



高等学校电子信息类专业规划教材

软件工程与软件文档写作

文 斌 刘长青 田 原 编 著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



北京交通大学出版社

<http://press.bjtu.edu.cn>



21 世纪高等学校电子信息类专业规划教材

软件工程与软件文档写作

文 斌 刘长青 田 原 编 著

清华大学出版社

北京交通大学出版社

· 北京 ·

内 容 简 介

本书全面系统地讲述了现代软件工程的基本概念、原理及应用,重点是面向对象的软件开发方法、工具和过程,并针对软件生命周期的主要阶段,结合具体案例和相关标准,讲述软件主要文档的写作规范和技巧,全面提高软件技术人员的软件开发和软件文档编写能力,是指导计算机软件开发、维护与文档写作的工程类教材。

本书具有丰富的开发案例和软件文档模板,理论与实践相结合,每章后面备有形式多样的练习题和思考题。内容循序渐进、深入浅出,力求通俗易懂、突出重点、侧重应用。

本书不仅可作为高等院校高年级本科生或研究生和培训机构软件工程与软件文档写作的教材,也可作为广大自学软件工程及软件文档写作的读者的自学指导书和计算机软件技术人员的参考书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

软件工程与软件文档写作/文斌,刘长青,田原编著. —北京:清华大学出版社;北京交通大学出版社,2005.7

(21世纪高等学校电子信息类专业规划教材)

ISBN 7-81082-570-4

I. 软… II. ①文… ②刘… ③田… III. ①软件工程-高等学校-教材
②软件工程-应用文-写作-高等学校-教材 IV. ①TP311.5 ②H152.3

中国版本图书馆CIP数据核字(2005)第076636号

责任编辑:杨 玮 特邀编辑:孙江宏

出版者:清华大学出版社 邮编:100084 电话:010-62776969

北京交通大学出版社 邮编:100044 电话:010-51686414

印刷者:北京瑞达方舟印务有限公司

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:22.5 字数:543千字

版 次:2005年8月第1版 2005年8月第1次印刷

书 号:ISBN 7-81082-570-4/TP·212

印 数:1~4 000册 定价:29.00元

本书如有质量问题,请向北京交通大学出版社质监组反映。对您的意见和批评,我们表示欢迎和感谢。
投诉电话:010-51686043, 51686008; 传真:010-62225406; E-mail:press@center.bjtu.edu.cn。

前 言

信息社会在 21 世纪将以更快的速度、更高的要求 and 更新的技术飞速发展，其原因在于因特网和 IT 技术在全球的广泛应用和迅猛发展。以因特网作为平台，以计算机作为信息处理工具正成为时代发展的趋势。信息社会的发展使计算机软件的需求越来越旺盛，无论在数量上还是技术上都对软件开发提出了更高的要求，软件工程和软件产业面临着更大的挑战。

软件工程是指导软件生产、维护的一门工程学科，是计算机科学与技术研究的一个热点，自 20 世纪 60 年代末以来取得了长足的进步，并形成了较为完整的计算机软件开发方法，积累了一定的克服“软件危机”的经验。但是，与社会对计算机软件的要求相比，目前软件工程学科的发展还没有取得大的突破，软件的工业化开发程度还不高，软件工程学科要研究的课题还有很多，也正因为需求牵引的巨大能量，我们对软件工程学科的发展充满期待和信心，更有必要在软件从业人员和高等学校中大力普及软件工程知识。

软件文档是指在软件生命周期不同阶段中的某种数据媒体和其中所记录的数据。在软件工程中，软件文档常常用来表示对活动、需求、过程或结果进行描述、定义、规定、报告或认证的任何书面或图示的信息。软件文档是软件产品的一部分，没有文档的软件就不成其为软件。软件文档的编制在软件开发工作中占有突出的地位和相当大的工作量。高质量、高效率地开发、分发、管理和维护文档对于转让、变更、修正、扩充和使用文档，对于充分发挥软件产品的效益有着重要的意义。

