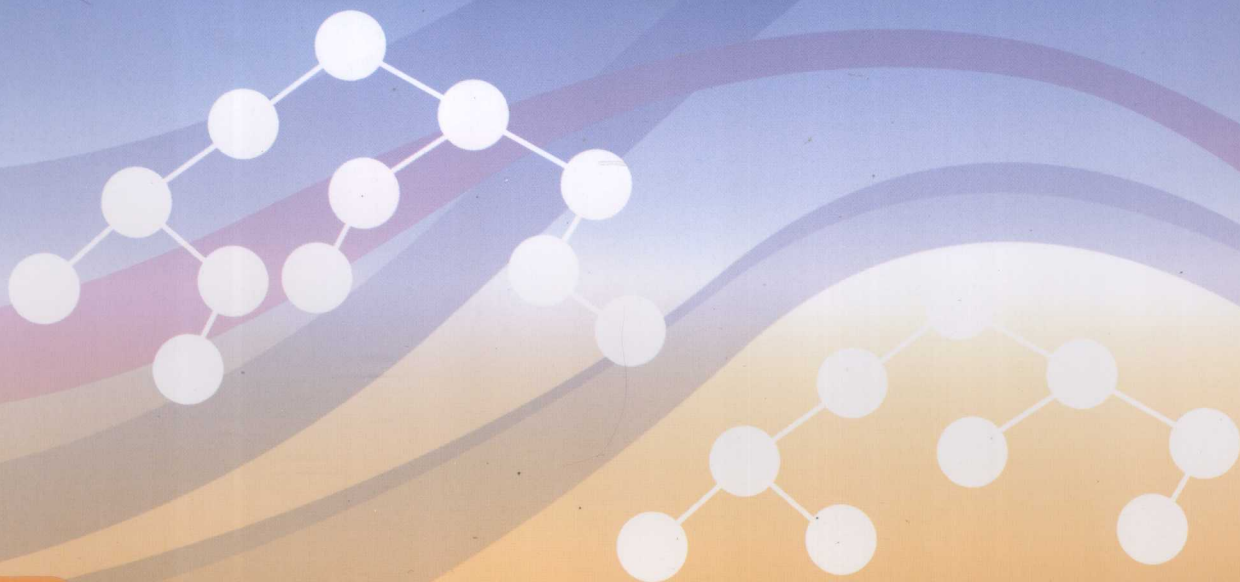


高等学校计算机专业规划教材

软件工程 基础与应用



马小军 等编著



清华大学出版社

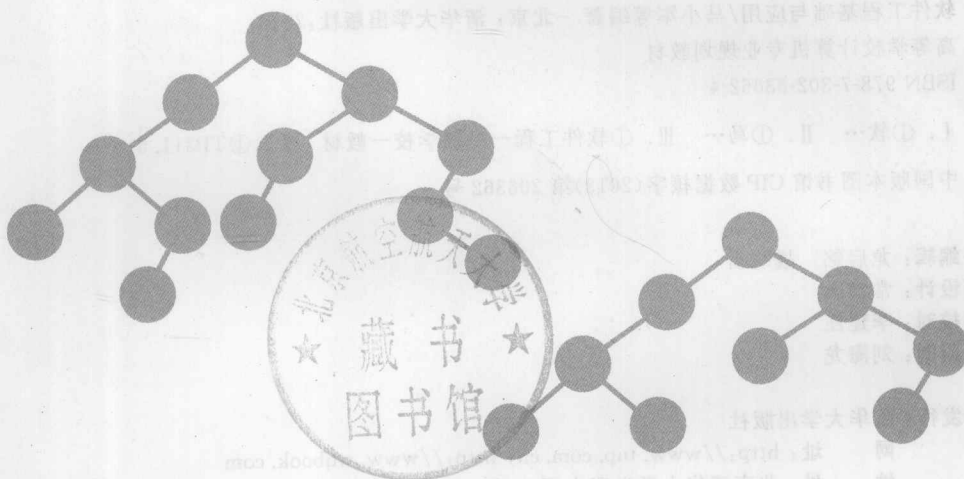
013071096

高等学校计算机专业规划教材

TP311.5
552

软件工程 基础与应用

马小军 等编著



北航

C1680240

清华大学出版社
北京

TP 311.5

552

内 容 简 介

本书以软件工程基本理论知识的普及和常用开发方法具体运用的介绍为出发点,对软件工程的主要原理、基本概念、主流技术方法的特点和使用规则、软件测试与维护的一般方法以及软件规模估算与项目的管理等进行了全面描述。全书共 10 章,第 1 章和第 2 章是软件工程综述,第 3 章是需求调研与可行性研究,第 4 章至第 6 章是系统分析与设计,并重点介绍基于结构化技术的分析与设计方法以及编码实现的基本原则,第 7 章和第 8 章是基于面向对象技术的分析与设计方法以及统一建模语言 UML,第 9 章是软件测试技术与维护方法,第 10 章是软件项目管理。配合知识点的介绍,每章不但有简单举例和丰富的习题,而且以一个规模和难度适中的项目为中心进行综合举例说明,并贯穿书中的主要章节,便于对所学内容的正确理解和实际应用的认识。

本书既注重理论也强调实际应用,内容清晰实用,案例简明完整。本书既可作为高等院校相关专业本科生软件工程课程的教材或参考书,也可作为应用软件开发人员、项目管理人员和专业技术人员编写技术文档的参考资料。

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。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

软件工程基础与应用/马小军等编著. —北京:清华大学出版社,2013

高等学校计算机专业规划教材

ISBN 978-7-302-33662-4

I. ①软… II. ①马… III. ①软件工程—高等学校—教材 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2013)第 206362 号

责任编辑:龙启铭 战晓雷

封面设计:常雪影

