

21 世纪高等院校规划教材 · 计算机类



软件工程

主编 王忠群

RUANJIAN
GONGCHENG

中国科学技术大学出版社

000766767



21 世纪高等院校规划教材·计算机类

软 件 工 程

王忠群 主编

中国科学技术大学出版社

2009·合肥

内 容 简 介

软件工程是研究软件开发、维护和管理的一门工程科学,是计算机科学技术及相关专业的主干课程。

全书共 12 章,主要包括:软件工程概述;软件需求基础;软件设计基础;结构化分析与设计;面向对象的分析与设计;人机界面设计;编码;软件测试;软件维护;软件项目管理;软件工程标准化与软件文档;Web 工程等。

图书在版编目(CIP)数据

软件工程/王忠群主编. —合肥:中国科学技术大学出版社,2009. 11
ISBN 978-7-312-02572-3

I. 软… II. 王… 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 133979 号

出版发行 中国科学技术大学出版社
地址:安徽省合肥市金寨路 96 号,邮政编码:230026
网址:<http://press.ustc.edu.cn>
电话:发行部 0551-3602905 邮购部 0551-3602906

印	刷	合肥华星印务有限责任公司
经	销	全国新华书店
开	本	787 mm×1092 mm 1/16
印	张	21.5
字	数	520 千
版	次	2009 年 11 月第 1 版
印	次	2009 年 11 月第 1 次印刷
印	数	1—4000 册
定	价	32.00 元

前 言

软件工程是研究软件开发、维护和软件管理的一门工程科学,是计算机科学技术及相关专业的主干课。自 1968 年在著名的 NATO 会议上为解决“软件危机”而提出“软件工程”的概念以来,在不到半个世纪的时间里,软件工程在理论和实践两个方面都取得了长足的进步,取得了大量研究成果,软件工程的应用水平已成为促进软件产业健康发展的关键。

随着计算机的日益普及和广泛应用,尤其是 Internet 的出现,软件系统的规模和复杂度与日俱增,软件技术面临着许多新的挑战。大型复杂软件的开发是一项特殊的工程,不仅与传统工程一样,需要按照工程化的方法去组织管理软件的开发,而且软件开发更具特殊性、复杂性。因此软件工程已经成为计算机科学与技术学科的重要学科方向。

针对绝大多数本科院校以培养应用型人才为目标,着眼于学生综合素质和应用能力、创新意识的培养,贯彻以必需、够用、实用和会用为度的原则,优化课程内容体系,适当体现新技术、新方法,突出软件工程课程知识的连贯性、逻辑性,试图通过实例来化解软件工程的较强抽象性,为读者提供一本既保持知识的系统性,适当反映软件工程最新发展成果,又易于学习和易于应用于实践的教科书。

全书共分 12 章。第 1 章概括介绍了软件的基本概念、软件危机产生原因和解决途径以及软件工程学产生的背景、基本原理、概念和方法等。第 2 章概述了软件需求分析、介绍可行性研究、初步需求获取方法、需求分析的过程和需求规格说明和需求评审。第 3 章概述了软件设计的任务、目标和过程,详细阐述软件设计的基本概念,介绍了软件体系结构设计、数据结构设计、过程设计以及软件设计规格说明和评审。第 4 章详细介绍了结构化分析、设计方法以及设计启发式设计策略。第 5 章在对面向对象的概念和 UML 介绍的基础上,详细叙述了面向对象分析、设计方法和 Rational Rose 建模工具。第 6 章介绍了软件界面设计。第 7 章到第 9 章分别介绍了软件的编码、软件测试和软件维护。第 10 章介绍了软件项目管理和软件过程成熟度模型(CMM)。第

11 章介绍软件工程标准化与软件文档。第 12 章是软件工程发展的部分新成果。针对运行于 Internet 环境下软件的开发,第 12 章较详细介绍了 Web 工程,包括 Web 系统的特点、Web 工程概念、Web 项目策划、Web 应用的分析、设计和测试等。

