



软件工程

— 理论、方法及实践

刘忠宝 主 编



国防工业出版社

National Defense Industry Press

软 件 工 程

——理论、方法及实践

刘忠宝 主编

国防工业出版社

·北京·

内 容 简 介

本书根据软件技术的最新发展,结合目前软件教学的需要,全面系统地讲述了软件工程以及软件项目管理的概念、原理和方法,通过软件工程实践和软件项目管理实训,使读者能够理论联系实际,全面掌握软件开发所需的知识体系。

本书强调理论与实践相结合、技术与管理相结合,注重培养实际开发能力和文档的写作能力,具有很强的实用性和可操作性。全书内容条理清晰、语言流畅、通俗易懂。

本书可作为高等院校计算机专业和其他相关信息类专业本科高年级或研究生教材,也可作为从事软件开发、管理、维护 and 应用的工程技术和管理人员的参考书。

图书在版编目(CIP)数据

软件工程——理论、方法及实践/刘忠宝主编. —北京:
国防工业出版社, 2012. 1
ISBN 978-7-118-07783-4

I. ①软... II. ①刘... III. ①软件工程 IV. ①
TP311.5

中国版本图书馆 CIP 数据核字(2011)第 280348 号

※

国防工业出版社 出版发行
(北京市海淀区紫竹院南路 23 号 邮政编码 100048)
北京奥鑫印刷厂印刷
新华书店经售

*

开本 787 × 1092 1/16 印张 15½ 字数 347 千字
2012 年 1 月第 1 版第 1 次印刷 印数 1—3000 册 定价 30.00 元

(本书如有印装错误,我社负责调换)

国防书店: (010)88540777 发行邮购: (010)88540776
发行传真: (010)88540755 发行业务: (010)88540717

前 言

计算机软件在当今信息社会起着越来越重要的作用,软件技术和方法已经成为信息社会高技术竞争的关键领域之一。从最早的科学计算、文字处理、数据库管理、银行业务处理到工业自动控制和生产、新闻媒体、娱乐等,软件的应用已经覆盖了各行各业。随着人们对软件的重视,软件的设计与开发成为人们生活的组成部分。软件技术发展速度快、成果多,特别在开发工具、支撑环境、分析设计方法、软件工程理论与软件管理方法等方面取得了巨大的进步。但在实践中,虽然软件开发的效率与质量大大提高,而要使其实现如传统工业那样的标准化与工业化,仍然感到困难。很多组织虽然不断引进新的技术与方法,但在软件生产中的许多传统问题依然存在并且没有得到质的改善;在软件生产管理中,开发者与管理者的不协调比任何其他行业都明显;在软件生产理论研究中,研究者与实践者的脱节也比任何其他领域都严重。为解决上述问题特编写本书。本书从实际应用出发,全面总结软件技术与管理的发展现状,指出软件设计与开发过程中存在问题的根源,系统介绍软件开发和管理的基本概念、原理和方法,使读者能从技术与管理两个层面领会软件项目设计与开发思想和方法。

本书作者从事计算机相关工作 10 余年,具有扎实的专业基础和良好的科研能力。先后多次主持、参与国家级和省部级项目,并在国内外核心期刊上发表多篇科研论文,得到业界专家的认可。此外,本书作者积极参与各级各类教学研究,多年讲授“软件工程”、“软件项目管理”、“Java 程序设计”、“软件测试技术”等软件工程专业核心课程,积累了较为丰富的教学经验。

本书由刘忠宝、赵怡、樊东燕、闫俊伢等编写。刘忠宝全面负责统稿工作,闫俊伢、郭慧、许敏参与统稿。全书共 4 篇 14 章,其中第 5 章、第 12 章、第 14 章由刘忠宝编写,第 3 章、第 13 章由赵怡编写,第 2 章、第 6 章由樊东燕编写,第 8 章、第 10 章由闫俊伢编写,第 7 章、第 9 章由郭慧编写,第 1 章、第 4 章由许敏编写,第 11 章由邓赵红编写。张永奎、王建珍、李月娥等教授对本教材提出了许多宝贵意见,苏彩、张书浩、宋波等同学完成部分图表绘制工作。本书还得到作者所在学校各方面的大力支持,强大的合作团队、优良的专业素质、严谨的工作作风使本书得以顺利完成。在本书的编写过程中,我们参阅了大量的资料,在此对所有编著者以及对本书的编写和出版做过工作的老师和同学一并表示衷心的感谢。