本书以软件工程学科基本理论为基础，以软件生命周期中不同阶段的软件文档产品的写作为主线，重点培养读者的软件工程化开发的意识和对于不同软件文档的写作能力，既遵循有关标准又有一定的灵活性，编排上具有一定的独特性。其特色如下：

一是软件相关专业的教学越来越注重实践能力的培养，本书采用案例加模板的方法，有针对性地提高读者的软件开发和文档编写能力，做到学以致用。

二是以往的软件工程类教材只注意讲述软件开发的三要素（方法、工具、过程），而对软件文档只是泛泛而谈，甚至一笔带过。我们认为软件是程序和文档的集合，不管是国内的软件开发还是针对国外的外包市场，软件文档和文档写作能力都是非常重要的。产业界和高等学校读者都亟需了解在现代软件工程思想下指导软件文档写作的相关理论和技术。

三是本书将结合当代软件工程技术的发展实际，将软件开发方法主要分为面向数据流和面向对象两大类，系统地讲述软件工程相关知识和主要阶段的软件文档写作规范及技巧，并选择典型的软件工程开发案例及文档实例进行剖析，使读者能够在牢固掌握理论知识的同时，熟练运用所学技能和标准完成软件开发组织中文档写作的任务。特别是面向对象部分，使用 UML + Rose 进行开发方法讲述和文档写作能力培养，符合当前软件工程的发展方向。

四是对软件项目管理、软件质量与可靠性、CMM、软件复用等内容在本书中不予讲述，因为这些内容都有专门的课程来介绍。这样可以节约篇幅。

本书如用于教学，课程的考核方式建议以参加实际软件开发项目，编写主要软件文档

为主。

全书共分为 12 章。

第 1 章，软件工程与软件文档。主要介绍软件工程和软件文档的基本概念，包括软件、软件工程、软件文档在软件开发中的地位等等。

第 2 章，软件生命周期、过程模型和 CASE。这一章介绍软件开发过程的基本概念，包括软件生命周期、过程模型、CASE 工具等等；同时讲述了可行性研究报告的主要内容及写作要求。

第 3 章，软件文档写作。主要讲述软件文档的定义、软件文档的编写步骤以及如何写好计算机软件文档。

第 4 章，需求分析基础。介绍软件需求分析的基础知识，包括软件需求分析的目标和过程、需求获取技术、需求建模技术、需求分析人员及需求规格说明书等等。

第 5 章，面向数据流的分析。主要讲述面向数据流的需求分析方法，包括结构化分析方法、面向数据流的软件需求过程和面向数据流的软件需求分析文档。

第 6 章，面向对象的需求分析方法。这一章学习面向对象的需求分析方法，主要包括面向对象的基本概念、面向对象需求分析的过程、面向对象分析 OOA、面向对象需求分析文档。

第 7 章，统一建模语言 UML。全面介绍了统一建模语言 UML，主要包括软件建模的意义、UML 简介、支持 UML 建模的 CASE 工具 Rose、UML 用例分析、使用 UML 的软件过程、利用 UML 视图的需求说明书等。

第 8 章，面向数据流的软件设计。主要介绍软件设计的基本知识、面向数据流的软件设计和面向数据流的设计文档。

第 9 章，面向对象的软件设计。主要讨论面向对象设计的主要阶段、Coad 与 Yourdon 的 OOD、设计模式以及基于 UML 的面向对象的设计文档。

第 10 章，程序编码。主要讨论与程序编码有关的内容，重点介绍对编码质量有重要影响的编码风格和语言选择。

第 11 章，软件测试与测试文档。主要介绍软件测试的目标与原则、测试过程、测试基本技术、测试自动化以及软件测试文档写作。

第 12 章，软件维护。主要讲述软件维护的概念、软件维护的流程、软件再工程以及软件维护阶段文档。

本书第 3~7 章及第 11 章由文斌编写，第 1、2 章由刘长青编写，第 10、12 章由田原编写，第 8、9 章由刘长青和文斌共同编写。全书由文斌统一修改定稿。