本书由王忠群主编。第 1 章由修宇编写,第 2 章由郁书好编写,第 3、11 章由郦丹芸编写,第 4 章由秦学勇编写,第 5 章由张佩云编写,第 6、7 章由刘涛编写,第 8 章由姚保峰编写,第 9 章由王池社编写,第 10 章由王勇编写,第 12 章由王忠群编写。全书由王忠群负责统稿和定稿工作,修宇也参与部分统稿工作。

在本书的编写过程中得到了有关专家的热心指导和无私帮助,同时中国科学技术大学出版社也为本书的尽快出版做了大量工作,编者在此一并表示衷心的感谢。此外,本书编写时还参考了大量文献资料,在此向这些有关文献资料的作者深表谢意。

由于作者学术水平有限,书中疏漏、欠妥、谬误之处在所难免,恳请各位专家、读者批评指正。

编 者

2009 年 7 月

目 录

前言	(I)
第1章 软件工程概述	(1)
1.1 软件	(1)
1.1.1 软件的概念与特点	(1)
1.1.2 软件的分类	(4)
1.2 软件危机	(4)
1.3 软件工程	(6)
1.3.1 软件工程的概念	(6)
1.3.2 软件工程的基本目标	(7)
1.3.3 软件工程的基本原理	(7)
1.3.4 软件工程的原则	(8)
1.4 软件生存周期	(9)
1.4.1 软件的定义阶段	(10)
1.4.2 软件的开发阶段	(10)
1.4.3 软件的使用、维护和退役阶段	(11)
1.5 软件的开发模型	(12)
1.5.1 瀑布模型	(12)
1.5.2 原型模型	(13)
1.5.3 螺旋模型	(14)
1.5.4 喷泉模型	(16)
1.5.5 基于第四代技术的模型	(16)
1.5.6 增量模型	(17)
1.5.7 迭代模型	(18)
1.6 软件开发方法和工具	(22)
1.6.1 结构化方法	(22)
1.6.2 面向对象方法	(23)
1.6.3 软件工具与开发环境	(23)
小结	(24)
习题	(25)
第2章 软件需求基础	(26)
2.1 需求分析概述	(26)
2.1.1 需求分析的难点	(26)
2.1.2 需求分析的基本原则	(26)

2.1.3	需求分析的基本任务	(27)
2.2	可行性研究	(31)
2.2.1	可行性研究的内容	(31)
2.2.2	可行性研究的具体步骤	(32)
2.2.3	可行性研究报告的内容	(33)
2.3	初步需求获取方法	(33)
2.3.1	访谈与会议	(33)
2.3.2	观察用户工作流程	(34)
2.3.3	建立联合小组	(34)
2.3.4	其他获取方法	(34)
2.4	需求分析的过程及方法	(36)
2.4.1	需求分析过程	(36)
2.4.2	软件需求建模	(38)
2.4.3	需求分析方法	(39)
2.5	软件需求规格说明和需求评审	(41)
2.5.1	软件需求规格说明和初步用户手册	(41)
2.5.2	软件需求评审	(43)
小结		(44)
习题		(44)
第3章	软件设计基础	(46)
3.1	软件设计概述	(46)
3.1.1	软件设计在开发阶段的重要性	(46)
3.1.2	软件设计的任务	(47)
3.1.3	软件设计的目标	(48)
3.1.4	软件设计的过程	(48)
3.2	软件设计的基本概念	(51)
3.2.1	抽象与逐步求精	(51)
3.2.2	模块化	(54)
3.2.3	信息隐蔽	(56)
3.2.4	模块独立	(57)
3.3	软件体系结构设计	(67)
3.3.1	基本概念	(67)
3.3.2	体系结构设计的风格	(70)
3.4	数据结构设计	(73)
3.4.1	数据结构设计	(73)
3.4.2	在设计程序结构时数据结构的选择方法	(73)
3.5	软件过程设计	(75)
3.5.1	结构化程序设计方法	(75)
3.5.2	图形设计工具	(77)