本书可作为高等院校计算机专业和其他相关信息类专业本科高年级或研究生教材,也可作为从事软件开发、管理、维护和应用的工程技术和管理人员的参考书。

由于时间仓促、水平有限,本书难免存在不妥之处,恳请读者见谅并能及时提出宝贵意见。

编者
2011年12月

目 录

第一篇 软件工程方法学

第1章 软件工程概述	2
1.1 软件	2
1.1.1 软件的定义	2
1.1.2 软件的特点	2
1.1.3 软件的分类	3
1.1.4 软件的发展及软件危机	3
1.2 软件工程概述	5
1.2.1 软件工程的定义	5
1.2.2 软件工程的内容	6
1.2.3 软件工程的基本原理	7
1.3 软件生命周期及软件过程模型	8
1.3.1 软件生命周期	8
1.3.2 软件过程模型	10
小结	16
习题	17
第2章 传统方法学	18
2.1 结构化分析	18
2.1.1 可行性分析	19
2.1.2 需求分析	32
2.2 结构化设计	39
2.2.1 结构化设计概述	39
2.2.2 概要设计	43
2.2.3 详细设计	48
2.3 结构化实现	50
2.3.1 编码	50
2.3.2 测试	51
2.3.3 维护	57
小结	62

习题	62
第3章 面向对象方法学	63
3.1 面向对象方法概述	63
3.1.1 面向对象的定义	63
3.1.2 面向对象的基本概念	64
3.1.3 面向对象的软件工程方法	67
3.2 面向对象建模	68
3.2.1 面向对象建模概述	68
3.2.2 对象模型	69
3.2.3 动态模型	72
3.2.4 功能模型	74
3.3 面向对象分析	76
3.3.1 面向对象分析概述	76
3.3.2 建立功能模型	77
3.3.3 建立对象模型	78
3.3.4 建立动态模型	80
3.4 面向对象设计	81
3.4.1 子系统的分解	82
3.4.2 问题域部分的设计	83
3.4.3 人机交互部分的设计	84
3.4.4 任务管理部分的设计	84
3.4.5 数据管理部分的设计	84
3.5 面向对象实现	85
3.5.1 程序设计语言及风格	85
3.5.2 面向对象测试策略	86
小结	88
习题	88

第二篇 软件项目管理

第4章 软件项目管理概述	90
4.1 项目的概念	90
4.1.1 项目	90
4.1.2 项目管理	91
4.2 项目管理的相关理论体系	93
4.2.1 项目管理知识体系	94
4.2.2 受控环境中的项目	96

4.2.3	WWPMM	97
4.3	我国软件项目管理发展历程	98
4.3.1	我国项目管理的产生及发展历程	98
4.3.2	我国软件项目管理的现状	99
4.3.3	我国软件项目管理发展展望	100
小结		101
习题		101
第5章	软件项目需求管理	102
5.1	软件需求概述	102
5.1.1	软件需求的层次划分	102
5.1.2	用户需求与特点分析	103
5.1.3	系统需求与类型划分	103
5.1.4	软件需求规格说明书	104
5.2	需求管理方法与内容	105
5.2.1	需求管理的含义	105
5.2.2	需求管理的复杂性	106
5.2.3	需求管理的方法	106
5.2.4	需求管理的过程	107
5.3	软件项目的任务分解	108
5.3.1	工作分解结构	108
5.3.2	工作分解的操作步骤	109
5.3.3	工作分解结构的表示形式	110
5.3.4	任务分解的注意事项	110
5.4	软件需求的变更控制	110
5.4.1	不可避免的需求变更	111
5.4.2	需求变更的原因分析	111
5.4.3	管理需求变更的请求	111
5.4.4	需求变更的控制流程	111
小结		113
习题		113
第6章	软件项目进度管理	115
6.1	软件项目进度管理概述	115
6.1.1	项目进度管理的重要性	115
6.1.2	项目进度管理的内容及原理	116
6.1.3	项目进度管理的阶段划分	117
6.2	项目的进度计划	120
6.2.1	项目的进度计划概述	120