责任校对:李建庄

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:14.5

字 数:360 千字

版 次:2013 年 9 月第 1 版

印 次:2013 年 9 月第 1 次印刷

印 数:1~2000

定 价:29.00 元

产品编号:049125-01



软件工程学科的发展有其历史的必然。近半个世纪以来,随着通信、计算机和网络应用的普及,作为其灵魂的软件的开发显得越来越重要。无数的反例证明,如果软件产品的质量达不到要求,带来的损失是极其严重的。为保证或提高软件产品的质量,最关键的问题就是从技术和管理两方面双管齐下,使用得到实际考验的一系列最佳软件开发实践,作为我们工作的指导原则,切实做好软件开发的各项活动。

有人讲,软件工程课程学不学没有用处。事实上是这样吗?绝对不是。我在给清华大学计算机系的工程硕士上软件工程课的时候,他们说,他们上的所有课程中,这门课最有用。因为他们都是从事软件开发多年的在职研究生,他们反映,多年来困惑他们的很多问题,从课程中都能找到对应的解决方法。有时,看上去是一句普通的“原则”,在实践上却能解决大问题。所以,虽然课本中讲了许多条条框框,其实都是有其实践背景的。

还有人讲,软件工程这门课程太枯燥,听不懂,做不会,学习起来提不起兴趣。这就是学习方法问题了。软件工程有一条主线,即软件生命周期过程。它的特点是分阶段、有迭代。

从软件开发方法来看,不论是传统的结构化方法,还是面向对象方法,或新的面向服务架构,它们都有各自的适用领域,有不同的视角、不同的活动组织方式和不同的架构。

从软件工程过程来看,最基本的是开发过程、运行过程和维护过程,此外还有各种支持过程和组织过程,它们为基本过程提供辅助支持和各种保证。

从软件工程管理来看,有整体管理,包括启动、计划、执行、控制和收尾5大过程,此外,还要考虑需求、成本、进度、质量、人员、沟通和风险等方面的管理活动。

如果我们明确了软件工程的主要方面,就可以有目的、系统地进行课程的学习了。

总之,对于软件工程,应首先想到它是有用的,也可能将来工作后用不到操作系统原理、计算机原理等课程所讲的内容,但只要是从从事软件开发,软件工程就是回避不了的。其次,要有它不难学的思想。关键是对将来自己工作的领域要有规划,找准方向,有针对性地学习。特别地,由于社会需求