本书可作为高等学校高年级本科生或研究生教材，也可供从事计算机软件工作的人员或用户阅读。

在本书编写过程中，参考了大量相关的文献资料，同时也得到了各方面有关专家和教师的大力支持和帮助，笔者在此一并表示衷心感谢。由于时间仓促，水平有限，书中难免有不妥之处，敬请读者不吝赐教。

编 者

2005 年 7 月

目 录

第 1 章 软件工程与软件文档	(1)
1.1 软件	(1)
1.1.1 软件的概念	(1)
1.1.2 软件的特点	(2)
1.2 软件的分类和发展	(4)
1.2.1 软件的分类	(4)
1.2.2 软件的发展历史和软件危机	(5)
1.3 软件工程	(8)
1.3.1 软件工程构成三要素	(9)
1.3.2 软件工程目标	(10)
1.3.3 软件工程的原则	(11)
1.3.4 12 个在不同场合反复出现的概念	(12)
1.4 软件文档在软件开发中的地位和作用	(14)
1.4.1 文档的地位和作用	(14)
1.4.2 对文档的基本要求	(15)
1.4.3 软件文档的种类	(18)
习题 1	(20)
第 2 章 软件生命周期、过程模型和 CASE	(21)
2.1 软件生命周期	(21)
2.1.1 软件定义	(22)
2.1.2 软件开发	(24)
2.1.3 软件使用、维护和退役	(25)
2.2 软件开发模型	(25)
2.2.1 瀑布模型	(26)
2.2.2 快速原型模型	(27)
2.2.3 增量模型	(28)
2.2.4 螺旋模型	(29)
2.2.5 面向对象开发模型	(30)
2.2.6 其他模型	(31)
2.2.7 各种模型的比较	(32)
2.3 CASE 工具和环境	(33)
2.3.1 计算机辅助软件工程	(33)
2.3.2 CASE 工具	(34)

2.4	可行性分析阶段的系统流程图	(35)
2.4.1	系统流程图	(35)
2.4.2	使用系统流程图的例子	(36)
2.5	可行性研究报告和项目开发计划	(38)
2.5.1	可行性研究报告的编写目的	(38)
2.5.2	可行性研究报告的主要内容及写作要求	(39)
2.5.3	项目开发计划	(44)
	习题2	(45)
第3章	软件文档写作	(46)
3.1	软件文档	(46)
3.1.1	软件文档定义	(46)
3.1.2	软件文档的工作	(47)
3.1.3	对文档编制的质量要求	(48)
3.2	软件文档的编写步骤	(50)
3.2.1	准备工作	(50)
3.2.2	确定写作内容	(52)
3.2.3	编写定稿	(54)
3.2.4	更新完善	(55)
3.3	如何写好计算机软件文档	(56)
3.3.1	做好准备	(56)
3.3.2	讲究文风	(59)
3.3.3	注重表达	(60)
3.3.4	加强文档编写使用的组织管理	(61)
3.3.5	文档写作中值得注意的几个问题	(63)
	习题3	(66)
第4章	需求分析基础	(67)
4.1	软件需求分析的目标和过程	(67)
4.1.1	需求分析的目标	(67)
4.1.2	需求分析的过程	(68)
4.2	需求获取技术	(69)
4.2.1	需求获取技术	(69)
4.2.2	实例分析	(70)
4.3	需求分析和描述技术	(71)
4.3.1	需求建模	(71)
4.3.2	问题抽象、问题分解与多视点分析	(71)
4.3.3	用于支持需求分析的快速原型化方法	(72)
4.3.4	需求管理的内容	(75)
4.4	需求分析人员	(78)
4.4.1	需求分析人员面临的挑战	(78)