6.2.2	项目进度计划方法	121
6.2.3	项目进度计划的工具	124
6.3	项目的进度控制	129
6.3.1	项目进度控制的概念	129
6.3.2	项目进度控制的主要影响因素	130
6.3.3	项目进度控制的常用方法	131
6.3.4	项目进度计划的调整	132
小结		133
习题		133
第7章	软件项目成本管理	134
7.1	概述	135
7.1.1	软件项目成本的分类	135
7.1.2	软件项目成本的影响因素	136
7.2	成本估算	136
7.2.1	软件开发项目成本估算过程	137
7.2.2	软件项目成本估算方法	137
7.2.3	经验成本估算模型	139
7.3	成本预算	140
7.3.1	软件成本预算的特性和原则	140
7.3.2	软件项目成本预算方法	141
7.3.3	成本预算的结果	142
7.4	成本控制	142
7.4.1	软件项目成本控制内容	142
7.4.2	软件项目成本控制方法	143
小结		145
习题		145
第8章	软件项目质量管理	146
8.1	软件质量的基本概念	146
8.1.1	质量的含义与属性	146
8.1.2	软件质量的含义和特性	147
8.1.3	影响软件质量的因素	149
8.2	软件质量管理的发展过程及其实施	150
8.2.1	软件质量管理的发展过程	150
8.2.2	软件质量管理的实施	150
8.2.3	软件质量管理的原则	152
8.3	软件质量管理的内容	153
8.3.1	软件项目的质量计划	153

8.3.2	软件项目的质量保证	154
8.3.3	软件项目的质量控制	155
小结		156
习题		156
第9章	软件项目风险管理	157
9.1	概述	157
9.1.1	风险与风险管理	157
9.1.2	软件项目风险	159
9.1.3	软件项目风险管理	159
9.2	风险管理计划编制	161
9.2.1	风险管理计划编制的依据	162
9.2.2	风险管理计划编制的方法	162
9.2.3	风险管理计划内容	162
9.3	风险识别	163
9.3.1	风险识别的依据	163
9.3.2	常见的软件风险	164
9.3.3	风险识别的过程	165
9.3.4	风险识别的方法	166
9.4	风险分析	168
9.4.1	风险分析过程	168
9.4.2	风险分析技巧	171
9.4.3	风险分析的结果	172
9.5	风险应对	173
9.5.1	风险应对策略	173
9.5.2	风险应对过程	174
9.6	风险监控	174
9.7	风险管理验证	175
小结		175
习题		176
第10章	软件项目配置管理	177
10.1	配置管理规划	177
10.1.1	软件配置管理的概念	177
10.1.2	配置管理计划	178
10.2	配置管理过程	179
10.2.1	配置项标识	179
10.2.2	版本编号	180
10.2.3	变更控制	181

10.2.4 配置状态报告	182
小结	183
习题	183

第三篇 软件工程实践

第 11 章 软件工程工具	185
11.1 统一建模语言	185
11.1.1 统一建模语言的由来与发展	185
11.1.2 标准建模语言的主要特点	185
11.2 软件需求分析、设计和构建工具	186
11.2.1 IBM Rational RequisitePro	186
11.2.2 IBM Rational Software Modeler	186
11.2.3 Microsoft Visio	187
11.2.4 IBM Rational Rose	187
11.2.5 IBM Rational Software Architect	188
11.2.6 Microsoft Visual Studio 2005 Team Edition for Software Architects	188
11.2.7 其他工具及发展趋势	189
11.3 软件测试和质量保证工具	190
11.3.1 IBM Rational PurifyPlus	190
11.3.2 WinRunner	191
11.3.3 LoadRunner	191
11.3.4 CODETEST	191
11.3.5 Visual Studio 2005 Team Test Edition	192
11.4 软件配置管理工具	193
11.4.1 IBM Rational ClearCase	193
11.4.2 IBM Rational ClearQuest	193
11.5 过程和项目管理工具	194
11.5.1 RUP 统一软件开发过程	194
11.5.2 IBM Rational Method Composer	194
11.5.3 IBM Rational Team Unifying Platform	195
11.6 数据库建模工具	195
11.6.1 PowerDesigner	195
11.6.2 IBM Rational Rose Data Modeler	196
小结	196
习题	196
第 12 章 结构化方法实验	197
12.1 可行性研究实验	197