是不断变化的,软件开发技术是不断进步的,软件开发工具是层出不穷的,我们不要一味追求新的,而应该从最成熟、最可靠的技术和工具学起。在学校里打好基础,将来在工作中就很容易接受新东西。

马小军老师从事软件工程的教學已經 20 多年了,我與她的相識是在 20 世紀 90 年代初她聽我的軟件工程課的時候。她是一位對工作很認真的老師,在這麼多年的教學中已經積累了豐富的教學經驗,並領導了許多軟件開發項目,從如何培養應用型技術人才角度出發,對軟件工程的教學體系、課程內容選材和實踐活動組織等各方面都有很好的想法,這本教材的編寫就體現了她的教學思路和多年教學的體會,有理論,有案例,全書在內容組織方面注重科學性和系統性,在行文敘述方面頗具簡潔性和可讀性。我推薦同學們認真閱讀,切實領會其實質,不斷進步。

清華大學計算機系教授 殷人昆

2013 年 7 月



前言

软件工程以研究如何高效率地开发高质量、高可靠性、易维护的软件产品为核心内容,自从20世纪60年代末推出以来,对软件产业的发展起到了巨大的推动作用。软件工程思想的严谨性、开发过程的规范化,为软件项目开发提供了理论保障;各种开发技术和开发方法的涌现和使用、计算机硬件性能的极大提高,以及网络技术和多媒体技术等不断完善,为开发功能健全、性能良好、用户满意的软件提供了技术支持。由此也促进了软件在各行各业、各个领域中的广泛应用,成为企事业单位全面实现信息化建设的核心内容。随着软件应用的广泛化与内部功能更新的频繁化,用户对软件质量和健壮性的要求更加突出,只有严格遵循工程化和规范化思想的指导,软件开发才可能获得成功。

在很多人的概念里,软件开发就是编写程序,即便是一些公司里面的技术人员,对软件工程也缺乏正确的认识,项目开发过于随意,导致公司虽然能够获得眼前的经济效益,但因产品在后期不断出现问题,而直接影响了企业的形象,长远收益大打折扣。21世纪以来,软件工程作为独立的学科体系,与计算机科学、信息工程和计算机工程等并存,体现出国家对软件技术人才的培养更加重视。信息化发展步伐的迈进,使软件人才的社会需求量也显著提高。软件开发不再是软件专业学生独自掌握的知识和专有技能,而变得更加大众化和普遍化。因软件工程不仅强调技术及其应用,也是一种分析问题和解决问题的思想和方法。所以,系统学习软件工程的知识,对任何专业以及将来做任何工作的人而言,都是十分有意义和必要的。

目前市面上出版的软件工程教材比较多,所介绍的理论知识和开发技术也很全面,但各部分的举例系统化和完整性略显不够,特别是对于一些非计算机专业的学生而言,欠缺很多软件方面的专业基础知识,理解起来有一定困难。其结果导致学生在对各章节知识的理解和持续性,特别是技术方法的完整运用方面出现困难。

为此,我们结合对软件工程的了解与感悟以及多年的教学体会与项目开发经验,对软件工程的基本概念、基本理论和主流技术进行系统梳理后编写了本书,书中不仅进行了理论、技术和工具的介绍,同时选择了一个难度和规模适中且学生容易理解的项目作为综合案例,贯穿书中的主要章节,目的是帮助学生正确理解所学内容,系统化地掌握和认识软件工程的思想和技术在实际开发中的具体运用,从而使教材的阅读、学习达到理论与实践的

