对象设计的方法	134
6.4	实时软件设计	137
6.4.1	系统设计	138

6.4.2	数据采集系统	139
6.5	软件复用	139
6.5.1	软件复用概念	140
6.5.2	基于组件的开发	141
6.6	用户界面的设计	143
6.6.1	设计原则	143
6.6.2	用户交互	144
6.6.3	信息表示	145
6.6.4	用户支持	146
6.6.5	界面评价	147
6.7	小结	147
6.8	习题	147
第7章	编码	149
7.1	程序设计语言	149
7.1.1	程序设计语言分类	149
7.1.2	程序设计语言的特点	151
7.1.3	程序设计语言的选择	153
7.2	编码风格	155
7.2.1	源程序文档化	155
7.2.2	数据说明	157
7.2.3	语句结构	157
7.2.4	输入/输出	160
7.3	程序效率	160
7.3.1	程序效率准则	160
7.3.2	算法对效率的影响	161
7.3.3	影响存储器效率的因素	161
7.3.4	影响输入/输出的因素	161
7.4	编程安全	162
7.4.1	冗余程序设计	162
7.4.2	防错程序设计	163
7.5	面向对象程序设计步骤	164
7.6	编码优化	164
7.6.1	优化的内容与原则	164
7.6.2	程序结构优化	165
7.6.3	程序代码优化	165
7.7	小结	165
7.8	习题	166
第8章	软件质量与质量保证	167
8.1	软件质量的定义	167

8.2	影响软件质量的因素	167
8.3	软件质量保证策略	168
8.4	软件质量保证活动	169
8.5	软件评审	170
8.5.1	设计质量的评审内容	171
8.5.2	程序质量的评审内容	175
8.6	软件质量保证的标准	177
8.7	结构化程序的测试	178
8.7.1	软件测试的目的	179
8.7.2	软件测试的原则	179
8.7.3	软件测试的对象	180
8.7.4	软件测试的基本过程	180
8.7.5	软件测试技术	181
8.7.6	设计测试方案	182
8.7.7	测试的步骤	184
8.7.8	软件纠错技术	188
8.8	面向对象的软件测试	190
8.8.1	面向对象分析和面向对象设计的模型测试	190
8.8.2	面向对象的测试策略	192
8.8.3	面向对象软件测试集设计	193
8.9	测试计划与测试分析报告	195
8.10	软件维护	196
8.10.1	软件维护分类与特点	196
8.10.2	软件维护步骤	199
8.10.3	软件的可维护性	201
8.10.4	软件维护的副作用	202
8.10.5	逆向工程和再生工程	203
8.11	小结	205
8.12	习题	205
第9章	项目计划与管理	206
9.1	软件项目特点及软件管理功能	206
9.1.1	软件项目的特点	206
9.1.2	软件管理的功能	207
9.1.3	确定软件项目的工作范围	207
9.2	确定软硬件资源	207
9.2.1	人力资源	207
9.2.2	硬件	208
9.2.3	软件	208
9.3	人员的计划和组织	209

9.4	成本估计及控制	210
9.4.1	软件开发成本估计方法	210
9.4.2	专家估算法	210
9.4.3	成本估算模型	211
9.5	进度计划	213
9.5.1	软件工作的特殊性	213
9.5.2	各阶段工作量的分配	214
9.5.3	制定开发进度	214
9.6	软件配置管理	215
9.6.1	基线	215
9.6.2	软件配置项	216
9.6.3	软件配置管理过程	218
9.7	软件管理方案	221
9.8	小结	221
9.9	习题	222
第10章	软件开发工具与环境	223
10.1	软件工具概述	223
10.2	软件开发工具的功能	223
10.3	软件开发工具的特性	224
10.4	软件开发工具的分类	226
10.5	软件开发环境	227
10.6	常用开发环境	228
10.6.1	Windows 98 开发环境	229
10.6.2	Windows NT 开发环境	233
10.6.3	Linux 开发环境	236
10.6.4	UNIX 程序开发环境介绍	240
10.7	软件开发工具的发展	244
10.8	CASE 技术	245
10.9	小结	247
10.10	习题	247
第11章	UML 与 Rose	248
11.1	UML 概述	248
11.1.1	UML 的定义	248
11.1.2	UML 的组成	248
11.1.3	UML 的应用领域	253
11.2	UML 的表示法	254
11.2.1	概述	254
11.2.2	用例图	257
11.2.3	类图	260