3.5.3 判定表和判定树	(79)
3.5.4 过程设计语言 PDL	(81)
3.6 软件设计规格说明及评审	(81)
小结	(83)
习题	(84)
第4章 结构化分析与设计	(85)
4.1 结构化分析方法概述	(85)
4.2 数据流图	(87)
4.2.1 数据流图的图形表示	(87)
4.2.2 分层数据流图的画法	(88)
4.2.3 分层数据流图的审查	(92)
4.3 数据字典	(94)
4.3.1 数据字典的描述符号	(94)
4.3.2 字典条目	(95)
4.3.3 数据字典的实现	(98)
4.4 描述基本加工的小说明	(99)
4.5 E-R 图	(102)
4.6 结构化设计方法概述	(103)
4.7 数据流图到软件体系结构的映射	(104)
4.7.1 信息流类型	(104)
4.7.2 映射步骤	(105)
4.7.3 变换分析	(106)
4.7.4 事务分析	(107)
4.8 启发式设计策略	(108)
小结	(110)
习题	(110)
第5章 面向对象的分析与设计	(112)
5.1 面向对象的概念	(113)
5.1.1 对象(Object)	(113)
5.1.2 类(Class)	(114)
5.1.3 实例(Instance)	(114)
5.1.4 消息(Message)	(115)
5.1.5 属性(Attribute)	(115)
5.1.6 操作和方法(Operation and method)	(115)
5.1.7 继承(Inheritance)	(116)
5.1.8 封装(Encapsulation)	(116)
5.1.9 多态性(Polymorphism)	(116)
5.1.10 重载(Overload)	(116)

5.1.11 面向对象方法的要点	(117)
5.2 UML 概述	(117)
5.2.1 UML 发展历史	(118)
5.2.2 UML 简介	(118)
5.3 RUP 统一过程开发模型	(120)
5.4 用例建模	(122)
5.4.1 用例图(Vse Case Diagram)	(122)
5.4.2 用例图分析流程	(125)
5.4.3 用例建模小结	(126)
5.5 静态建模	(126)
5.5.1 类图(Class Diagram)	(127)
5.5.2 类与类之间的静态关系	(128)
5.5.3 对象图(Object Diagram)	(132)
5.5.4 包图	(133)
5.5.5 静态建模的基本方法及举例	(134)
5.5.6 静态建模小结	(137)
5.6 动态建模	(138)
5.6.1 顺序图(Sequence Diagram)	(138)
5.6.2 合作图(Collaboration Diagram)	(142)
5.6.3 状态图(State Diagram)	(144)
5.6.4 活动图(Activity Diagram)	(147)
5.6.5 动态建模小结	(151)
5.7 物理体系结构建模	(151)
5.7.1 构件图(Component Diagram)	(151)
5.7.2 部署图(Deployment Diagram)	(154)
5.8 面向对象的软件建模工具	(155)
5.8.1 建模工具简介	(155)
5.8.2 Rational Rose 简介	(156)
小结	(159)
习题	(160)
第6章 人机界面设计	(162)
6.1 人的因素	(162)
6.1.1 人对感知过程的认识	(162)
6.1.2 用户的技能和用户间的差异	(162)
6.2 人机界面的风格与交互方式	(163)
6.2.1 人机界面的风格	(163)
6.2.2 人机界面的交互方式	(165)
6.3 人机界面的设计过程	(167)
6.3.1 界面设计中涉及的模型	(167)

6.3.2 任务分析与建模	(168)
6.3.3 界面设计中的一般问题	(168)
6.3.4 实现工具	(170)
6.3.5 界面评估	(170)
6.4 界面设计中应遵循的原则	(171)
小结	(174)
习题	(174)
第7章 编码	(176)
7.1 编码概述	(176)
7.1.1 编码的质量要求	(176)
7.1.2 程序设计语言的发展与特性	(177)
7.1.3 程序设计语言的特点与选择	(179)
7.2 编码的风格	(182)
7.2.1 程序的风格	(182)
7.2.2 程序的效率	(184)
7.3 程序编码优化	(185)
7.3.1 程序编码结构优化的基本方法	(186)
7.3.2 程序编码效率优化的基本方法	(187)
7.4 程序的复杂性度量	(189)
7.4.1 代码行度量法	(189)
7.4.2 McCabe 度量法	(189)
7.4.3 Halstead 方法	(191)
小结	(192)
习题	(192)
第8章 软件测试	(194)
8.1 基本概念	(194)
8.1.1 软件测试的定义	(194)
8.1.2 软件测试的基本原则	(195)
8.1.3 软件测试的种类	(196)
8.1.4 软件测试的信息流	(197)
8.1.5 软件测试的对象	(198)
8.2 软件测试技术	(198)
8.2.1 黑盒测试	(198)
8.2.2 白盒测试	(205)
8.3 软件测试的策略	(212)
8.3.1 单元测试	(213)
8.3.2 集成测试	(215)
8.3.3 确认测试	(218)