4.4.2	需求分析人员的原则和策略	(79)
4.4.3	需求分析人员应采用的技术	(79)
4.5	软件需求规格说明和需求评审	(80)
4.6	软件需求规格说明书	(81)
习题4		(83)
第5章	面向数据流的分析	(84)
5.1	结构化分析方法	(84)
5.1.1	数据建模	(85)
5.1.2	功能建模和数据流	(86)
5.1.3	行为建模	(91)
5.1.4	数据词典	(93)
5.1.5	画数据流图的注意事项	(94)
5.2	面向数据流的分析过程	(96)
5.2.1	创建实体关系图	(96)
5.2.2	创建数据流模型	(97)
5.2.3	创建行为模型	(100)
5.2.4	编写加工规格说明	(101)
5.3	面向数据流的软件需求分析文档	(101)
习题5		(106)
第6章	面向对象的需求分析方法	(108)
6.1	什么是面向对象	(108)
6.1.1	面向对象概念	(108)
6.1.2	对象	(109)
6.2	面向对象需求分析的过程	(115)
6.3	面向对象分析	(116)
6.3.1	OOA 基本认识和基本方法	(116)
6.3.2	Coad 与 Yourdon 的 OOA 方法	(119)
6.4	面向对象的需求分析文档	(125)
习题6		(128)
第7章	统一建模语言 UML	(129)
7.1	软件建模的意义	(129)
7.2	UML 简介	(130)
7.2.1	视图	(131)
7.2.2	图	(132)
7.2.3	模型元素	(134)
7.2.4	语义规则	(135)
7.2.5	UML 主要部分的逻辑关系	(135)
7.3	UML 的用例分析过程	(138)
7.4	类和对象建模	(147)

7.4.1	类图和对象图	(147)
7.4.2	确定类	(148)
7.4.3	UML 中类之间的关系	(150)
7.4.4	包	(156)
7.4.5	模板	(157)
7.4.6	一个类图的实例	(157)
7.5	动态建模	(158)
7.5.1	消息	(158)
7.5.2	状态图	(158)
7.5.3	时序图	(159)
7.5.4	协作图	(165)
7.5.5	活动图	(169)
7.6	物理体系结构建模	(171)
7.6.1	逻辑体系结构和物理体系结构	(173)
7.6.2	构件图	(174)
7.6.3	配置图	(176)
7.7	支持 UML 的 CASE 工具——ROSE	(178)
7.7.1	Rose 的功能与特点	(178)
7.7.2	Rose 主界面	(179)
7.7.3	Rose 的基本操作	(181)
7.7.4	使用 Rose 的若干体会	(182)
7.8	使用 UML 的过程	(183)
7.8.1	UML 过程的基础	(183)
7.8.2	面向对象方法的一般过程	(185)
7.8.3	Rational Objectory 过程	(187)
7.9	利用 UML 视图的需求说明书	(188)
习题 7	(195)
第 8 章	面向数据流的软件设计	(196)
8.1	软件设计基础	(196)
8.1.1	软件设计的原则	(198)
8.1.2	软件体系结构	(201)
8.1.3	有效的模块设计	(203)
8.2	面向数据流的设计	(207)
8.2.1	面向数据流的概要设计	(207)
8.2.2	面向数据流的详细设计	(214)
8.2.3	面向数据流的设计举例	(223)
8.3	设计规格说明与设计评审	(225)
8.4	概要设计说明书和详细设计说明书	(227)
8.4.1	概要设计说明书	(227)