11.3	UML 软件开发过程概述	266
11.3.1	初始阶段	267
11.3.2	细化阶段	267
11.3.3	细化阶段结束	268
11.3.4	构造阶段	268
11.3.5	移交阶段	269
11.4	Rational Rose 概述	269
11.4.1	Rose 的内容	269
11.4.2	可视化设计	269
11.4.3	建模的概念	270
11.4.4	使用 Rose 实现建模	270
11.4.5	Rational Rose 系列产品	271
11.5	小结	272
11.6	习题	272
附录 A	可行性研究报告规范	273
附录 B	项目开发计划规范	277
附录 C	需求规格说明书规范	280
附录 D	概要设计说明书规范	284
附录 E	详细设计说明书规范	287
附录 F	测试计划	290
附录 G	测试分析报告	292
附录 H	项目开发总结报告规范	294
参考文献	296

第1章 概述

1.1 软件

软件是一种逻辑产品，也是开发和运行产品的载体。作为一种产品，它表达了由计算机硬件体现的计算潜能。不论它是驻留在设备中，还是在主机中，软件都是一个信息转换器，能够产生、管理、获取、修改、显示或转换信息。这些信息可以很简单，也可以很复杂，如多媒体信息。作为开发运行产品的载体，软件是计算机工作和信息通信的基础，也是创建和控制其他程序的基础。

信息是21世纪最重要的产品，软件充分体现了这一点。通过软件处理数据，凸显了数据的重要性；软件管理商业信息，增强了商业竞争力。软件不仅提供了通往全球信息网络的途径，而且也提供了获取信息的多种手段。

1.1.1 软件发展

1. 程序设计阶段

计算机发展的早期阶段（20世纪50年代初期~20世纪60年代中期）为程序设计阶段。在这个阶段硬件已经通用化，而软件的生产却是个体化的。这时，由于程序规模小，几乎没有什么系统化的方法可遵循。对软件的开发没有任何管理方法，一旦计划推迟了或者成本提高了，程序员才开始弥补。在通用的硬件已经非常普遍的时候，软件产品还处在初级阶段，对每一类应用均需自行再设计，应用范围很有限。设计往往仅是人们头脑中的一种模糊想法，而文档根本不存在。

2. 程序系统阶段

计算机系统发展的第二阶段（20世纪60年代中期~20世纪70年代末期）为程序系统阶段。多道程序设计、多用户系统引入了人机交互的新概念。交互技术打开了计算机应用的新世界，硬件和软件配合达到了一个新层次，出现了实时系统和第一代数据库管理系统。这个阶段的另一个特点就是软件产品的使用和“软件作坊”的出现。被开发的软件可以在较宽广的范围中应用。主机和微机上的程序能够有数百甚至上千的用户。

在软件的使用过程中，当发现错误时，当用户需求和硬件环境发生变化时都需要修改软件，这些活动统称为软件维护。在软件维护上的花费以惊人的速度增长。更为严重的是，许多程序的个性化特性使得它们根本不能维护。“软件危机”出现了。

3. 软件工程阶段

计算机系统发展的第三阶段始于20世纪70年代中期并跨越了近10年，被称为软件工程阶段。在这一阶段，以软件的产品化、系列化、工程化、标准化为特征的软件产业发展起来，打破了软件生产的个体化特征，有了可以遵循的软件工程化的设计原则、方法和标准。在分布式系统中，各台计算机同时执行某些功能，并与其他计算机通信，极大地提高了计算

机系统的复杂性。广域网、局域网、高带宽数字通信以及对即时数据访问需求的增加都对软件开发者提出了更高的要求。

计算机系统发展的第四阶段已经不再着重于单台计算机和计算机程序，而是针对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机、广域网络和局域网络，配以先进的软件应用已成为标准。计算机体系结构迅速地从集中的主机环境转变为分布的客户机/服务器环境。世界范围的信息网提供了一个基本结构，信息高速公路和网际空间连通已成为令人关注的热点问题。事实上，Internet 可以看做是能够被单个用户访问的软件，计算机发展正朝着社会信息化和软件产业化方向发展，从技术的软件工程阶段过渡到社会信息化的计算机系统阶段。随着第四阶段的进展，一些新技术开始涌现。面向对象技术将在许多领域中迅速取代传统软件开发方法。表 1-1 给出了计算机发展 4 个阶段的典型技术。

