

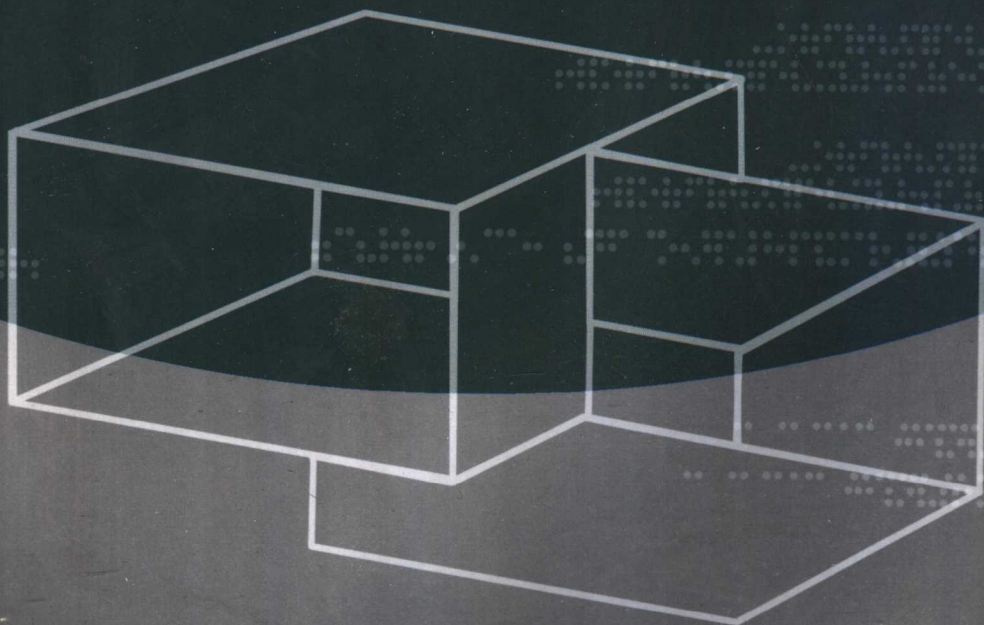


应用型本科规划教材

SOFTWARE ENGINEERING PRACTICE

软件工程实务

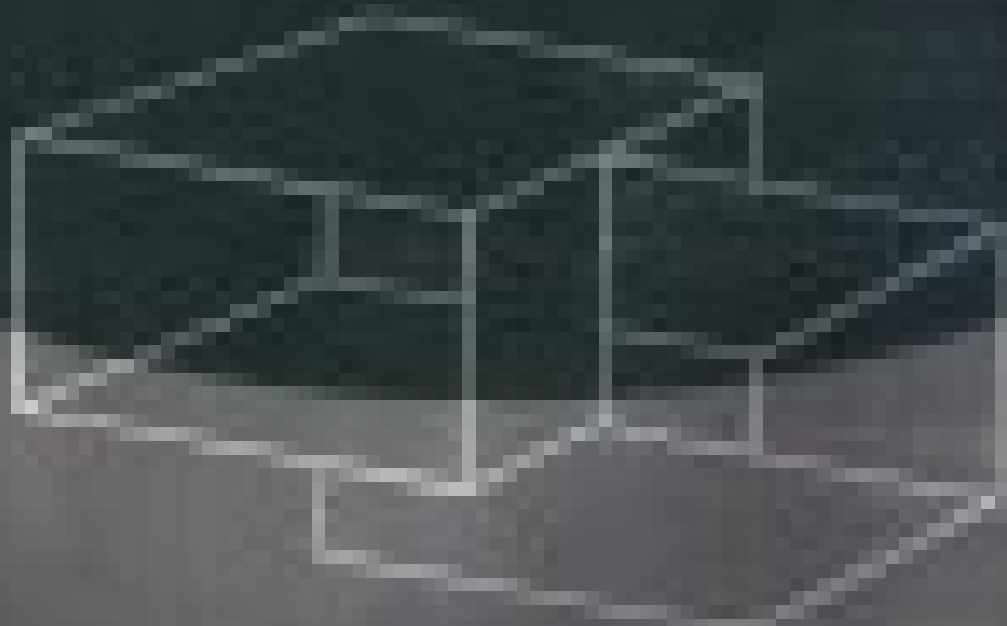
◆ 刘学俊 李继芳 刘汉中 编著



ZHEJIANG UNIVERSITY PRESS
浙江大学出版社

软件工程实务

第2版 李国良 李国良 李国良 李国良



清华大学出版社

● 应用型本科规划教材

Software Engineering Practice

软件工程实务

刘学俊 李继芳 刘汉中 编著

浙江大學出版社

图书在版编目(CIP)数据

软件工程实务/刘学俊,李继芳,刘汉中编著. —杭州:
浙江大学出版社,2007.8
ISBN 978-7-308-05527-7

I. 软... II. ①刘...②李...③刘... III. 软件工程 IV.
TP311.5

中国版本图书馆 CIP 数据核字(2007)第 137323 号

软件工程实务

刘学俊 李继芳 刘汉中 编著

责任编辑 石国华

封面设计 刘依群

出版发行 浙江大学出版社

(杭州市天目山路 148 号 邮政编码 310028)

(网址: <http://www.zjupress.com>)

排 版 星云光电图文制作工作室

印 刷 杭州杭新印务有限公司

开 本 787mm×1092mm 1/16

印 张 17.75

字 数 443 千

版 印 次 2007 年 8 月第 1 版 2007 年 8 月第 1 次印刷

书 号 ISBN 978-7-308-05527-7

定 价 28.00 元