8.3.4	系统测试	(219)
8.4	测试工具	(220)
8.4.1	人工测试	(220)
8.4.2	测试工具的分类	(222)
8.4.3	测试工具的选择	(223)
8.5	调试	(224)
8.5.1	调试过程	(224)
8.5.2	调试的策略	(225)
8.5.3	调试的原则	(226)
	小结	(227)
	习题	(227)
第9章	软件维护	(229)
9.1	软件维护的概念	(229)
9.1.1	软件维护的含义	(229)
9.1.2	软件维护的内容	(229)
9.2	软件维护的特点	(230)
9.2.1	软件工程与软件维护的关系	(230)
9.2.2	维护成本	(231)
9.2.3	影响维护工作量的因素	(232)
9.3	软件维护的过程	(233)
9.3.1	建立维护机构	(233)
9.3.2	维护工作流程	(233)
9.3.3	保存维护记录	(235)
9.3.4	评价维护结果	(236)
9.4	软件的可维护性	(236)
9.4.1	可维护性的地位	(236)
9.4.2	软件的可维护性度量	(236)
9.4.3	提高软件可维护性的方法	(237)
9.5	程序的修改	(239)
9.5.1	程序修改的步骤	(239)
9.5.2	修改程序的副作用	(240)
9.6	软件再生工程	(241)
	小结	(242)
	习题	(242)
第10章	软件项目管理	(244)
10.1	软件项目管理概述	(244)
10.1.1	什么是软件项目管理	(244)
10.1.2	软件项目管理的内容	(244)

10.2	软件度量	(246)
10.2.1	度量、测量和估算	(246)
10.2.2	面向规模的度量	(246)
10.2.3	面向功能的度量	(248)
10.2.4	代码行度量与功能度量的比较	(250)
10.3	软件质量的度量	(250)
10.3.1	软件质量及三层次度量模型	(250)
10.3.2	软件质量要素	(251)
10.3.3	软件质量要素评价准则	(251)
10.4	软件项目的估算	(253)
10.4.1	代码行、功能点和工作量估算	(253)
10.4.2	软件项目估算模型	(255)
10.5	软件项目进度安排	(256)
10.5.1	进度安排中应考虑的几个问题	(256)
10.5.2	进度安排的方法	(257)
10.6	软件风险管理	(258)
10.6.1	风险的分类	(258)
10.6.2	风险的识别	(258)
10.6.3	风险评估	(259)
10.6.4	风险的管理与控制	(260)
10.7	软件项目的组织	(261)
10.7.1	软件项目管理的组织模式	(261)
10.7.2	人员组织与管理	(262)
10.8	软件配置管理	(263)
10.8.1	软件配置管理基本概念	(263)
10.8.2	软件配置管理的主要活动	(264)
10.9	软件过程成熟度模型(CMM)	(265)
10.9.1	基本概念	(266)
10.9.2	软件过程能力成熟度等级	(266)
小结		(267)
习题		(268)
第11章	软件工程标准化与软件文档	(269)
11.1	软件工程标准化的意义	(269)
11.1.1	什么是软件工程标准	(269)
11.1.2	软件工程标准化的意义	(269)
11.1.3	软件工程标准的制定与推行	(270)
11.2	我国的软件工程标准化工作	(271)
11.2.1	软件工程标准分类	(271)
11.2.2	我国的软件工程标准化工作	(272)