表 1-1 4 个阶段的典型技术

阶 段	第一 阶段	第二 阶段	第三 阶段	第四 阶段
典型技术	<ul style="list-style-type: none"> ● 面向批处理 ● 有限的分布 ● 自定义软件 	<ul style="list-style-type: none"> ● 多用户 ● 实时 ● 数据库 ● 软件产品 	<ul style="list-style-type: none"> ● 分布式系统 ● 嵌入“智能” ● 低成本硬件 ● 消费者的影响 	<ul style="list-style-type: none"> ● 强大的桌面系统 ● 面向对象技术 ● 专家系统 ● 人工神经网络 ● 并行计算 ● 网络计算

1.1.2 软件定义

计算机系统是通过运行程序来实现各种不同应用的。把各种不同功能的程序，包括用户为自己的特定目的编写的程序、检查和诊断机器系统的程序、支持用户应用程序运行的系统程序、管理和控制机器系统资源的程序等，通常称为软件。它是计算机系统中与硬件相互依存的另一部分，与硬件合为一体完成系统功能。软件定义如下：

- 1) 在运行中能提供所希望的功能和性能的指令集（即程序）。
- 2) 使程序能够正确运行的数据结构。
- 3) 描述程序研制过程和方法所用的文档。

随着计算机应用的日益普及，软件变得越来越复杂，规模也越来越大，从而使人与人、人与机器间相互沟通，保证软件开发与维护工作的顺利进行显得特别重要，因此，文档（即各种报告、说明、手册的总称）是不可缺少的。特别是在软件日益成为产品的今天，文档的作用就更加重要。

1.1.3 软件特点

软件在整个计算机系统中是一个逻辑部件，而硬件是一个物理部件。因此，软件相对硬件而言有许多特点。为了能全面、正确地理解计算机软件及软件工程的重要性，必须了解软件的特点。软件的特点可归纳如下：

- 1) 软件是一种逻辑实体，而不是具体的物理实体，因而它具有抽象性。这个特点使它

件的形态，而必须通过测试、分析、思考、判断去了解它的功能、性能及其他特性。

2) 软件是通过人们的智力活动，把知识与技术转化成信息的一种产品，是在研制、开发中被创造出来的。一旦某一软件项目研制成功，以后就可以大量地复制同一内容的副本，这就是软件的生产过程。也就是说软件的研制成本远远大于其生产成本。软件故障往往是在开发时产生而在测试时没有被发现的问题，要保证软件的质量，必须着重于软件开发过程，加强管理。

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。软件维护比硬件维护要复杂得多，与硬件的维修有着本质的差别，参阅图 1-1、图 1-2 和图 1-3。图 1-1 所示的是硬件的故障率随时间变化的曲线，如图 1-2 所示的是在理想情况下软件故障率随时间变化的曲线，如图 1-3 所示的是软件的实际故障率曲线。

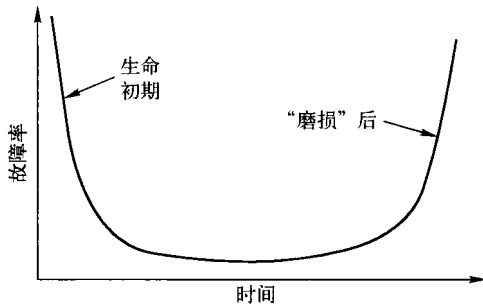


图 1-1 硬件的故障率随时间变化的曲线

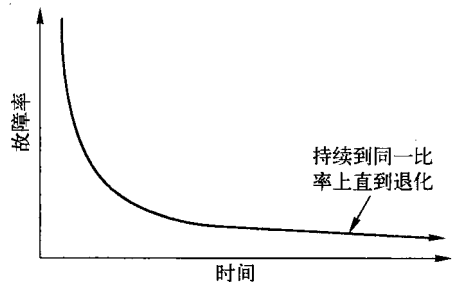


图 1-2 理想情况下的软件故障率随时间变化的曲线

4) 软件的开发和运行经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。为了解除这种依赖，在软件开发中提出了软件移植的问题，并且把软件的可移植性作为衡量软件质量的因素之一。