版权所有 翻印必究 印装差错 负责调换

浙江大学出版社发行部邮购电话(0571)88072522

序

据国家信息产业部统计,到 2007 年国家紧缺软件人才高达 80 万人,东部沿海地区软件人才匮乏与经济发展速度严重不相符的状态,已经引起了政府和各界的广泛关注。随着计算机技术高速发展,高校计算机类教材可谓让人眼花缭乱。每到选择教材时任课教师们都要为选择合适的教材而劳神。软件工程是一门指导计算机软件开发和维护的工程学科,是一门新兴的边缘学科,是培养软件人才的核心课程。翻开专业经典的软件工程书籍,抽象的理论让人望而却步,难以适合从业人员学习的需求。研究和建设适应高校应用型人才培养的教材是高校教学改革和教材建设的紧迫任务。

《软件工程实务》一书是在浙江省 21 世纪教学改革项目——“计算机实践性课程‘项目化’训练模式的研究与实践”的研究基础上形成的,书内很多例子来源于近两年学生参与社会科研项目实践之所得。该书由具有 20 余年软件研发经验和工程管理经验的高级工程师和 20 余年高校教学经验的教师合作完成,本书所探讨的软件工程观念、方法、策略和规范都是朴实无华的,既遵循于高校教学规律,内容精炼,循序渐进,重点突出,又较好地总结了实际工程项目的技术和方法,融合了多年工程经验与教育智慧。该书区别于其他同类教材的主要特点是始终强调软件工程理念在工程实践中的运用,特别是将软件工程中的沟通理念作为一个重要问题提出并加以讨论,富有特色和新意,在现有教材中尚不多见。

书中的实验部分体现了应用本书进行教学的组织特色——软件工程理论与软件工程实践紧密结合,强调了实务、实践和实验。因此,本书不失为一本极佳的应用型软件人才培养的教学参考书,在应用型人才培养的院校中极具使用价值。希望该教材能为软件人才的培养作出积极的贡献。