8.4.2 详细设计说明书	(229)
习题 8	(231)
第 9 章 面向对象的软件设计	(232)
9.1 面向对象设计	(232)
9.1.1 高层设计	(232)
9.1.2 类设计的目标和方针	(234)
9.1.3 通过复用设计类	(235)
9.1.4 类设计方法	(237)
9.1.5 面向对象设计的应用状况	(238)
9.2 对象模型技术	(238)
9.2.1 对象模型	(239)
9.2.2 动态模型	(244)
9.2.3 功能模型	(247)
9.2.4 基于三个模型的分析过程	(248)
9.3 Coad 与 Yourdon 面向对象设计方法	(248)
9.3.1 问题论域部分的设计	(248)
9.3.2 用户界面部分的设计	(250)
9.3.3 任务管理部分的设计	(252)
9.3.4 数据管理部分的设计	(253)
9.4 设计模式	(255)
9.4.1 设计模式的研究状况	(255)
9.4.2 设计模式对软件开发的作用	(256)
9.4.3 设计模式的定义	(256)
9.4.4 面向对象的设计模式	(256)
9.4.5 设计模式的描述	(257)
9.5 框架	(258)
9.5.1 框架的定义	(258)
9.5.2 白盒框架	(258)
9.5.3 黑盒框架	(260)
9.5.4 框架与设计模式	(261)
9.6 基于 UML 的面向对象软件设计文档	(261)
习题 9	(270)
第 10 章 程序编码	(271)
10.1 编码概念	(271)
10.2 结构化程序设计	(272)
10.3 程序设计语言	(275)
10.3.1 程序设计语言分类	(275)
10.3.2 程序设计语言的基本成分	(276)
10.3.3 程序设计语言选择	(279)

10.4	编码风格	(281)
10.5	编码策略与编码 CASE 工具	(283)
10.5.1	编码策略	(283)
10.5.2	支持编码的 CASE 工具	(284)
习题 10	(285)
第 11 章	软件测试与测试文档	(288)
11.1	软件测试的概念和原则	(288)
11.1.1	软件的错误	(288)
11.1.2	软件测试的目标和原则	(289)
11.1.3	软件测试信息流	(291)
11.2	软件测试方法	(292)
11.2.1	软件测试方法概述	(292)
11.2.2	黑盒测试方法	(294)
11.2.3	白盒测试方法	(296)
11.2.4	程序的静态测试	(298)
11.3	软件测试过程管理	(301)
11.3.1	软件测试过程	(301)
11.3.2	单元测试	(302)
11.3.3	集成测试	(305)
11.3.4	确认测试	(307)
11.3.5	系统测试	(309)
11.3.6	软件调试	(310)
11.3.7	测试中的可靠性分析	(312)
11.4	软件测试自动化	(314)
11.4.1	测试自动化的意义	(314)
11.4.2	测试自动化的若干做法	(315)
11.4.2	支持测试自动化的 CASE 工具	(318)
11.5	软件测试文档	(319)
11.5.1	测试文档的类型和作用	(319)
11.5.2	测试计划的主要内容和写作要求	(320)
11.5.3	测试分析报告的主要内容和写作要求	(322)
习题 11	(323)
第 12 章	软件维护	(325)
12.1	软件维护的概念	(325)
12.1.1	软件维护的定义	(325)
12.1.2	软件维护的特点	(326)
12.1.3	软件的可维护性	(328)
12.1.4	提高可维护性的方法	(329)
12.2	软件维护的实施	(332)

12.2.1 软件维护流程	(332)
12.2.2 软件维护的副作用	(334)
12.3 软件再工程	(335)
12.3.1 逆向工程与软件再工程	(335)
12.3.2 基于 ROSE 的逆向工程	(339)
12.4 软件维护阶段文档	(341)
12.4.1 维护申请单	(342)
12.4.2 维护报告单	(342)
12.4.3 用户手册的主要内容及写作要求	(343)
习题 12	(345)
参考文献	(346)

第 1 章 软件工程与软件文档

随着计算机技术的飞速发展，计算机成为科学和技术各个领域、工业和社会各个部门不可缺少的重要部分。特别是在当今高度信息化的时代，计算机提供了全球信息网络的通讯途径，成为人们以各种形式获取信息的手段。

遗憾的是，计算机在使社会生产力得到迅速解放、社会高度自动化和信息化的同时，却没有使计算机本身的软件生产得到类似的巨大进步。软件开发面临着过分依赖人工、软件无法重用、开发大量重复和生产率低下等问题，特别是软件危机的出现，促使人们努力探索软件开发的新思想、新方法和新技术，软件工程学便应运而生。

软件文档记录软件开发的活动和阶段性成果，它具有永久性并能供人或机器阅读。它不仅用于专业人员和用户之间的通信和交流，而且还可以用于软件开发过程的管理和运行阶段的维护。

这一章介绍软件工程和软件文档的基本概念，包括软件、软件工程、软件文档在软件开发中的地位等等。

1.1 软 件

1.1.1 软件的概念

软件是计算机系统中与硬件相互依存的另一部分，它是包括程序、数据及其相关文档的完整集合。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操纵信息的数据结构；文档是与程序开发、维护和使用有关的图文材料（见图 1-1）。

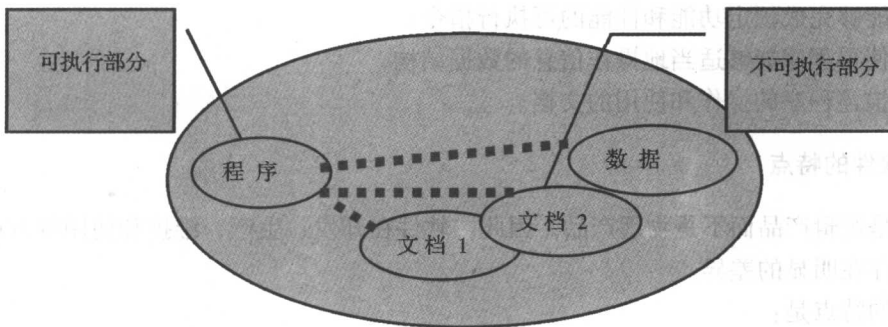


图 1-1 软件

程序是由程序设计语言所描述、能为计算机所识别、理解和处理的语句序列，程序设计语言具有良好、严格的语法和语义。

程序例子：

```

Main ( )
{ int i, j;           // 变量定义
  char Str \ [10 \];
  .....
  i = i + j;         // 语句说明
  .....;
}

```

目前程序设计语言主要有以下几种类型：

- 面向机器，如汇编语言、机器语言等；
- 面向过程，如 Fortran、Pascal、C 等等；
- 面向对象，如 Java 等等；
- 面向问题，如结构化查询语言 SQL 等等。

软件文档是记录软件开发活动和阶段性成果、理解软件所必需的阐述性资料。计算机软件不仅仅是程序，还应该有一整套文档资料。这些文档资料应该是在软件开发过程中产生的，而且应该是“最新式的”（即与程序代码完全一致的）。软件开发组织的管理人员可以利用这些文档资料作为“里程碑”，来管理和评价软件开发工程的进展状况；软件开发人员可以利用它们作为通信工具，在软件开发过程中准确地交流信息。对于软件维护人员而言，这些文档资料更是至关重要且必不可少的。缺乏必要的文档资料或者文档资料不合格，必然给软件开发和维护带来许多严重的困难和问题。软件文档依据软件生命周期分为许多种，例如：

- 需求分析文档；
- 软件设计文档等等。

编写软件文档的目的：

- 促进对软件的开发、管理和维护；
- 便于各种人员（用户和开发人员）的交流。

关于软件的概念，我们给出一个形式化的定义，即软件是：

- (1) 能够完成预定功能和性能的可执行指令；
- (2) 使得程序能够适当地操作信息的数据结构；
- (3) 描述程序的操作和使用的文档。

1.1.2 软件的特点

软件是逻辑产品而不是物理产品。因此，软件在开发、生产、维护和使用等方面与硬件相比均存在明显的差异。

软件的特点是：

- (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。
- (2) 软件的生产与硬件不同，它没有明显的制造过程。对软件的质量控制，必须着重在软件开发方面下功夫。
- (3) 在软件的运行和使用期间，没有硬件那样的机械磨损、老化等问题。

任何机械、电子设备在运行和使用中，其失效率大都遵循如图 1-2 (a) 所示的 U 形

曲线（即浴盆曲线）。即硬件在生命初期具有较高的故障率，这些故障主要是由于设计或制造的缺陷造成的。当这些缺陷修正后，故障率在一段时期内会降低到一个稳定的曲线上。随着时间的推移，硬件构件由于种种原因受到不同程度的损害，故障率又升高了。也就是说，硬件已经开始磨损了。

而软件的情况与此不同，因为它不存在磨损和老化问题。然而它存在退化问题，必须要多次修改（维护）软件。如图1-2（b）所示表明了软件的故障率曲线，在软件的生命初期隐藏的错误会使程序具有较高的故障率。在理想情况下，当这些错误改正后，曲线便趋于平稳，但实际情况是随着这些修改有可能引入新的错误，从而使故障率曲线呈现图中所示的锯齿状。于是，软件的退化由于修改而发生了。

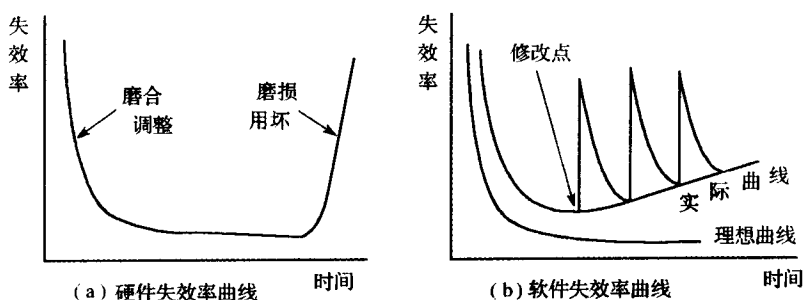


图 1-2 失效率曲线

(4) 软件的开发和运行常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。为了解除这种依赖性，在软件开发中提出了软件移植的问题。

(5) 软件的开发至今尚未完全摆脱手工艺的开发方式。

(6) 软件本身是复杂的。软件的复杂性可能来自它所反映的实际问题的复杂性，也可能来自程序逻辑结构的复杂性。

软件比任何其他人类制造的结构更复杂，甚至硬件的复杂性和软件相比也是微不足道的。软件本质上的复杂性是软件产品难以理解，影响软件过程的管理，并使维护过程十分复杂。

(7) 软件成本相当昂贵。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，成本是比较高。

IBM360 操作系统的研制人员最多时可达 1 000 多人，从 1963 年到 1966 年共花了四年时间才完成，总计耗费了 5 000 多人年，以后又进行不断的修改和补充。该系统的整个研制费用为 5 亿美元，其中近一半花在软件上。

(8) 相当多的软件工作涉及到社会因素。许多软件的开发和运行涉及机构、体制及管理方式等问题，甚至涉及到人的观念和人们的心理。这些都直接影响到项目的成败。

1.2 软件的分类和发展

1.2.1 软件的分类

20世纪40年代以来, 尽管人们开发了大量的软件, 积累了丰富的软件资源并使之广泛应用于科学研究、教育、工农业生产、事务处理、国防和家庭, 但在软件的品种、质量和价格方面仍然满足不了日益增长的需要。计算机软件产业是一个年轻、充满活力和飞速发展的产业。

下面介绍计算机软件的一些常用划分标准。

(1) 按软件的功能进行划分

- 系统软件: 能与计算机硬件紧密配合在一起, 使计算机系统各个部件、相关软件和数据协调、高效工作的软件。例如, 操作系统、数据库管理系统、设备驱动程序以及通信处理程序等。