密切结合。

全书共 10 章,依照软件生存周期的理念展开。

第 1 章和第 2 章是软件工程综述,主要介绍软件的概念和特点、软件工程提出的背景和基本原理、几种主流开发方法、软件生存周期的组成和开发模型以及常用的建模工具等。

第 3 章是需求调研与可行性研究,主要介绍需求调研的基本方法、用户业务流程的描述、项目可行性分析及软件成本/效益分析的常用方法等。

第 4 章和第 5 章是系统分析与设计,并重点介绍基于结构化技术的分析与设计方法,包括主要任务、工作原理、基本原则以及数据流程图、数据字典和软件结构图的构建与优化等。

第 6 章是详细设计与编码实现,主要介绍算法的常用描述工具、界面设计需要注意的问题以及一般编码原则等。

第 7 章和第 8 章是基于面向对象技术的分析与设计方法以及统一建模语言 UML,主要介绍面向对象分析与设计的基本原理、工作过程及面向对象技术和 UML 中的主要视图模型与构建方法等。

第 9 章是软件测试技术与维护,主要介绍测试的概念与方法、软件调试的步骤与方法以及软件维护的概念、影响维护的因素和提高可维护性的方法。

第 10 章是软件项目管理。主要介绍软件规模估算方法、风险分析与监控、人员的组织与管理、进度与软件质量的控制等。

作为主编,马小军负责本书的组织结构和统稿,并编写了第 1 章、第 2 章和第 4 章,廖礼萍编写了第 5 章、第 7 章和第 8 章,第 3 章、第 9 章和第 10 章由张冰峰编写,第 6 章由马楠编写。全书由马小军和张玉祥审校。

在本书的编写过程中,参阅了大量的文献和资料,在此向这些文献资料的作者深表崇敬之意并致以衷心的感谢。

本书的编写以北京联合大学信息学院软件工程平台课教学改革为依托,突出内容的实用性和系统化以及图形的规范化和一致性,语言通俗易懂,结构编排合理;所有举例简明、完整,同时,书后配有丰富的习题。书中所有图形(除界面截图以外)均是用 Microsoft Office Visio 2003 或 2007 绘制的。

本书既可作为高等院校相关专业本科生软件工程课程的教材或参考书,也可作为项目管理人员、应用软件开发人员和专业技术人员的技术参考资料。

我们希望读者通过阅读本书,了解软件工程的理念,理解两种主流技术的特点和基本原则,掌握各种模型和建模工具的使用。同时,通过综合举例,感受软件项目开发过程中系统分析和系统设计的关系与具体实施方法,为今后独立从事一个小型软件项目的开发提供参考和帮助。但由于时间关系且作者水平有限,书中难免会存在问题和不妥之处,真诚地希望读者和软件工程领域专家能够提出宝贵的意见和建议,以帮助我们逐步完善和修正。作者联系方式:xxtxiaojun@bnu.edu.cn。

作者

2013 年 5 月于北京



目录

第 1 章 概述 /1

1.1 软件的基本概念及特点	1
1.1.1 软件是什么	1
1.1.2 软件的分类	3
1.1.3 软件的发展	4
1.2 软件工程的提出	6
1.2.1 软件工程提出的导火索	6
1.2.2 软件工程的原理	7
1.2.3 软件工程的目标	9
1.3 软件开发方法	10
1.3.1 结构化方法	10
1.3.2 原型化方法	11
1.3.3 面向对象方法	12
1.3.4 敏捷开发方法	13
本章小结	14
习题	15

第 2 章 软件生存周期模型 /16

2.1 软件生存周期及其组成	16
2.1.1 软件生存周期的概念和提出的意义	16
2.1.2 软件生存周期的划分与组成	17
2.2 软件开发模型	18
2.2.1 瀑布模型	19
2.2.2 快速原型模型	20
2.2.3 增量模型	21
2.2.4 螺旋模型	21
2.2.5 喷泉模型	22
2.3 常用软件建模工具	23
2.3.1 Microsoft Visio	23
2.3.2 Rational Rose	25