钱同英

2007 年 8 月

目 录

上篇 软件工程实务

第 1 章 软件、软件工程及沟通艺术	1
1.1 软 件	2
1.1.1 软件的概念	2
1.1.2 软件的特点	3
1.1.3 软件的分类	4
1.1.4 软件的发展	7
1.1.5 软件危机	8
1.2 软件工程概述	10
1.2.1 软件工程的概念	10
1.2.2 软件工程的研究内容	11
1.2.3 软件工程过程和软件生存周期	12
1.2.4 软件工程的目标及原则	13
1.2.5 软件工程的发展	15
1.3 软件开发模型	15
1.3.1 瀑布模型	16
1.3.2 原型模型	17
1.3.3 螺旋模型	17
1.3.4 喷泉模型	19
1.3.5 智能模型	20
1.4 CASE 工具及环境	20
1.5 软件工程中的沟通理念	21
1.5.1 软件工程中沟通的重要性	21
1.5.2 软件工程中的沟通理念	22
思考与讨论	35
第 2 章 软件项目管理基础	36
2.1 项目、软件项目及其特点	36

2.1.1	项目的四个要素	36
2.1.2	项目的周期性	37
2.1.3	软件项目的特点	38
2.2	项目、软件项目管理含义及特点	40
2.2.1	项目管理的含义	40
2.2.2	项目管理的特点	40
2.2.3	软件项目管理的特点	41
2.3	项目管理的基本职能	43
2.3.1	项目计划	43
2.3.2	项目组织	43
2.3.3	评价与控制	44
2.4	项目、软件项目管理的主要内容	44
2.4.1	项目管理的主要内容	44
2.4.2	软件项目管理的主要内容	45
2.5	软件项目的度量与估算	50
2.5.1	面向软件规模的度量	50
2.5.2	面向软件功能的度量	51
2.5.3	软件质量度量	53
2.5.4	影响估算的因素	57
2.5.5	目标分解估算法	58
2.5.6	软件项目成本预算	60
2.6	项目风险的识别、分析及预防	60
2.6.1	风险识别	60
2.6.2	风险分析	61
2.6.3	风险预防	62
2.7	软件项目计划与组织	63
2.7.1	制定项目实施计划的基础和主要内容	63
2.7.2	实施计划的进度安排	63
2.7.3	软件项目组织与任务责任矩阵	66
	思考与讨论	72
第3章	需求获取	73
3.1	系统界定与建立模型	73
3.1.1	系统与模型	73
3.1.2	系统模型建造举例	74
3.2	基于计算机的系统及其系统工程	75

3.2.1 基于计算机的系统	75
3.2.2 计算机系统工程	75
3.3 需求和需求的层次	78
3.3.1 什么是需求	78
3.3.2 需求的层次	79
3.3.3 需求获取的任务	80
3.4 需求获取的原则与方法	80
3.4.1 需求获取规范	80
3.4.2 需求获取的原则和方法	83
3.5 需求规格说明书	89
3.5.1 软件需求规格说明书	89
3.5.2 提出和验证初始需求	90
3.5.3 初始需求的汇编整理	90
3.5.4 需求的可跟踪性	91
3.5.5 确认 SRS 草案	92
3.5.6 需求规格说明书的复核与评审	92
3.5.7 平衡工作量	94
3.5.8 系统测试计划	94
3.5.9 修订和发布	94
3.6 需求获取阶段系统分析员责任和素质要求	94
思考与讨论	95
第4章 结构化分析与设计	96
4.1 软件结构化分析与设计方法	96
4.1.1 软件系统分析的目标	96
4.1.2 软件系统分析与设计活动的本质	96
4.1.3 软件系统基本结构模型	97
4.1.4 系统分析向结构设计过渡的工作任务	98
4.1.5 系统分析的分析过程和方法	100
4.1.6 软件设计阶段系统分析员的素质	104
4.2 软件设计基本规范	105
4.2.1 概要设计	105
4.2.2 模块化与模块设计	108
4.2.3 分析与设计模块的基本规范	111
4.3 软件设计规范的量化运用	116
4.4 表示软件结构的基本方法	118