12.2	需求分析实验	200
12.3	软件设计实验	203
12.4	软件测试实验	208
第 13 章	面向对象方法实验	213
13.1	面向对象系统分析	213
13.1.1	实验目的	213
13.1.2	实验环境	213
13.1.3	实验内容	213
13.1.4	实验成果	218
13.1.5	实验思考题	218
13.2	面向对象系统设计	218
13.2.1	实验目的	218
13.2.2	实验环境	218
13.2.3	实验内容	218
13.2.4	实验成果	220
13.2.5	实验思考题	220

第四篇 软件项目管理实训

第 14 章	软件项目管理实训	222
14.1	项目启动阶段	223
14.2	项目计划阶段	224
14.2.1	软件项目计划阶段任务	225
14.2.2	软件项目计划阶段典型文档	226
14.3	项目执行控制阶段	228
14.3.1	软件项目执行控制阶段的任务	228
14.3.2	软件项目执行阶段的典型文档	229
14.4	项目结束阶段	232
	小结	233
	习题	233
	参考文献	234

第一篇

软件工程方法学

软件工程学是一门软件开发的工程方法学,它在软件开发中的指导意义和基础地位已经越来越得到 IT 界的重视。软件工程学包括支持软件开发和维护的理论、技术和方法。这些内容对于软件研发人员和软件项目管理人员都是必需的。本章全面系统地介绍了软件工程理论、原理和方法,同时兼顾了传统软件开发方法和时下较为新颖的技术和方法。本篇突出实用性,通过具体案例描述软件开发过程,帮助读者理解和消化所学内容,迅速提高实践能力。

第 1 章 软件工程概述

自 20 世纪 40 年代出现了世界上第一台计算机以来,计算机软件经历了三个发展阶段。在发展过程中,出现了“软件危机”的困扰,并发现人们开发软件产品的能力大大落后于计算机硬件的发展和社会对计算机软件不断增长的需求。

为了消除“软件危机”,更有效地开发与维护软件,在 1968 年北大西洋公约组织(NATO)的一次会议上正式提出了“软件工程”的概念,并在之后的 40 多年时间里得到了迅速的发展,在指导人们科学地开发软件、制作软件产品、集成计算机系统、保证软件产品的质量,按期并以合理的成本完成软件产品的生产等方面起到了巨大的作用。本章将主要介绍软件和软件工程的相关概念,并简要介绍几种软件过程模型。

1.1 软 件

1.1.1 软件的定义

“软件”一词是在 20 世纪 60 年代出现的,其含义是为了特定目的而开发的程序、数据和文档的集合。其中:

(1) 程序,即为了完成特定功能而编制的一组指令集。

(2) 数据,即执行程序所必需的数据和数据结构。

(3) 文档,即与程序开发、维护和使用有关的图文资料,如软件开发计划书、需求规格说明书、设计说明书、测试分析报告和用户手册等。

1.1.2 软件的特点

要理解软件的含义,首先要了解软件的特点。虽然,软件与硬件一样也是产品,但两者之间是有差别的,了解这种差别对理解软件工程是非常重要的。

1. 软件具有抽象性

软件是开发而成的,是一种逻辑实体,不是制造产生的,即不是具体的物理实体,因而它具有抽象性。

2. 软件“开发”有别于硬件“制造”

软件没有明显的制造过程,因而软件的质量主要取决于软件的“开发”。在开发过程中,通过人们的智力活动和有效的管理,把知识和技术转化为信息产品。一旦某一软件产品开发成功,以后就可以大量地复制同一内容的副本。

3. 软件不会“磨损”,但会“退化”