本章小结	26
习题	27

第3章 用户需求与可行性研究 /28

3.1 用户需求调研	28
3.1.1 关于需求	28
3.1.2 用户需求调研的必要性	28
3.1.3 需求调研方法	30
3.1.4 调研内容和步骤	30
3.2 业务描述	33
3.2.1 业务流程定义	33
3.2.2 业务流程图	34
3.3 软件项目可行性分析	36
3.4 成本/效益分析	37
3.4.1 软件开发成本估算	38
3.4.2 软件效益分析	39
本章小结	41
习题	41

第4章 基于结构化方法的需求分析 /42

4.1 需求分析简述	42
4.1.1 何为需求分析	42
4.1.2 需求分析的主要任务	43
4.1.3 需求分析的原则	44
4.1.4 需求分析方法的分类	45
4.2 结构化分析	45
4.2.1 结构化分析的特点和原则	45
4.2.2 结构化分析的主要工具	46
4.3 数据流程图的构建	47
4.3.1 构建数据流程图的作用	47
4.3.2 数据流程图的基本符号	48
4.3.3 构建数据流程图的步骤	49
4.3.4 构建数据流程图需要注意的问题	50
4.4 数据字典	53
4.4.1 构建数据字典的意义和用途	53
4.4.2 数据字典的主要内容	53
4.4.3 构建数据字典使用的符号	54
4.4.4 数据字典举例	55



4.5 结构化分析综合举例	55
4.5.1 项目简介	55
4.5.2 系统数据流程图	56
4.5.3 系统数据字典	59
本章小结	61
习题	61

第5章 系统概要设计 /63

5.1 简述	63
5.1.1 概要设计基本任务	63
5.1.2 概要设计基本方法	65
5.2 概要设计基本原理	66
5.2.1 模块化	66
5.2.2 抽象	67
5.2.3 逐步求精	67
5.2.4 信息隐藏和局部化	68
5.2.5 模块独立性	68
5.3 结构化系统设计	71
5.3.1 软件结构图	71
5.3.2 数据流程图的分类	72
5.3.3 变换型设计	73
5.3.4 事务型设计	74
5.4 HIPO图	78
5.5 软件结构的优化准则	79
5.6 结构化设计综合举例	81
本章小结	83
习题	83

第6章 详细设计与编码实现 /85

6.1 详细设计的基本任务	85
6.2 详细设计的常用工具	85
6.2.1 程序流程图设计	86
6.2.2 盒图	87
6.2.3 PAD图	87
6.2.4 判定表与判定树	88
6.2.5 过程设计语言	90
6.3 人机界面的设计	90
6.3.1 人机界面一般风格	91



6.3.2	人机界面设计原则及相关问题	92
6.3.3	人机界面设计过程	93
6.3.4	人机界面设计各级定义与基本要求	94
6.4	详细设计综合举例	95
6.4.1	人机界面	95
6.4.2	数据库设计	96
6.4.3	模块详细设计	98
6.5	编码实现	102
6.5.1	编码工具介绍	102
6.5.2	编码工具的选择	102
6.5.3	编码原则	102
6.5.4	编码举例	103
	本章小结	104
	习题	104

第7章 面向对象技术 /106

7.1	面向对象概述	106
7.1.1	面向对象提出的背景	106
7.1.2	面向对象方法简介	107
7.1.3	面向对象基本概念	108
7.1.4	面向对象的特征	111
7.2	面向对象分析	111
7.2.1	面向对象分析任务概述	112
7.2.2	面向对象分析模型	113
7.2.3	面向对象分析过程	119
7.3	面向对象设计	120
7.3.1	面向对象设计任务	120
7.3.2	面向对象设计与面向对象分析的关系	121
7.3.3	面向对象设计基本原则	122
7.3.4	面向对象设计模型	124
7.3.5	面向对象设计过程	124
	本章小结	127
	习题	127