11.3	软件文档的作用与分类	(273)
11.3.1	什么是软件文档	(273)
11.3.2	软件文档的作用	(274)
11.3.3	文档的分类	(274)
11.4	软件文档管理	(275)
11.4.1	主要文档	(275)
11.4.2	文档的使用	(276)
11.4.3	软件生命周期与文档编写	(276)
11.4.4	编写文档应考虑的因素	(277)
11.4.5	文档管理	(279)
11.5	GB/T8567—2006 建议的基本文档格式规范	(281)
11.5.1	GB/T8567—2006 标准的结构	(281)
11.5.2	面向对象软件的文档编制	(283)
11.5.3	GB/T8567—2006 标准建议的基本文档格式规范	(283)
小结		(297)
习题		(297)
第12章	Web 工程	(299)
12.1	基于 Web 的系统与应用特点	(299)
12.2	Web 工程	(301)
12.2.1	Web 工程过程	(301)
12.2.2	Web 工程开发方法	(303)
12.2.3	开发方法应具备的特性	(303)
12.2.4	Web 工程开发工具	(304)
12.3	Web 项目策划	(304)
12.3.1	Web 工程团队	(304)
12.3.2	组建团队	(305)
12.3.3	Web 工程的项目管理	(306)
12.4	Web 应用分析	(308)
12.4.1	Web 应用分析的任务	(308)
12.4.2	Web 应用分析活动	(309)
12.5	Web 应用设计	(310)
12.5.1	Web 应用设计的任务	(310)
12.5.2	Web 应用设计活动	(310)
12.6	Web 应用测试	(315)
12.6.1	Web 应用测试的基本概念	(315)
12.6.2	Web 测试策略	(316)
12.6.3	Web 测试过程	(316)
12.6.4	内容测试	(318)
12.6.5	用户界面测试	(319)

12.6.6 构件级测试	(322)
12.6.7 导航测试	(323)
12.6.8 配置测试	(324)
12.6.9 可信、安全性测试	(325)
12.6.10 性能测试	(326)
小结	(327)
习题	(327)
参考文献	(328)

第1章 软件工程概述

随着计算机技术、电子技术的迅速发展,计算机硬件的性能和质量越来越高,而其成本和价格越来越便宜。计算机软件的应用使得信息的获取、处理越来越便捷。计算机与全球互联网络 Internet 的连接,使今天的社会进入了以计算机为核心的信息社会。计算机的应用从单纯的科学计算,渗透到工业、农业、教育、国防、商业等各个领域。

计算机应用越来越广泛的同时,人们对计算机软件的需求量越来越大,这促使人们对计算机软件的品种、数量、成本和开发时间等提出越来越高的要求。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。和计算机硬件相比,计算机软件的开发效率却远远跟不上软件应用的需求。要想使计算机软件功能越强、使用越方便,开发出来的软件就越复杂、越庞大,人们的软件开发能力越显得力不从心,以致软件开发计划一拖再拖,成本失去控制,软件质量得不到保证。软件质量和软件生产率满足不了计算机系统发展的需求,在某种程度上成为计算机系统发展的瓶颈。为了扭转这种局面,自 20 世纪 60 年代末期以来,人们努力探索软件开发的新思想、新方法和新技术,并在这些领域取得了重要的成果,从而形成了计算机科学技术领域的一门新兴学科——计算机软件工程学。

本章介绍软件和软件工程的基本概念,包括软件、软件危机、软件工程、软件生存期和开发模型、软件工具和环境等等。

1.1 软 件

要正确理解软件工程的概念,首先要正确理解软件的含义,认识软件的特点,了解软件的分类。

1.1.1 软件的概念与特点

软件对人类而言是一个全新的东西,其发展历史有五六十年,人们对软件的认识是随着计算机技术的发展经历了一个由浅到深的过程。

自第一台计算机问世以来,计算机软件经历了多个发展阶段,计算机软件系统也不断成熟和更为复杂,其发展历史可以分为如下四个阶段。

第一阶段(20 世纪 50 年代初期至 20 世纪 60 年代初期):这个阶段是计算机系统开发的初级阶段。这个阶段计算机的存储容量小,运算速度慢。编写程序的工具只有低级语言,程序的规模较小且程序的开发主要采用个体工作方式,程序的编写者和使用者往往是同一个组或同一个人。软件编程简单,不存在什么系统化的方法,程序的质量完全依赖于程序员个人的技巧。此时人们尚无“软件”的概念,人们对程序相关文档的重要性认识不足,开发没有任何管理,开发的软件除了程序清单外,几乎没有其他文档资料保存下来。实际上,这一阶段的主要研究对象是顺序程序,开发的计算机系统大多采用批处理技术,计算机的应用领域主要是以数值数据处理为主的科学计算。