在软件的运行和使用期间,没有硬件那样的机械磨损和老化问题。如图 1.1 所示,图(a)显示的是硬件的失效率曲线,它存在老化与磨损的特点;图(b)显示的是软件的失效

率曲线,它存在退化问题,必须要多次修改(维护)软件。图(b)中凸出的部分是由于修改而产生的副作用造成故障率的提高。

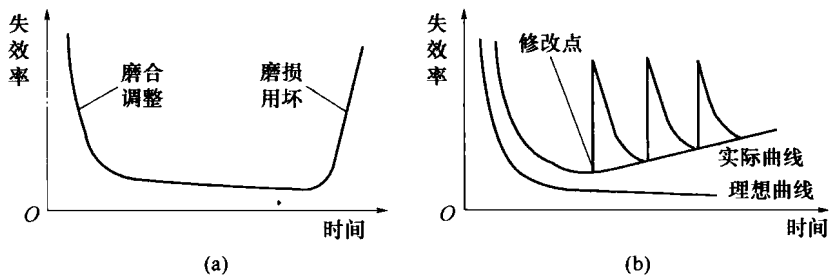


图 1.1 失效率曲线

(a) 硬件失效率曲线; (b) 软件失效率曲线。

4. 软件是定制的且为手工完成

在硬件制造业,构件的复用是非常普遍的。但由于软件本身的特殊性,构件复用才刚刚起步。理想情况下,软件构件应该被设计成能够被复用于不同的程序,尽管今天的面向对象技术、构件技术已经使软件的复用逐渐流行,但这种复用还不能做到像硬件产品那样拿来即用,还需要进行必要的定制(构件之间的组合,接口的设计,功能的修改与扩充等),而且软件开发中构件的使用比例也是有限的。整个软件产品的设计基本上还是依赖于人们的智力与手工劳动。

5. 开发过程复杂且费用昂贵

现代软件的体系结构越来越复杂,规模越来越庞大。所涉及的学科也越来越多,导致了软件的开发过程也异常复杂。靠一个人单枪匹马开发一套软件的时代已经一去不复返了,软件的开发需要一个分工明确、层次合理、组织严密的团队才能完成,显然软件的开发成本也会越来越昂贵。

1.1.3 软件分类

软件的应用非常广泛,几乎渗透到各行各业,因此要给出关于计算机软件一个科学、统一、严格的分类标准是不现实,也是不可能的,但可以从不同角度对软件进行适当的分类。常用的分类方法:

- (1) 按软件功能分:系统软件、支撑软件、应用软件。
- (2) 按软件工作方式分:实时软件、分时软件、交互式软件、批处理软件。
- (3) 按软件规模分:微型软件、小型软件、中型软件、大型软件、超大型软件、极大型软件。
- (4) 按软件服务对象范围分:定制软件、产品软件。
- (5) 按应用领域分:操作系统、数据库管理系统、软件开发系统、办公软件(包括字处理软件、电子表格软件等)、财务软件、网络工具软件、图形图像处理软件、多媒体软件、游戏软件、家庭教育软件等。

1.1.4 软件的发展及软件危机

自 20 世纪 40 年代出现了世界上第一台计算机以后,就有了程序的概念。其后经历

了几十年的发展,计算机软件经历了三个发展阶段。

1. 程序设计阶段(1946 年—1956 年)

这一阶段从第一台计算机上的第一个程序出现持续到高级程序设计语言出现以前。这一阶段所采用的程序设计语言是汇编及机器语言,主要应用于科学计算。其主要特点:

(1) 程序设计是一种由人发挥创造才能的技术领域。编写出的程序只要能在计算机上运行速度快、占用内存少,并能得出正确的结果,程序的写法就可以不受任何约束,很少考虑到结构清晰、可读性和可维护性。程序开发人员把自己编写的程序看做按个人意图创造的“艺术品”,过于强调编程技巧。

(2) 程序开发者只是为了满足自己的需要。这种自给自足的个体生产方式效率低下,程序的设计和编制工作复杂、集琐、费时和易出差错。研究范围局限于科学计算程序、服务性程序和程序库,研究对象是顺序程序。对和程序有关的文档的重要性认识不足。