4.4.1 系统概念设计	118
4.4.2 层次图和 HIPO 图	121
4.4.3 模块设计的精细设计方法	123
4.4.4 面向数据流的设计方法	125
思考与讨论	131
第 5 章 界面设计	133
5.1 界面设计的平面原则	133
5.2 软件操作过程设计原则	135
5.3 人机界面设计过程	138
思考与讨论	139
第 6 章 编码实现	140
6.1 编码概述	140
6.2 编码的标准	142
6.3 程序效率	148
6.4 程序设计语言的适用性	149
6.5 程序复杂性质量	150
思考与讨论	152
第 7 章 软件测试	153
7.1 软件测试的几个基本问题	154
7.1.1 什么是软件测试、测试工作的对象	154
7.1.2 项目组中的测试角色	155
7.1.3 Bug 和 Bug 跟踪	155
7.1.4 软件测试工作的原则	157
7.1.5 测试用例的不可穷举性	158
7.2 软件测试的基本方法和工具	159
7.2.1 静态分析和动态分析	160
7.2.2 黑盒测试和白盒测试	160
7.2.3 覆盖测试	161
7.2.4 使用测试	161
7.2.5 手动测试和自动测试	162
7.2.6 软件测试工具	162
7.3 软件测试策略	163
7.3.1 单元测试	163

7.3.2 集成测试	164
7.3.3 确认测试	164
7.3.4 系统测试	164
7.4 软件测试实务	165
7.4.1 测试与开发的并行关系	165
7.4.2 测试计划	166
7.4.3 测试过程规格说明书	168
7.4.4 测试过程	173
7.4.5 测试总结	174
思考与讨论	176
第8章 软件过程改进	177
8.1 软件过程	178
8.1.1 什么是过程	178
8.1.2 过程成熟度	179
8.1.3 过程定义是否妨碍了创造力的发挥	180
8.2 有效的软件过程架构	181
8.2.1 过程架构	181
8.2.2 规范过程的重要性	188
8.2.3 过程文档的意义	189
8.2.4 建设有效的软件过程	189
8.3 软件能力成熟度模型简介	191
8.3.1 CMM 中定义的软件开发团队角色与职责	192
8.3.2 CMM 中能力成熟度的级别	192
8.3.3 CMM 的内部结构——关键过程域	194
8.3.4 CMM 的使用	201
思考与讨论	202
第9章 面向对象方法概述	203
9.1 什么是面向对象方法	203
9.2 结构化与面向对象的思想差异	205
9.3 面向对象方法的主要概念	206
9.3.1 对象	206
9.3.2 其他概念	207
9.4 面向对象建模	209
9.4.1 对象模型	210

9.4.2 动态模型	214
9.4.3 功能模型	216
9.4.4 三种模型之间的关系	216
9.5 OO方法的SRS、OOA、OOD概要	217
9.5.1 面向对象的需求获取	217
9.5.2 面向对象的分析与设计	218
思考与讨论	220
附录1 软件工程文档规范	221
附录2 匈牙利命名法	239

下篇 软件工程实验指导

实验一 体验软件工程的“隔板画”	242
实验二 需求获取	248
实验三 结构化设计	254
实验四 结构化设计和实现	265
实验五 软件测试及其评估	266
实验六 软件过程改进实验	272

上篇 软件工程实务

第1章

软件、软件工程及沟通艺术

主要内容:软件;软件工程概述;软件工程的沟通技巧;CASE 工具及环境。

教学目标:使学生了解什么是软件及软件开发行业的一般性问题,了解软件开发行业中的几种角色和素质修养,启发学生思考择业的角色方向,学习软件工程中的沟通理念,通过“写项目”的锻炼,培养学生的综合能力。

难点和重点:软件工程中的沟通理念,软件生存周期。

自 1946 年计算机诞生以来,计算机的发展先后经历了电子管、晶体管、集成电路、大规模集成电路时代,目前,超大规模集成电路的计算机已经广泛应用。据悉,当今计算机的硬件性能平均每十几个月就会提高一倍!