第8章 UML 建模 /128

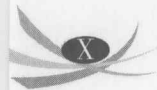
8.1	UML 概述	128
8.1.1	发展历史	128
8.1.2	UML 简介	129



8.1.3	视图	129
8.1.4	UML 中的图	130
8.2	用例模型的构建	131
8.2.1	用例图的模型元素	132
8.2.2	确定参与者	133
8.2.3	确定用例	133
8.3	静态模型的构建	137
8.3.1	类图和对象图	137
8.3.2	标识类的方法	141
8.3.3	包图	145
8.4	动态模型的构建	147
8.4.1	顺序图	147
8.4.2	协作图	149
8.4.3	状态图和活动图	149
8.5	物理体系结构的构建	153
8.5.1	组件图	153
8.5.2	部署图	153
8.6	综合举例	155
8.6.1	系统分析	155
8.6.2	系统设计	158
8.6.3	物理结构设计	158
	本章小结	161
	习题	161

第9章 测试与维护 /163

9.1	软件测试简介	163
9.1.1	软件测试定义	163
9.1.2	软件测试的原则	164
9.1.3	软件测试的对象	165
9.2	软件测试模型	165
9.2.1	V 模型	165
9.2.2	W 模型	166
9.3	软件测试的一般过程	167
9.3.1	测试需求分析阶段	167
9.3.2	测试计划阶段	167
9.3.3	测试方案设计阶段	168
9.3.4	测试用例实现阶段	168
9.3.5	测试执行阶段	168



9.3.6	评价与关闭阶段	169
9.4	软件测试常用方法	169
9.4.1	黑盒测试与白盒测试	169
9.4.2	黑盒测试法	169
9.4.3	白盒测试法	174
9.5	面向对象测试	177
9.6	软件调试	179
9.6.1	软件调试基本概念	179
9.6.2	软件测试和软件调试的区别	179
9.6.3	软件调试的步骤	179
9.6.4	软件调试方法	180
9.7	软件维护	181
9.7.1	软件维护的定义	181
9.7.2	影响维护工作量的因素	181
9.7.3	软件维护成本	182
9.7.4	软件维护过程	182
9.7.5	软件的可维护性与提高方法	184
	本章小结	185
	习题	185

第 10 章 软件项目管理 /187

10.1	软件规模估算	187
10.1.1	软件规模估算概述	187
10.1.2	软件规模估算常用方法	187
10.2	风险管理	189
10.2.1	软件项目风险定义及管理重要性	189
10.2.2	风险识别	190
10.2.3	风险分析及量化	190
10.2.4	风险应对计划	190
10.2.5	风险监控	191
10.3	人员组织	191
10.3.1	团队管理概述	191
10.3.2	软件项目组织计划	191
10.3.3	项目组织形式	192
10.3.4	团队的组建	193
10.3.5	团队合作与沟通	195
10.4	进度管理	195
10.4.1	进度管理的定义	195



10.4.2	进度管理过程.....	195
10.4.3	软件项目任务分解.....	196
10.4.4	软件项目进度计划.....	197
10.4.5	进度控制.....	201
10.5	质量管理.....	202
10.5.1	软件质量的定义.....	202
10.5.2	软件质量模型.....	203
10.5.3	软件质量保证.....	204
10.5.4	软件质量控制.....	205
	本章小结.....	206
	习题.....	206

附录 软件工程文档模板 /207

参考文献 /216

第 1 章

概 述

随着软硬件技术的发展,在软件工程思想指导之下进行软件开发,已经被广大软件技术人员和管理人员普遍认可、接受并在实际项目中运用。了解和学习软件工程的知識,对于开发出高质量的软件产品至关重要。