- 支撑软件: 是协助用户开发软件的工具性软件, 其中包括帮助程序人员开发软件产品的工具, 也包括帮助管理人员控制开发进程的工具。

- 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。

(2) 按软件规模进行划分

按开发软件所需的人力、时间以及完成的源程序行数, 可确定六种不同规模的软件(见表1-1)。

表 1-1 软件规模的分类

类别	参加人员数	研制期限	产品规模 (源程序行数)
微型	1	1 ~ 4 周	500
小型	1	1 ~ 6 月	$1 \times 10^3 \sim 2 \times 10^3$
中型	2 ~ 5	1 ~ 2 年	$5 \times 10^3 \sim 50 \times 10^3$
大型	5 ~ 20	2 ~ 3 年	$5k \sim 50 \times 10^3 \sim 100 \times 10^3$
甚大型	100 ~ 1 000	4 ~ 5 年	1×10^6
极大型	2 000 ~ 5 000	5 ~ 10 年	$1 \times 10^6 \sim 10 \times 10^6$

规模大、时间长、参加人员多的软件项目, 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也得有软件工程概念, 遵循一定的开发规范。无论开发规模大小, 基本原则是一样的, 只是对软件工程技术依赖的程度不同而已。

(3) 按软件工作方式划分

- 实时处理软件: 指在事件或数据产生时立即予以处理, 并及时反馈信号, 控制需要监测和控制的过程的软件。主要包括数据采集、分析和输出三部分。

- 分时软件: 允许多个联机用户同时使用计算机。

- 交互式软件：能实现人机通信的软件。
- 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。

(4) 按软件服务对象的范围划分

• 项目软件：也称定制软件，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。例如军用防空指挥系统、卫星控制系统。

• 产品软件：是由软件开发机构开发出来直接提供给市场，或是为千百个用户服务的软件。例如，文字处理软件、文本处理软件、财务处理软件、人事管理软件等。

(5) 按使用的频度进行划分

有的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，如天气预报软件。

(6) 按软件失效的影响进行划分

有的软件在工作中出现了故障，造成软件失效，可能给整个系统带来的影响不大。有的软件一旦失效，可能酿成灾难性后果，例如财务金融、交通通信、航空航天等软件。我们称这类软件为关键软件或生命攸关软件。

1.2.2 软件的发展历史和软件危机

1. 软件的发展历史

随着计算机硬件性能的极大提高和计算机体系结构的不断变化，计算机软件系统更加成熟和更为复杂，从而促使计算机软件的角色发生了巨大的变化，其发展历史大致可以分为如图 1-3 所示的四个阶段。

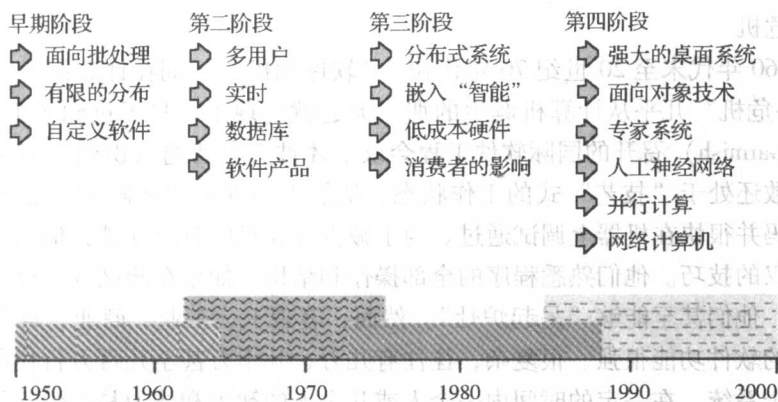


图 1-3 软件的发展阶段

(1) 早期阶段

在计算机发展的早期阶段，人们认为计算机的主要用途是快速计算，软件编程简单，不存在什么系统化的方法，开发没有任何管理，程序的质量完全依赖于程序员个人的技巧。