第二阶段(20 世纪 60 年代中期至 20 世纪 70 年代末期):在这个阶段,随着计算机硬件的

迅速发展,计算机存储器越来越大,外围设备也越来越丰富。高级程序设计语言的出现,使得工程技术人员也能进行程序的编制。多用户系统引入了人机交互的概念,实时系统的出现使得进程的控制和输入的产生以毫秒来进行。为了适应大量数据处理问题的需要,产生了第一代数据库管理系统。

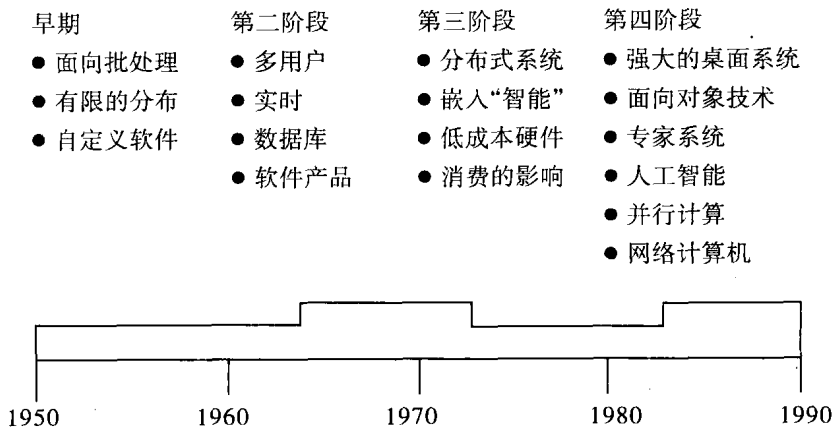


图 1-1 计算机软件发展的四个阶段

在这个时期,人们逐渐认识到开发文档的重要性,并出现了软件和“软件作坊”的概念。开发人员不再像早期那样只为自己工作需要而编写程序,更多的是为满足用户的需要而开发软件,帮助用户通过软件更好地使用计算机。程序的编制方式逐步从个体转向合作方式,出现了“软件作坊”。由于“软件作坊”仍然沿用早期形成的个体开发方法,缺乏有效的工程化方法的指导,使得“软件作坊”开发的软件很多不能按照计划完成,有的夭折,有的软件难以维护且耗费惊人的资源,这使得软件开发整体局面变的难以控制,从而出现了早期的软件危机。人们随之开始寻求采用工程化的方法来解决软件危机问题,并进一步导致了软件工程实践的出现。

第三阶段(20 世纪 70 年代中期至 20 世纪 80 年代末):分布式系统的出现极大地提高了计算机系统的复杂性,计算机网络的发展对计算机软件的开发提出了更高的挑战,微处理器的出现使得嵌入式系统得到了广泛应用。软件产品的销售额不断增加,软件产品的品种和质量有了一定程度的提高,然而硬件价格持续下降与发展大大超过了人们对软件的需求的速度,进一步导致了软件危机的加剧,致使更多的科学家和工程师着手研究软件工程学的科学理论、方法等一系列问题。

第四阶段(20 世纪 80 年代末至今):这个阶段软件产业在信息产业中已经占据举足轻重的地位。在强大的桌面系统和计算机网络迅速发展的时期,面向对象的技术在许多领域取代了传统的软件开发方法,计算机体系结构由中央主机控制方式变为客户机/服务器方式,人工智能技术和专家系统开始应用于软件。多媒体技术提高了计算机信息处理的能力,改变了与最终用户的通信方式,同时高性能计算机的体系结构也发生了很大变化,并行计算、分布式计算、网格计算成为计算机软件领域研究的热点。