2. 程序系统阶段(1956 年—1968 年)

这一阶段从实用的高级程序设计语言出现持续到软件工程出现。该阶段所采用的程序设计语言是高级程序设计语言;其应用领域也逐步扩大,除了科学计算外,还增加了性质和科学计算有明显区别的大数据集处理问题。其主要特点:

(1) 就一项计算任务而言计算量不大,但输入、输出量较大。这时,机器结构转向以存储控制为中心,出现了大容量的存储器,外围设备也随之迅速发展。为了适应大量数据处理问题的需要,开始使用数据库及其管理系统。

(2) 人们逐渐认识到和程序有关的文档的重要性,开始将程序及其有关的文档融为一体。

(3) 随着计算机应用的日益普及,软件数量急剧膨胀,出现了“软件作坊”,但其基本上仍沿用早期形成的个体化软件开发方法。

在软件发展的这个阶段,软件的维护工作,如在程序运行时发现的错误必须设法改正;用户有了新的需求必须修改程序;硬件或操作系统更新时,也需要修改程序以适应新的环境等,以令人吃惊的比例耗费资源。更严重的是许多程序的个体化特性使它们最终成为不可维护的。“软件危机”就这样开始出现了。

软件危机是指在计算机软件的开发和维护过程中所遇到的一系列严重问题。包含以下两方面:

(1) 如何开发软件,以满足对软件日益增长的需求;

(2) 如何维护数量不断膨胀的已有软件。

产生软件危机的原因:

(1) 软件开发是一项复杂的工程,需要用科学的工程化的思想来组织和指导软件开发的各个阶段,但很多软件开发人员简单地认为软件开发就是程序设计。

(2) 没有完善的质量保证体系。软件的质量得不到保证,开发出来的软件产品往往不能满足需求,同时还可能需要花费大量的时间、资金和精力去修复软件的缺陷,从而导致软件质量下降和开发预算超支等后果。

(3) 软件文档的重要性没有得到软件开发人员和用户的足够重视。软件文档是软件开发团队成员之间交流和沟通的重要平台,也是软件开发项目管理的重要工具。如果不能充分重视软件文档的价值,势必会给软件开发带来很多不便。

(4) 从事软件开发的专业人员对这个产业认识不充分,缺乏经验。

(5) 软件独有的特点也给软件的开发和维护带来困难。软件的抽象性和复杂性使得软件在开发之前,很难对开发过程的进展进行估计。再加上软件错误的隐蔽性和改正错误的复杂性,都使得软件开发和维护在客观上比较困难。

其典型表现:

(1) 软件的成本和开发进度不能预先估计;

(2) “已完成的”软件系统不能满足客户的需求;

(3) 软件产品质量差,可靠性不能保证;

(4) 软件产品的可维护性差;

(5) 软件没有合适的文档资料;

(6) 软件的发展速度跟不上硬件的发展和用户的需求,软件成本高等。

如果软件危机不消除,软件的发展是没有出路的,要解决软件危机问题,需要采取以下措施:

(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) 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。

由此,软件的发展进入软件工程阶段,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。

3. 软件工程阶段(1968 年至今)

软件工程出现以后迄今为软件工程阶段。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。

1.2 软件工程概述

1.2.1 软件工程的观念

1968 年,在北大西洋公约组织举行的一次学术会议上,首次提出了“软件工程”的概念,并将其定义为“为了经济地获得可靠的和能在实际机器上高效运行的软件,而建立和使用的健全的工程规则”。这个定义肯定了工程化的思想在软件工程中的重要性,但是并没有提到软件产品的特殊性。

经过 40 多年的发展,软件工程已经成为一门独立的学科,人们对软件工程也逐渐有了更全面、更科学的认识。我国 2006 年的国家标准 GB/T 11457—2006《软件工程术语》中对软件工程定义为“应用计算机科学理论和技术以及工程管理原则和方法,按预算和进度,实现满足用户要求的软件产品的定义、开发、发布和维护的工程或进行研究的学科”。

软件工程的提出是为了解决软件危机所带来的各种弊端。具体地讲,软件工程的目标主要包括以下几点: