

软件工程 教程

陈明 编著

科学出版社

软件工程教程

陈 明 编著

科学出版社

2002

内 容 简 介

本书主要介绍了软件工程学及应用的有关内容, 主要包括可行性研究、需求分析、概要设计、详细设计、编码、测试、维护和有关软件开发工具与软件开发环境方面的内容。为了保持教材内容的先进性, 本书还介绍了面向对象软件工程学方面的内容和较流行的 PowerDesigner 软件开发工具。

本书可作为高等院校相关专业的教材, 也可作为从事软件开发与应用的工程技术人员参考书。

图书在版编目 (CIP) 数据

软件工程学教程/陈明编著. —北京: 科学出版社, 2002

ISBN 7-03-010069-7

I. 软… II. 陈… III. 软件工程-教材 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2002) 第 010108 号

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

新蕾印刷厂 印刷

科学出版社总发行 各地新华书店经销

*

2002年3月第一版 开本: 787×1092 1/16

2002年3月第一次印刷 印张: 27

印数: 1—4 000 字数: 637 000

定价: 39.00 元

(如有印装质量问题, 我社负责调换〈路通〉)

前 言

为了克服与摆脱软件危机，人们希望通过工程技术方法和管理手段使软件开发纳入工程化的轨道，由此诞生了软件工程学。从 20 世纪 60 年代末提出软件工程概念以来，历经 30 多年的飞速发展，软件工程学逐渐成熟，现已成为计算机科学与技术领域中的一门重要的学科。软件工程学的目标是以提高软件生产的质量与效率为宗旨，研究一套科学的工程方法，以及与此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工作，并起到技术保障与促进作用。

软件开发工具是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，它们组织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、改进软件质量起着越来越大的作用。

本书系统地介绍了软件工程学的内容，主要包括：

- (1) 软件工程概述
- (2) 可行性研究
- (3) 需求分析
- (4) 概要设计
- (5) 详细设计
- (6) 面向对象的分析和设计方法
- (7) 编码
- (8) 软件质量与质量保证
- (9) 项目计划与管理
- (10) 软件开发工具与环境概述
- (11) 数据库模型设计工具软件 PowerDesigner
- (12) 课程实验

本书内容较为系统，注重基本概念的解释和方法的说明，各章都附有内容小结和习题。由于作者水平有限，书中不足之处在所难免，敬请读者批评指正。

陈 明

目 录

第一章 软件工程概述	1
1.1 软件	1
1.1.1 软件的发展	1
1.1.2 软件的定义	2
1.1.3 软件的特点	2
1.1.4 软件分类	4
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	9
1.2.3 软件工程的原则	9
1.3 软件生存周期与软件开发模型	10
1.3.1 软件生存周期	10
1.3.2 软件开发模型	11
小结	19
习题一	19
第二章 可行性研究	20
2.1 可行性研究的任务	20
2.2 可行性研究的步骤	21
2.3 系统流程图	22
2.4 成本效益分析	24
2.4.1 成本估计	24
2.4.2 费用估计	25
2.4.3 几种度量效益的方法	26
小结	27
习题二	27
第三章 需求分析	28
3.1 需求分析的任务与步骤	28
3.1.1 需求分析的任务	28
3.1.2 需求分析的步骤	30
3.1.3 需求分析的原则	33
3.2 需求分析的方法	34
3.2.1 面向数据流的需求分析方法	35
3.2.2 数据流图	36
3.2.3 数据字典	43

3.3 图形工具.....	48
3.3.1 层次方框图	48
3.3.2 Warnier 图	49
3.3.3 IPO 图	49
3.4 需求规格说明与评审	50
3.4.1 需求规格说明的主要内容	50
3.4.2 需求分析的评审	51
3.4.3 需求分析评审的方法	51
小结	52
习题三	52
第四章 概要设计	54
4.1 概要设计的任务与步骤.....	54
4.1.1 概要设计任务	54
4.1.2 概要设计的过程	54
4.2 软件设计的概念与原则.....	56
4.2.1 模块化	56
4.2.2 抽象与逐步求精	57
4.2.3 信息隐蔽和局部化	59
4.2.4 模块独立性	59
4.2.5 结构设计原则	64
4.3 面向数据流的设计方法.....	67
4.3.1 基本概念	68
4.3.2 系统结构图的组成	69
4.3.3 变换分析	70
4.3.4 事务分析	74
4.3.5 设计优化	76
4.4 面向数据结构的分析设计方法.....	76
4.4.1 Jackson 系统开发方法.....	76
4.4.2 Warnier 方法	86
4.5 概要设计文档评审.....	87
小结	88
习题四	88
第五章 详细设计	89
5.1 详细设计的任务与原则.....	89
5.1.1 详细设计的任务	89
5.1.2 详细设计的原则	89
5.2 详细设计的方法.....	90
5.2.1 程序流程图	90
5.2.2 N-S 图	92

5.2.3	PAD 图	93
5.2.4	PDL	95
5.2.5	HIPO 图	96
5.2.6	详细设计工具的选择	97
5.3	详细设计规格说明与复审	98
5.3.1	详细设计说明书	98
5.3.2	设计复审	99
小结		99
习题五		100
第六章	面向对象的分析和设计方法	101
6.1	面向对象方法的基本概念和特征	101
6.1.1	面向对象方法概述	101
6.1.2	面向对象的软件工程 (OOSE)	103
6.1.3	面向对象的基本概念和特征	107
6.2	面向对象的分析	115
6.2.1	面向对象分析基本过程及原则	115
6.2.2	确定对象、类	119
6.2.3	确定属性	122
6.2.4	定义服务	123
6.2.5	对象间通信	126
6.3	面向对象设计	132
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	UML 方法	137
6.4.1	UML 的发展	137
6.4.2	UML 的表示法	141
6.4.3	UML 软件开发过程概述	154
小结		156
习题六		156
第七章	编码	157
7.1	程序设计语言	157
7.1.1	程序设计语言分类	157
7.1.2	程序设计语言的特点	159
7.1.3	程序设计语言的选择	161
7.2	编码风格	163
7.2.1	源程序文档化	164
7.2.2	数据说明	166
7.2.3	语句结构	166
7.2.4	输入/输出 (I/O)	169

7.3 程序效率	169
7.3.1 有关程序效率的几条准则	169
7.3.2 算法对效率的影响	170
7.3.3 影响存储器效率的因素	170
7.3.4 影响输入/输出的因素	170
7.4 编程安全	171
7.4.1 冗余程序设计	171
7.4.2 防错程序设计	172
7.5 面向对象程序设计步骤	173
小结	173
习题七	174
第八章 软件质量与质量保证	175
8.1 软件质量的定义	175
8.2 影响软件质量的因素	175
8.3 软件质量保证策略	176
8.4 软件质量保证活动	177
8.5 软件评审	178
8.5.1 设计质量的评审内容	179
8.5.2 程序质量的评审内容	184
8.6 软件质量保证的标准	186
8.7 结构化的软件测试	188
8.7.1 软件测试的概念和原则	188
8.7.2 软件测试技术	190
8.7.3 测试的步骤	197
8.7.4 软件纠错技术	201
8.8 面向对象的软件测试	203
8.8.1 面向对象分析和面向对象设计的模型测试	203
8.8.2 面向对象的测试策略	205
8.8.3 面向对象软件测试集设计	206
8.9 软件测试计划与测试分析报告	208
8.10 软件维护	210
8.10.1 软件维护的定义、分类、特点	210
8.10.2 软件维护步骤及组织	212
8.10.3 软件的可维护性	216
8.10.4 软件维护的副作用	217
8.10.5 逆向工程和再生工程	218
小结	219
习题八	220
第九章 项目计划与管理	221

9.1 软件项目的特点与软件管理功能	221
9.1.1 软件项目的特点	221
9.1.2 软件管理的功能	222
9.1.3 确定软件项目的工作范围	222
9.2 确定软硬件资源	222
9.2.1 人力资源	223
9.2.2 硬件	223
9.2.3 软件	223
9.3 人员的计划和组织	224
9.4 成本估计及控制	225
9.4.1 软件开发成本估计方法	225
9.4.2 专家判定技术	226
9.4.3 成本估算模型	226
9.5 进度计划	229
9.5.1 软件工作的特殊性	229
9.5.2 各阶段工作量的分配	229
9.5.3 制定开发进度	230
9.6 软件配置管理	231
9.6.1 基线	231
9.6.2 软件配置项	232
9.6.3 软件配置管理过程	234
9.7 软件管理方案	237
小结	238
习题九	238
第十章 软件开发工具与环境概述	239
10.1 软件开发工具概述	239
10.2 软件开发工具的功能	239
10.3 软件开发工具的特性	240
10.4 软件开发工具的分类	241
10.5 软件开发环境	243
10.6 常用开发环境	245
10.6.1 Windows 98 开发环境	245
10.6.2 Windows NT 开发环境	249
10.6.3 Linux 开发环境	252
10.6.4 Unix 程序开发环境	257
10.7 软件开发环境与工具的研究、应用及发展	261
10.8 CASE 技术	262
小结	264
习题十	264

第十一章 PowerDesigner	265
11.1 PowerDesigner 6.0 简介	265
11.1.1 PowerDesigner 6.0 概述	265
11.1.2 PowerDesigner 6.0 的模块组成	265
11.1.3 PowerDesigner 6.0 的模型和对象特性	267
11.2 ProcessAnalyst	267
11.2.1 概述	267
11.2.2 PA 应用实例	272
11.3 DataArchitect	290
11.3.1 DataArchitect 概述	290
11.3.2 DataArchitect 的概念数据模型 (CDM)	297
11.3.3 DataArchitect 的物理数据模型	304
小结	309
习题十一	309
第十二章 课程实验	310
12.1 实验一：撰写可行性研究报告	310
12.2 实验二：撰写需求规格说明书	315
12.3 实验三：撰写概要设计说明书	322
12.4 实验四：撰写详细设计说明书	330
12.5 实验五：UML 的使用	347
12.6 实验六：使用 Word 编写软件工程文档	366
12.7 实验七：熟悉 PowerDesigner 环境	376
附录	398
附录 1 可行性研究报告规范	398
附录 2 项目开发计划规范	402
附录 3 需求规格说明书规范	405
附录 4 概要设计说明书规范	409
附录 5 详细设计说明书规范	412
附录 6 项目开发总结报告规范	414
附录 7 用 Word 撰写文档规范	417
参考文献	421

第一章 软件工程概述

1.1 软 件

软件是一种产品，同时又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。不管它是驻留在设备中，还是在主机中，软件是一个信息转换器，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。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，软件充分地体现了这一点。软件处理数据，使得这些数据更为有用；软件管理商业信息，更增强了商业竞争力。它不仅提供了通往全球信息网络的途径，而且也提供了以各种形式获取信息的手段。

1.1.1 软件的发展

1. 程序设计阶段

计算机发展的早期阶段（20 世纪 50 年代初期至 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 年代中期到 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，使硬件和软件的配合达到了一个新层次，实时系统和第一代数据库管理系统出现了。这个阶段还有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，当发现程序错误时，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，许多程序的个体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十年，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。

4. 第四阶段

计算机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 可以看作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。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 给出了计算机发展四个阶段的典型技术。

表 1-1 四个阶段的典型技术

阶段	第一阶段	第二阶段	第三阶段	第四阶段
典型技术	- 面向批处理 - 有限的分布 - 自定义软件	- 多用户 - 实时 - 数据库 - 软件产品	- 分布式系统 - 嵌入“智能” - 低成本硬件 - 消费者的影响	- 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 - 网络计算机

1.1.2 软件的定义

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：

- 在运行中能提供所希望的功能和性能的指令集（即程序）；
- 使程序能够正确运行的数据结构；
- 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，这就使得人与人、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 软件的特点

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。这个特点使它与计算机硬件或其他工程对象有着明显的差别。人们可以把它记录在介质上，但却无法看到软件的形态，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，即其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，所以要保证软件的质量，必须着重于软件开发过程，加强管理。

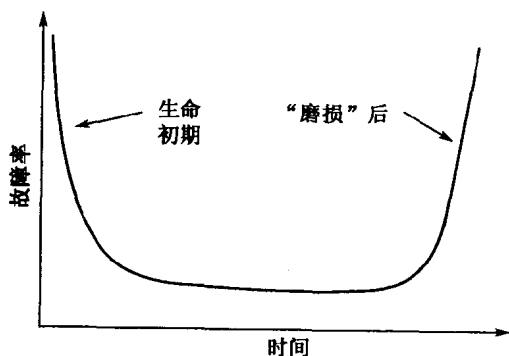


图 1-1 硬件的故障率随时间变化的曲线

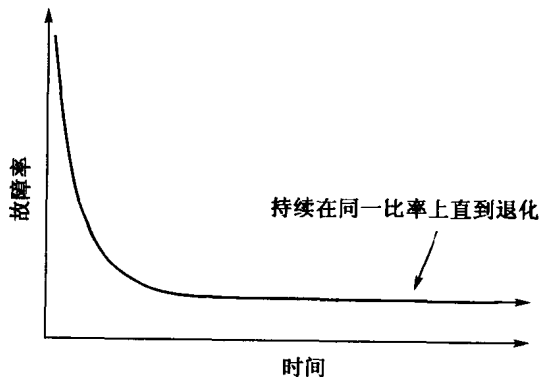


图 1-2 理想情况下的软件故障率随时间变化的曲线

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，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，图 1-3 所示的是软件的实际故障率曲线。

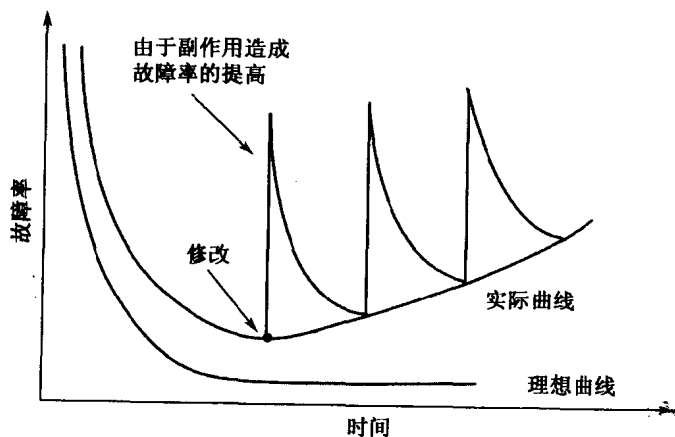


图 1-3 软件的实际故障率曲线

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的开发方式。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。

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到安全等问题, 因而管理是软件开发过程中必不可少的内容。

1.1.4 软件的分类

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、严格的计算机软件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: 是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。例如, 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往, 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: 它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。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: 是在特定领域内开发、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/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. 基于软件工作方式的划分

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、输出三部分，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，可划分为下述六种不同规模的软件。

1) 微型软件：指一个人在几天之内完成的、程序不超过 500 行语句且仅供个人专用的软件。通常这类软件没有必要做严格的分析，也不必要有完整的设计、测试资料。

2) 小型软件：指一个人在半年之内完成的 2 千行以内的程序。这种程序通常没有与其他程序的接口，但需要按一定的标准化技术、正规的资料书写以及定期的系统审查，只是没有大型软件那样严格。

3) 中型软件：指 5 个人以内一年多时间里完成的 5 千到 5 万行的程序。中型软件开始出现了软件人员之间、软件人员与用户之间的联系和协调的配合关系问题，因而计划、资料书写以及技术审查需要比较严格地进行。在开发中使用系统的软件工程方法是完全必要的，这对提高软件产品质量和程序开发人员的工作效率起着重要的作用。

4) 大型软件：指 5 至 10 个人在两年多的时间里完成的 5 万到 10 万行的程序。参加工作的软件人员需要进行二级管理。对于这样规模的软件，采用统一的标准、实行严格的审查是绝对必要的。由于软件的规模庞大以及问题的复杂性，往往在开发过程中会出现一些事先难以估计的不测事件。

5) 甚大型软件：指 100 至 1000 人在四五年时间里完成的具有 100 万行的程序。这种甚大型项目可能会划分成若干个子项目，每一个子项目都是一个大型软件。子项目之间具有复杂的接口。例如，实时处理系统、远程通信系统、多任务系统、大型操作系统、大型数据库管理系统。很显然，如果这类问题没有软件工程方法的支持，它的开发工作是不可想象的。

6) 极大型软件：指 2000 至 5000 人在 10 年内完成的具有 1000 万行以内的程序。

这类软件很少见，往往用于军事指挥、弹道导弹防御系统等。

可以看出，规模大、时间长、多人参加的软件项目，其开发工作必须要有软件工程的知识做指导，而规模小、时间短、参加人员少的软件项目的开发也得遵循软件工程的概念和开发规范，其基本原则是一样的。

4. 基于软件失效的影响进行划分

工作在不同领域的软件，在运行中对可靠性也有不同的要求。事实上，随着计算机进入国民经济各个重要领域，软件的可靠性越来越重要。人们一般称这类软件为关键软件，其特点在于：

- 可靠性质量要求高；
- 常与完成重要功能的大系统的处理部件相连；
- 含有的程序可能对人员、公众、设备或设施的安全造成影响，还可能影响到环境的质量和国家安全。