与计算机硬件技术迅猛发展相伴随,计算机软件技术也得到了高速发展。从中国近 30 年的发展看,软件开发技术由一项只在“计算专业”讲授的计算机应用技术,发展到几乎所有理工科专业都不同程度地讲授软件开发技术,70%的工程技术人员接受过软件开发知识的培训,计算机应用技术已经成为各类学校的公共基础课。现实生活中的计算机应用,已经从 20 年前的“专用空调房”走向了车间、厂房、旷野、沙漠、天空和海底……互联网技术使得整个世界结成一体,计算机应用几乎无处不在。

随着计算机应用的迅猛普及,伴随而来的是人们对应用软件的数量和质量提出了更多、更好、更高的要求,计算机应用领域形成了多方向的产业化发展需求。遗憾的是,软件开发技术自身并没有像硬件技术那样飞速发展,为软件产品开发提供“催化剂”般的开发效率。相比之下,随着应用计算机技术期望值的高涨,人们却发现软件技术的发展存在着“瓶颈”:软件的可靠性没有保障、维护费用不断上升、开发进度和成本难以控制等等。人们甚至用“软件危机”来形容当今软件开发行业面临的困境。

为了扭转这样的局面,人们期望用类似工程的方法来解决软件开发中的困难。早在 1968

年,北大西洋公约组织的一次学术会议上,软件工程(Software Engineering)作为一个概念首次被提出,这意味着人们开始注重用工业化的开发模式替代小作坊的开发模式,软件开发劳动从主要以个体的小规模编程活动进入到团队性的工业化生产技术产品的时代。软件工程理论的核心思想是把软件产品看作是一个像其他工业产品一样的工程产品来处理,把协调控制其他工程的理念类似地应用于软件开发过程当中,实现对开发进度、费用和质量的控制;软件工程理论还从软件产品自身的特点出发,针对性地研究相应地问题和解决对策。

人们在如下两个方向上做了大量工作:

一是在开发工具方面,研究自动或半自动地进行软件开发的工具,而且有人把诸多软件工具集成起来,使得一种工具产生的信息可以为其他工具所使用。这样建立起一种称之为计算机辅助软件工程(CASE)的软件开发支撑系统(CASE 将各种软件工具、开发机器和一个存放开发过程信息的工程数据库组合起来形成一个软件工程环境)。

二是在开发软件过程方面,研究了大量不同规模、不同复杂程度、不同应用领域的软件开发过程实践,积累了相当数量的被实践证明了正确的管理理念和管理技术经验(简称为“最佳实践”)。特别是近 20 年来,“软件工程”课程作为这些经验的结晶,已经成为培养计算机专业及其应用技术人才的一门专业核心课程,越来越受到业内的高度重视。

1.1 软件

1.1.1 软件的概念

“软件”(Software)一词是在 20 世纪 60 年代初从国外翻译过来的,有人翻译为“软制品”,也有人翻译为“软体”,现在统一称之为“软件”。

早期,对于 Software 的一种公认的解释为,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。所谓“完整集合”强调的是缺乏程序、数据、相关文档中任何一部分均不能称之为软件。

还有一种解释为,软件是与计算机系统操作有关的程序、规程、规则及任何与之有关的文档和数据。它由两部分组成:一是计算机可执行的程序和数据;二是计算机不可执行的、与软件开发、运行、维护、使用和培训有关的文档。

近年来,有一种新的说法认为,软件是领域知识与程序、数据和文档的总称。

三种说法都是建立在如下基本概念的基础之上的: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作信息的数据结构的简称;文档是与程序开发,维护和使用有关的图文材料。