5) 软件的开发尚未完全摆脱手工开发方式。由于传统的手工开发方式仍然占据统治地位，软件开发的效率受到很大的限制。因此，应促进软件技术的进展，提出和采用新的开发方法。例如，近年来出现的充分利用现有软件的复用技术、自动生成技术和其他一些有效的软件开发工具或软件开发环境，既方便了软件开发的质量控制，又提高了软件开发的效率。

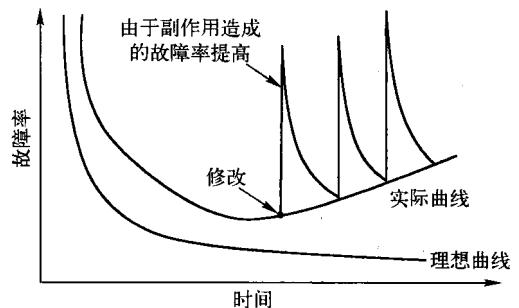


图 1-3 软件的实际故障率曲线

6) 软件的开发费用越来越高。软件的研制工作需要投入大量的、复杂的、高强度的脑力劳动，需要较高的成本。

7) 软件的开发是一个复杂的过程，例如，银行管理系统涉及安全等问题，因而管理是软件开发过程中必不可少的内容。

1.1.4 软件分类

人们在工作和学习中，经常接触到各式各样的软件。那么这些数量众多的软件究竟分为哪些类型，这就要考虑对计算机软件进行分类的依据。但事实上，要给出一个科学的、统一

的、严格的计算机分类标准是不现实的。但对软件的类型进行必要的划分，对于根据不同类型的工程对象采用不同的开发和维护方法是很有价值的，因此有必要从不同角度对计算机软件做适当的分类。

1. 基于软件功能的划分

1) 系统软件：与计算机硬件紧密配合以使计算机各个部件与相关软件及数据协调、高效工作的软件，如操作系统、数据库管理系统等。系统软件在工作时频繁地与硬件交互，以便为用户服务，共享系统资源，在这中间伴随着复杂的进程管理和复杂的数据结构的处理。系统软件是计算机系统必不可少的重要组成部分。

2) 支撑软件：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。可分为以下几类。

- 一般类型：包括文本编辑程序、文件格式化程序、程序库系统等。
- 支持需求分析：包括 PSL/PSA 问题描述语言、问题描述分析器、关系数据库系统、一致性检验程序等。
- 支持设计：包括图形软件包、结构化流程图绘图程序、设计分析程序、程序结构图编辑程序等。
- 支持实现：包括编译程序、交叉编译程序、预编译程序、连接编译程序等。
- 支持测试：包括静态分析程序、符号执行程序、模拟程序、测试覆盖检验程序等。
- 支持管理：包括 PERT 进度计划评审方法、绘图程序、标准检验程序和库管理程序等。

3) 应用软件：在特定领域内开发、为特定目的服务的一类软件。现在几乎所有的国民经济领域都使用了计算机，为这些计算机应用领域服务的应用软件种类繁多。其中商业数据处理软件是占比例最大的一类，工程与科学计算软件大多属于数值计算问题。应用软件还包括计算机辅助设计/计算机辅助制造 (CAD/CAM)、系统仿真、智能产品嵌入软件 (如汽车油耗控制、仪表盘数字显示、刹车系统)，以及人工智能软件 (如专家系统、模式识别) 等，此外，在事务管理、办公自动化、中文信息处理、计算机辅助教学 (CAI) 等方面的软件也得到迅速发展，产生了惊人的生产效率和巨大的经济效益。

2. 基于软件工作方式划分

1) 实时处理软件：指在事件或数据产生时，立即处理，并及时反馈信号，控制需要监测和控制过程的软件，主要包括数据采集、分析、输出 3 部分，其处理时间是应严格限定的，如果在任何时间超出了这一限制，都将造成事故。

2) 分时软件：允许多个联机用户同时使用计算机的软件。系统把处理机时间轮流分配给各联机用户，使各用户都感到只有自己在使用计算机。

3) 交互式软件：能实现人机通信的软件。这类软件接收用户给出的信息，但在时间上没有严格的限定，这种工作方式给予用户很大的灵活性。

4) 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，按顺序逐个处理的软件。

3. 基于软件规模的划分

根据开发软件所需的人力、时间以及完成的源程序行数，可划分为下述 6 种不同规模的软件。