5. 基于软件服务对象的范围进行划分

软件工程项目完成后可以有两种结果：

1) 定制软件，是受某个特定客户（或少数客户）的委托，由一个或多个软件开发机构在合同的约束下开发出来的软件。

2) 产品软件，是由软件开发机构开发出来直接提供给市场，或是为众多用户服务的软件。

1.2 软件工程概念

由于微电子学技术的进步，计算机硬件性能有了很大的提高，而且质量也稳步提高。然而，计算机软件成本却不断上升，质量也不尽人意，软件开发的生产率也远远不能满足计算机应用的要求。软件已经成为限制计算机系统进一步发展的关键因素。

更为严重的是计算机系统发展早期所形成的一系列错误概念和做法，已经严重地阻碍了计算机软件的开发，甚至有的大型软件根本无法维护，只能提前报废，造成大量人力、物力的浪费，从而导致软件危机。为了研究解决软件危机的方法，计算机科学技术领域中一门新兴的学科逐步形成了，这就是计算机软件工程学。

1.2.1 软件危机与软件工程的定义

1. 软件危机

软件危机指的是软件开发和维护过程中遇到的一系列严重问题。软件危机包含下述两方面的问题：如何开发软件，怎样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。具体地说，软件危机主要有下列表现：

- 产品不符合用户的实际需要。
- 软件开发生产率提高的速度远远不能满足客观需要，软件的生产率远远低于硬

件生产率和计算机应用的增长,使人们不能充分利用现代计算机硬件提供的巨大潜力。

- 软件产品的质量差。软件可靠性和质量保证的定量概念刚刚出现不久,软件质量保证技术(审查、复审和测试)没有贯穿到软件开发的全过程,这些都导致软件产品发生质量问题。
- 对软件开发成本和进度的估计常常不准确。实际成本比估计成本有可能高出一个数量级,实际进度比预期进度拖延几个月甚至几年。这种现象降低了软件开发者的信誉。而为了赶进度和节约成本所采取的一些权宜之计又往往降低了软件产品的质量,从而不可避免地会引起用户的不满。
- 软件的可维护性差。很多程序中的错误是难以改正的,实际上不能使这些程序适应硬件环境的改变,也不能根据用户的需要在原有程序中增加一些新的功能。没能实现软件的可重用,人们仍然在重复开发功能类似的软件。
- 软件文档资料通常既不完整也不合格。
- 软件的价格昂贵,软件成本在计算机系统总成本中所占的比例逐年上升。

2. 软件工程的定义

首先,采用工程化方法和途径来开发与维护软件。软件开发应是一种组织良好、管理严密、各类人员协同配合、共同完成的工程项目,必须充分吸取和借鉴人类长期以来从事各种工程项目所积累的行之有效的原理、概念、技术和方法,应该推广使用在实践中总结出来的软件开发的成功技术和方法,并且研究探索更好更有效的技术和方法,尽快消除在计算机系统早期发展阶段形成的一些错误概念和做法。将软件的生成在时间上分成若干阶段以便于分步而有计划地分工合作,在结构上简化若干逻辑模块。把软件作为工程产品来处理,按计划、分析、设计、实现、测试、维护的周期来进行生产。

其次,应该开发和使用更好的软件工具。在软件开发的每个阶段都有许多繁琐重复的工作需要做,在适当的软件工具辅助下,开发人员可以把这类工作做得既快又好。如果把各个阶段使用的软件工具有机地集成成一个整体,支持软件开发的全过程,则称为软件工程支撑环境。

最后,采取必要的管理措施。软件产品是把思维、概念、算法、组织、流程、效率、质量等多方面问题融为一体的产品。但它本身是无形的,所以有不同于一般工程项目的管理。它必须通过人员组织管理、项目计划管理、配置管理等来保证软件按时高质量完成。

总之,为了解决软件危机,既要有技术措施(包括方法和工具),又要有必要的组织管理措施。软件工程正是从管理和技术两方面研究如何更好地开发和维护计算机软件的一门新兴学科。

软件工程是指导计算机软件开发和维护的一门学科。它采用工程的概念、原理、技术和方法,把经过时间考验而证明是正确的管理技术和当前能够得到的最好的技术方法结合起来,用于开发与维护软件。

1968 年在前联邦德国召开的国际会议上正式提出并使用了“软件工程”这个术语,运用工程学的基本原理和方法来组织和管理软件生产。后来还发展了与软件有关的心理