虽然对于软件定义的说法还远不止上述三种,但是他们本质上是基本一致的,只是各自有所侧重。第一种说法全面、准确地概括了软件的内涵;第二种在第一种的基础上更加明确地指出了文档、数据与程序的相互关系;第三种定义指出程序、数据和文档,是对领域知识作出解释或描述的一种模型。从系统论出发,模型是不可能完备的,模型的完善一定是随着人们对领域问题认识的深入而完善的,也由此决定了软件的目标和软件生存与发展。

从软件工程学说的角度来看,我们更倾向于采用第三种说法作为软件的定义,即:软件是领域知识与程序、数据和文档的总称。

1.1.2 软件的特点

1. 软件是一种智力密集型的技术产品,由“1”到“百、千、万……”的专业复制性制造过程不是十分困难,而从“无”到“有”产生“1”的过程充满了大量的、高强度的脑力劳动。因此说,软件的生产过程主要是设计过程,它具有两个显著特点:抽象、不可见。

“抽象”讲的是从许多事物中舍弃个别的、非本质的属性,抽出共同的、本质的属性。它是形成概念的基础。软件的抽象性有两个层面:一是软件的功能层面,指的是已经完成的软件具有了哪些功能;二是软件的构建层面,指的是正在研制的软件应该具有怎样的功能以及如何实现这些功能。对一般用户来说,从功能层面理解软件并不困难;而要从构建层面理解软件就不是一件很容易的事情了,需要专门的学科性训练。这就是说“软件抽象”的原因,指的是理解它比较困难。

“不可见”讲的是指软件产品本身是一种不可见的逻辑实体,它是一个物理性状与功能性状相去甚远的“虚拟”存在(物理性状可以是穿孔的纸带、或卡片,也可以是磁盘;功能性状是可以回答问题的“智慧”),是附着在计算机这种特定的物理设备之上的“灵魂”(离开计算机以后,软件的存在就失去了现实意义)。特别是对于软件产品的质量,不论是使用者还是开发者都难以用简单的尺度加以度量,即便是专业人员,如果没有足够专业能力或者不够深入地了解软件产品的开发过程,想清晰地把握软件产品的质量也不是一件容易的事情。故此,软件具有不可见性。

正因为软件是逻辑实体,因此它具有不会磨损、老化的特点。

2. 开发软件过程具有不确定性,这是软件产品的另一显著特点。

在技术层面,软件开发过程是把人们主观的一些想法和概念,通过某种人为的设计,以结构、关系、算法、进程、流程等形式再现出来,是建造一个模拟以上想法和概念的模型的过程;人为设计的模型必然需要修改和完善,最终才能得到一个优化的结果,这是很自然的建造过程,而优化过程本身包含着丰富的内容和众多不同的方案,因此,软件产品的成功开发,是一个十分复杂、遍布着众多不确定性因素考验的过程。

在市场运作和商机捕捉层面,决策者的想法也会因时过境迁而发生转变;政治业绩的驱使和人际关系的变化等很多因素都会影响到软件的目标和开发资源的投入发生变化。问题领域非技术层面的情境变化对软件开发过程和软件产品质量产生着巨大的影响。

3. 目标的挑战性与问题的恒常性,对于从事软件开发人员的自身素质和修养提出了更高的要求。

通常,人们对于目标范畴以内的研究对象都抱有极大的兴趣,目标的挑战性会使热爱本职工作的人燃起热情之火,并迅速投入达到目标的工作,达到目标即为得到结果。但是,问题领域目标完善和目标变化的问题是恒常的——是永恒存在,经常发生的,是问题领域自身发展的必然结果。很多人没有对此做好足够的准备,表现为仅仅对实现目标有利的事情给予关注,其余一概不管不顾,一旦达到了目标了,便如同“马拉松”跑到了终点,一动也不想动了,什么完善资料、整理文档这些原本应该在开发进程中同时做好的工作,已经被抛到九霄云外去了。

开发软件必须首先要组建一支互相协作、优势互补的软件开发团队。人人具备写好文档的技能水平,具有健康积极的生活态度和强烈的社会责任感。团队负责人不仅要解决技术和管理理念水平的问题,还要解决团队文化建设问题。