从计算机技术发展的几个阶段可以看出,软件从个性化的程序变为工程化的产品经历了一个长期过程,在此期间人们对软件的认识也发生了变化。在早期有相当一部分人认为计算机软件就是计算机程序,这个理解是不完全的。

这里引用美国软件工程教材作者 R. S. Pressman 的定义:

软件(software)=程序(program)+数据(data)+文档(document)

其中程序是按照事先设计的功能和性能要求执行的指令或者语句序列;数据是程序能正常操纵信息的数据结构;文档是描述程序操作和使用的文档。

所以程序并不是软件,程序是软件的组成部分。随着软件开发技术的发展,人们对软件的认识发生了根本性的变化,从“软件=程序”发展为“软件=程序+数据+文档”。软件的需求成为软件发展的动力,软件的开发从自给自足模式发展为在市场中流通以满足广大用户的需要。软件工作的考虑范围也发生了很大变化,人们不再只顾及程序的编写,而是涉及到软件整个生命周期。

要真正理解软件的含义需要了解软件的特点,从而明白软件与人类建造的其他事物的区别。软件同传统的硬件相比,具有如下特点:

(1) 软件是一种逻辑实体,具有抽象性,不是一般的物理实体。它看不见,摸不着,没有物理形态。逻辑往往实际只存在于人的头脑当中,只能通过分析、判断、思考,使用智力活动才可以认识软件,了解其功能和性能等;也只有使用智力才可以对软件的相关属性(例如,其规模大小、开发难度等)进行把握。

(2) 软件的生产与硬件的存在某些相同点,但二者有根本上的不同。两者都可通过优秀的设计获得高得产品质量,硬件的产品质量在制造过程控制,而软件不存在明显的制造过程,其产品质量只能在研制开发过程中控制。软件的开发,是人的智力的高度发挥,而不是传统意义上的硬件制造,它更依赖于开发人员的素质、智力、人员和组织、合作和管理。在大多数场合,软件的开发、设计几乎都是从头开始的,开发的成本和进度很难估计。软件成本集中于开发上,而硬件生产主要在集中在制造上,这同时也意味着软件项目不能像硬件制造项目那样来管理。

(3) 软件维护与硬件的维修有着本质的差别。不仅软件在运行期间可能会暴露潜伏的错误时需要进行软件维护,用户有时需要提高和完善软件的性能或者支撑软件产品运行的硬件或软件的环境的改变等情况下也都需要提供软件维护。软件维护没有硬件维护那样有可替换的标准零件,有时还需经历软件开发的整个生命周期过程,因此软件维护更为复杂,成本更高。

(4) 软件在运行和使用期间,没有硬件那样的机械磨损,老化问题,但是软件存在退化问题。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)。硬件在早期具有较高的失效率,失效率主要是由于设计或生产的缺陷造成的。当这些缺陷纠正后,失效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,失效率又开始抬高,进入了磨损失效期。而软件是数字产品,不存在磨损和老化问题。图 1-2(b)表明了软件的失效率率曲线,在软件的生命初期隐藏的错误会使程序具有较高的失效率,理想的情况下当这些错误改正后曲线便趋于平稳,但实际情况是因为软件维护的需要,随着软件的每次修改和变更有可能引入新的错误,导致软件失效率曲线呈现图中所示的锯齿状,失效率会陡然上升,即软件出现了退化。不断的变更是软件退化的根本原因,软件的维护因为没有备用的构件也变得更为复杂。

(5) 基于构件的开发方法由于其自身的特点越来越受到人们的重视,这些技术可以减少开发时间、提高质量,并提高复用水平。虽然软件向着基于大规模构件的组装模式发展,但大多数软件是根据客户实际需求定制的,而不是通过已有构件组装而成的。而硬件工程师在构建新系统时可以通过大量已有的标准化构建进行,可复用的构件可以使硬件工程师关注设计中的创新部分。

从软件的特征可以看出软件比硬件的生产更为复杂。软件本质上的复杂性导致了软件产