自世界上第一台电子计算机在美国诞生至今,计算机技术的发展和應用已经完全超出了当时发明团队成员的预期,特别是近几年各种形式的计算机设备如笔记本、平板电脑和 iPad 等相继问世,给人们的工作和生活带来了极大的便捷,已经成为日常不可或缺的必需品和工具。但是,没有安装软件的计算机(通常称之为裸机)是无法为人提供任何服务的,硬件设备作用的发挥是离不开软件的控制的。软件是什么?与程序有何区别和联系?为何要提出软件工程?软件工程的提出对软件开发领域带来了怎样的影响?在强调信息化建设的时代,软件技术的先进性与实用性对信息化的发展、文化与系统安全有怎样的影响?本章将对这些问题进行详细论述。

本章要点:

- 软件的基本概念及特点;
- 软件工程的提出背景与主要思想;
- 主流的软件开发技术与方法。

1.1 软件的基本概念及特点

世界上第一台电子计算机 ENIAC 诞生于 1946 年,占地 170m²,重量约 3×10⁴kg,计算能力为每秒可执行 5000 次的加法运算。当时,人们见到这个体积庞大、能够代替人完成数据分析计算的设备,其内心的兴奋和神奇感难以言表。虽然相比于今天我们使用的 PC 和笔记本等计算机而言,ENIAC 内部结构相当简单,但是它对人类在现代化计算设备领域的创新发展开拓了方向,在数据处理技术的提出和改进方面奠定了基础,对整个社会进步带来了重要的影响。

1.1.1 软件是什么

在硬件设备不断发展、改进的同时,能否充分发挥它的作用,在其上完成更复杂的工作,是那个年代技术人员的理想和梦想,“软件”这一术语也正是在这个追梦的过程中于 20 世纪 60 年代被正式提出的。但是在很长时间里,人们认为软件就是程序,做软件就是

编程序,这种错误的认识直至今日依然存在,而且并不鲜见。软件到底是什么?与程序有何本质区别?怎样描述才能够准确体现它的内涵呢?

1. 程序

程序(program)是用计算机语言描述的、能够被计算机内部所包含的各种硬件元器件识别和处理的指令(或语句)序列。任何程序无论大小,编写目的都是为解决某一个特定问题,为需要的用户提供一个具体、有效的服务。

学习过计算机程序语言的人们都知道,编写程序强调逻辑性,但逻辑是否正确,只有在计算机上实际运行才能够证明。运行过程中,需要数据的支持,若没有具体的数据被接收或处理,程序是不能产生任何结果的,因此也无法验证程序逻辑的正确性。用户看不到操作结果,不可能接受和继续使用,程序则失去存在的价值。

2. 软件

IEEE(Institute of Electrical and Electronics Engineers,国际电子与电器工程师协会)对软件(software)给出了如下的定义:软件是计算机程序、方法与规则、相关的文档资料以及在计算机上运行时需要的数据的集合。与程序相比,软件更调整体的逻辑性,对于任何一个非计算机专业的人而言,它是一个脱离了硬件就完全看不出效果的、充满神奇的、非实物化的逻辑部件。程序只是软件的内容之一,是结果的最终体现手段和工具。编写程序的过程中通常会涉及一些特殊的计算方法,我们称之为算法,这些算法既依赖于数学中的公式和逻辑推理,也要与行业内部的管理方法与规则保持一致。因此,一个软件的推出并非几个程序的编写和运行所能代表,也不是像家具和电器产品那样利用物理材料制作而成,而是一项复杂的生产过程,是人类智慧劳动的结果。如果将软件概括描述为

软件=程序+规则+文档+数据

那么,缺少任何一项内容,都不能称为软件,而且随着软件规模和复杂度的不断提高,规则和文档的作用更加突出。所以,认为软件即程序的认识是十分狭隘和完全错误的。

3. 软件的特点

相比于计算机硬件设备,软件因其依赖于严谨的逻辑分析,具有严格的推理生产过程,同时受运行环境和硬件性能的影响,导致具有下列突出特点。

1) 高智能性

任何软件都是复杂的脑力劳动的结晶,是数学、管理经验和经济分析等多种知识的综合运用,不仅能够帮助人们完成复杂和高精度的计算,还能够对大量数据进行多种形式的分析,进而作为调整发展的决策依据。