4. 软件开发行业虽然取得辉煌的成绩,但与其他科学技术的发展水平相比较,仍然处在不

够成熟、甚至有些落后的发展阶段。

比如,软件开发仍然处在手工作业阶段,而很多成熟的科学技术及其相关产业产品生产大多已经达到自动化。软件产品大多是“定做”的,很少能做到利用现成的部件组装成所需的软件。尽管某些“最佳实践”达到了这种程度,但是从整个行业来说距此甚远。近年来软件技术虽然取得了不少进展,提出了许多新的开发方法,例如充分利用现成软件的复用技术、自动生成技术等,也研制了一些有效的软件开发工具或软件开发环境,但在软件项目中采用的比率仍然较低。由于传统的手工业开发方式仍然占据统治地位,开发的效率自然受到很大的限制。对于软件人员来说,开发工作仍然是一种高强度的脑力劳动。

5. 软件是复杂的。人类创造的产品的复杂性本质上取决于问题领域的复杂性。软件开发,特别是应用软件的开发常常涉及许多领域的专门知识,这对软件人员提出了很高要求。软件复杂性与开发软件技术的发展不相适应的状况越来越明显。开发软件技术发展落后于应用软件复杂性需求,并且随着时间的推移,这个差距日益加大。

6. 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,它的成本是比较高的。问题不仅于此,值得注意的是硬件和软件的成本 40 年来发生了戏剧性的变化。无论是直接研发,还是向厂家购买,在 20 世纪 50 年代末,软件费用仅占总费用的百分之十几,到了 80 年代这个比例完全颠倒过来,软件的费用大大超过硬件费用,90 年代的情况更是这样。

7. 相当多的软件工作涉及社会因素。类似于企业管理类型的软件自然是不言而喻的。许多软件的开发和运行涉及机构、体制及管理方式等问题,涉及人的观念和心理。对于这些人的因素重视得不够,常常是软件工作遇到的问题之一。例如,由于主管部门对正在开发的软件不够理解,因而软件开发得不到应有的重视和必要的支持,造成人力和资金上的困难,它直接影响到项目的成败。任何软件项目的结果,都是以提供一个特定的产品或服务为目标的,即便目标功能偶然相同,在实现的技术路线和方法上也会存在较大的差别,造成这种差别的原因经常是这种社会因素,并非是技术路线方面的深思熟虑。

1.1.3 软件的分类

事实上,给计算机软件作出科学的分类是很难的。但是鉴于不同类型的工程对象,对其进行开发和维护的工作会有很大差异,因此仍然需要对软件类型进行必要的划分。

1. 按软件的功能进行划分

(1) 系统软件:是与计算机硬件紧密配合在一起,使计算机系统各个部件以及相关的软件和数据协调、高效地工作的软件。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。系统软件的工作通常是频繁地与硬件和用户交互、资源的共享与复杂的进程管理,以及复杂数据结构的处理。系统软件是计算机系统必不可少的一个组成部分。

(2) 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。

(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,这些领域的应用软件种类繁多。其中商业数据处理软件是所占比例最大的一类,工程与科学计算软件大多属于数值计算问题。此外,应用软件在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)等方面大显神通,使得传统的产业部门面目一新,给人们带来惊人的生产效率和巨大的经济效益。事务管理、办公自动化方面的软件也在企事业单位迅速推广,中文信息处理、计算机辅助教学(CAI)等软件使得计算机向家庭普及。

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。

(1)微型:一个人在半日到几天之内完成的软件。写出的程序不到5百行语句,仅供个人专用。通常这种小题目无须作严格的分析,也不必要有一套完整的设计、测试资料。但是这并不是说可以随便地不讲任何方法地做。事实证明,即便是这样小的题目,如果经过一定的分析、系统设计和结构化编码,并且有步骤地进行测试,肯定也是非常有益的。

(2)小型:一个人半年之内完成的2千行以内的程序。例如,数值计算问题或是数据处理问题就是这种规模的课题。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,只是没有大题目那样严格。

(3)中型:3个人以内一年多时间里完成的5千到5万行的程序。这种课题开始出现软件人员之间、软件人员与用户之间的联系、协调的配合关系等问题。因而计划、资料书写以及技术审查需要比较严格地进行。这类软件课题比较普遍,许多应用程序和系统程序就是这样的规模。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人员的工作效率起着重要作用。

(4)大型: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。例如编译程序、小型分时系统、应用软件包、实时控制系统等是这类软件。参加工作的软件人员需要按二级管理,例如划分成若干小组,每组5人以下为好。在任务完成过程中,人员调整往往不可避免,因此会出现对新手的培训和逐步熟悉工作的问题。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,往往会在开发的过程中出现一些事先难以估计的不测事件。

(5)极大型:100至1000人参加用4到5年时间完成的具有100万行程序的软件项目。这种极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、军事指挥系统通常都有这样的规模。很显然,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)巨大型:2000人到5000人参加,10年内完成的1000万行以内的程序。这类软件很少见,往往是军事指挥、弹道导弹防御系统。

从以上介绍可知,规模大、时间长、很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作指导。而规模小、时间短、参加人员少的软件项目也得有软件工程理念,遵循一定的开发规范,其基本原则是一样的,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

(1)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,其处理时间应是严格限定的,如果在任何时间超出了这一限制,都将影响系统运行,甚至造成事故。

(2)分时软件:允许多个联机用户同时使用计算机,系统把处理机时间分成时间片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的软件。

(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。近年来,终端设备更加普及,交互式软件随处可见。一个重要的问题日益显得突出,这就是用户界面设计问题,良好的用户界面设计能够给用户带来极大的方便。

(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,按顺序逐个处理完的软件。这是最传统的工作方式。

4. 按软件服务对象的范围划分

完成软件工程项目后可以有两种情况提供给用户:

(1)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、卫星控制系统的软件就属于这一类。这类项目软件中有的软件带有试验研究性质,完成项目后根据需要可在此基础上作进一步开发。为取得客户的委托项目,软件开发机构的质量管理、技术实力、开发经验以及履行合同的信誉成为受到重视的问题。

(2)产品软件:是由软件开发机构开发出来直接提供给市场,或是为千百个用户服务的软件。这是一些服务于多个目的及多个用户的软件。例如,文字处理软件、财务处理软件、人事管理软件等。由于要参与市场竞争,其功能、使用性能以及培训和售后服务显得尤为重要。

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。由于若干年才进行一次普查,前些年开发的软件在若干年以后很难适用。有的统计资料或试验数据须按年度作统计分析,相应的软件每年运行一次。另外有些问题,需要每天及时进行数据处理,如天气预报。这类软件具有较高的使用频度。显然,开发不同使用频度的软件,有不同的要求,不可同等看待。

6. 按软件失效的影响进行划分

运行在不同领域的软件,适应其不同的需求,在运行中对可靠性也有不同的要求。有的软件如果在工作中出现了故障,造成软件失效,可能给整个软件系统带来的影响不大。但有的软件一旦失效,可能酿成灾难性后果,其严重损失难以挽回。例如控制载人飞行的软件,是必须万无一失的。

事实上,随着计算机进入国民经济和国防系统的各个重要领域,其软件的可靠性显得越来越重要。如财务金融、交通通信、航空航天等。人们一般称这类软件为关键软件,其特点在于:

- (1)可靠性等要求高。
- (2)常与完成重要功能的大系统的处理部件相连。
- (3)含有可能对以下各项造成影响的程序:
 - 人员或公众的安全;
 - 设备或设施的安全;
 - 环境的质量;
 - 国家的政务或部队的军务;
 - 数据、通信或实体的机密。