2) 无形态、不可见性

因为软件是逻辑分析的产物而非实物,所以是一个逻辑部件,既无外观形态,也不具有物理大小特征,是极其抽象的,无法按照一般实物的常用方法去制作和评判质量高低,更不能在运行之前凭主观想象对其进行效果评价和实用价值的定性描述。

3) 开发过程复杂性

通常软件是一个庞大的逻辑系统,其制作是一个复杂的开发过程,与其他行业的产品制作过程最大的区别是,中间过程的成果都是抽象的方案,既要体现业务流程,又要在满足功能要求的前提下最大限度地减少重复设计,协调调用关系,所以描述结果是一般人无

法理解的,即便是专业人员也可能会因为对内部复杂关系认识不足或理解不透而影响后期工作的进行。这使得软件开发成为非常复杂的工作。

4) 产品化的低成本性

原版软件开发完成之后,只要对其进行复制即可形成批量产品,然后安装在不同的计算机上,依照操作要求就可以运行使用,而不需要再投入大量的人力、财力和原材料。

5) 无磨损、无老化性

所有的物理设备,如计算机硬件、家用电器和汽车等,随着使用时间的延长,都会出现部件损坏或老化的情况,导致故障频繁出现,但是所有软件只要功能要求没有变化,本身没有人为的删除或破坏,新的运行环境能够兼容,即可以永久地使用下去,绝对不会因为反复使用而出现自动老化或自我损坏。例如,在 C++、C# 被大量用来开发基于面向对象技术的网络化系统的今天,作为结构化程序设计语言典型代表的 C 语言依然活跃于一些操作系统或底层控制系统的开发中,也仍然是企业招聘软件开发人才的一个基本技能考核内容。

6) 对硬件的依赖性

软件只有在硬件设备上运行,才能发挥并体现它的可使用性和价值,而且随着软件的发展,对硬件的性能要求也越来越高。如一些图像处理的软件或游戏软件,要求计算机内存和硬盘足够大,处理器的主频足够快,分辨率足够高等等。达不到基本的配置要求,软件的运行效果很难令人满意或根本无法运行。软硬件之间的这种密切依赖关系在其他实物化产品中是基本没有的。

7) 维护复杂性

由于软件是依据用户的要求而开发的,在投入使用的过程中,通常会暴露出一些潜在的错误,同时用户的工作流程和需求也会发生变化。遇到这些问题,都需要对软件进行修改和维护,而这项工作的复杂度有时相比于开发一个新软件并不简单,其投入是不可预估的。

1.1.2 软件的分类

随着计算机的发展与普及,软件也逐渐形式多样,用途各异,不同人士对软件的类别划分也存在一定的差异,目前主要提出了下列几种分类方法。

1. 按功能划分

根据软件的作用和功能,主要分为系统软件、应用软件、支撑软件和可复用软件 4 类。

- 系统软件:是计算机的底层软件,直接操控计算机硬件设备并与之进行交互,同时为其他软件的运行提供底层服务与支持。例如 Windows 和 UNIX 操作系统、数据库系统、打印机驱动程序等,都属于系统软件。
- 应用软件:能够为用户群提供综合服务的软件,通常又分为行业应用软件和通用软件等,如教务管理系统、财务软件、网络银行和游戏软件等。此类软件是通过具体的工具开发而成的,其运行基于系统软件。受服务的局限,每个应用软件的用户群有一定的范围,但相比于系统软件,其形式多样,使用领域广泛。
- 支撑软件:也被称为工具软件,是介于系统软件和应用软件之间的、具有支撑作